Review of level-set reinitialization methods in computational mechanics and materials science

Modesar Shakoor

OPEN ACCESS
IOP Publishing
Modelling and Simulation in Materials Science and Engineering
Modelling Simul. Mater. Sci. Eng. **33** (2025) 053001 (49pp) https://doi.org/10.1088/1361-651X/ade17b

Topical Review

Review of level-set reinitialization methods in computational mechanics and materials science

Modesar Shakoor

IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Materials and Processes, F-59000 Lille, France

E-mail: modesar.shakoor@imt-nord-europe.fr

Received 17 March 2025; revised 27 May 2025
Accepted for publication 5 June 2025
Published 16 June 2025

Abstract

The level-set (LS) method has been widely spread since its introduction in 1988. One of its main features is that interfaces are represented through signed distance functions, and the distance property eases the smoothing of discontinuities across an interface, and the computation of normal vectors and mean curvatures. This property can be lost whenever the LS function is evolved, for instance, through an LS transport equation. Numerous studies proposing so-called LS reinitialization methods to restore the distance property have been published since 1988. This paper is a review of numerical developments on LS reinitialization in the past decade (2014–2024). LS reinitialization methods are classified into three categories: direct methods which geometrically compute distances, local indirect methods which solve the Eikonal equation point-by-point, and global indirect methods which solve the Eikonal equation for all points at once. The review focuses on numerical methods and investigates the following questions. Can it be implemented in a parallel computing environment with nearly optimal scalability? Can it be used with any approximation method, and is it compatible with unstructured grids? Can it be extended to reach higher-order convergence rates? Can it be combined with mass change error attenuation techniques? Does it involve any numerical parameters that may affect its robustness? Is it limited only to some applications? Through a quantitative and qualitative analysis of the past decade's literature,

this review paper proposes novel insights on LS reinitialization. Research on direct methods should focus on parallel efficiency and robust higher-order distance computation techniques. More attention should be given to local indirect methods, especially regarding parallel and higher-order algorithms for unstructured grids. Control parameters for global indirect methods should be better determined or eliminated. More research is needed on the issue of blind spots in two-phase problems involving moving contact lines and their elimination.

Keywords: level-set, reinitialization, redistancing, review

1. Introduction

Due its multiple advantages, the level-set (LS) method has been widely used in the last decades for representing interfaces, especially for two-phase flow problems. Among the issues that limit its use, there is the requirement of implementing the so-called LS reinitialization or redistancing step.

1.1. LS method

A LS is a signed distance function to an interface. Let $\Omega = \Omega(t)$ be a simulation domain separated into two parts Ω_1 and Ω_2, with $\Omega_1 \cap \Omega_2 = \Gamma$ and $\Omega_1 \cup \Omega_2 = \Omega$, then the LS function ϕ is

$$\phi(\boldsymbol{x},t) = \begin{cases} +\mathrm{dist}(\boldsymbol{x},\Gamma(t)), \boldsymbol{x} \in \Omega_1(t), \\ 0, \boldsymbol{x} \in \Gamma(t), \\ -\mathrm{dist}(\boldsymbol{x},\Gamma(t)), \boldsymbol{x} \in \Omega_2(t). \end{cases} \tag{1}$$

The idea of representing interfaces implicitly through LS functions was initially proposed in [100]. It has widely spread and the LS method has become a serious alternative to volume-of-fluid methods, explicit interface meshing, and other usual candidates for representing interfaces. Among advantages of the LS method, it should be pointed out that topological changes such as bubble merge in fluid mechanics or grain nucleation in materials science can be modeled with no extra difficulty. The regularized Heaviside function smoothing discontinuities across an interface:

$$H_\epsilon(\phi) = \begin{cases} 1, & \phi > \epsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right)\right), & |\phi| \leqslant \epsilon, \\ 0, & \phi < \epsilon, \end{cases} \tag{2}$$

the normal vector:

$$\boldsymbol{n}(\phi) = \frac{\nabla\phi}{|\nabla\phi|}, \tag{3}$$

and the mean curvature:

$$\boldsymbol{\kappa}(\phi) = \nabla.\boldsymbol{n}(\phi), \tag{4}$$

can be computed easily thanks to the distance property. These quantities are often computed only in a small layer of half-thickness ϵ around the interface:

$$\Omega_\epsilon = \{\boldsymbol{x} \in \Omega, |\phi(\boldsymbol{x})| \leqslant \epsilon\}. \tag{5}$$

1.2. LS reinitialization

One of the key ingredients in an implementation of the LS method for applications in computational mechanics and materials science is the LS reinitialization or redistancing step. The LS method, indeed, usually involves solving an LS transport equation:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \frac{\partial\phi}{\partial t} + \nabla.(\boldsymbol{v}\phi) = 0, \text{ in } \Omega, \tag{6}$$

using a transport solver and a fixed mesh, a moving mesh, or a combination of both. The velocity field $\boldsymbol{v}$ might originate from a flow solver (Navier–Stokes equations) or another physical model. The distance property is usually lost during interface motion and deformation when solving equation (6), which is problematic if this property is used to calculate geometric information. Problems for which the LS method is the most interesting, in fact, such as those involving topological changes, are also those where the distance property is most likely to be lost. It is therefore important to periodically reinitialize the LS function or, in other words, restore the distance property.

Although LS reinitialization could be considered a small step or a marginal aspect of the LS method, a wide variety of approaches have been proposed and used in the literature. They can be distinguished in the way they restore the distance property: directly by computing distances, or indirectly by solving partial differential equations locally or globally. Other differentiating properties include:

- the computational cost and the parallel efficiency,
- the approximation method and the compatibility with unstructured grids,
- the accuracy and in particular the convergence rate and the amount of mass change errors,
- the robustness in relation to the presence or absence of numerical/tuning parameters,
- the capability to deal with a wide range of applications.

It seems therefore important to review the literature on this topic and dive deeper into it.

1.3. Review methodology

This paper proposes an original review of the literature in the past decade (2014–2024) on LS reinitialization methods. This review attempts to identify criteria that could explain why so many different methods have been proposed and used and why research on LS reinitialization is still active.

The methodology consisted of searching for the keywords *LS* and *reinitialization* (or *redistancing*) using the Web of Science™ tool and restricting the results to the last decade. Voluntarily, papers focusing on applications of the LS method to image segmentation (e.g. [6, 59]) or topology optimization (e.g. [25, 30, 52]) have been omitted, and the review focuses on papers involving simulations of moving fronts in computational mechanics (e.g. two-phase flows) and materials science (e.g. grain growth). These simulations rely on the distance property for physical models. LS reinitialization, therefore, plays a crucial role because simulations may diverge or give non-physical results if the LS function is not properly reinitialized.

All these aspects and some additional ones such as mass conservation are discussed in the present review.

LS reinitialization methods used in the literature in the last decade are presented in section 2, with a distinction between direct methods, local indirect methods and global indirect methods.

These methods are then discussed altogether in section 3, where they are differentiated in terms of target properties. Concluding remarks are finally presented in section 4.

2. LS reinitialization methods

LS reinitialization methods can be separated into two main categories. Direct reinitialization methods (DRMs), on the one hand, consist in geometrically computing distances by projecting each discretization point onto the interface $\Gamma_h(t) = \{\boldsymbol{x} \in \Omega(t), \phi_h(\boldsymbol{x},t) = 0\}$, where ϕ_h is the numerical approximation of ϕ. Indirect methods, on the other hand, consist in solving the Eikonal equation:

$$\begin{cases} \left(\nabla\hat{\phi}\right)^2 = 1, \text{ in } \Omega, \\ \hat{\phi} = 0, \text{ on } \Gamma, \end{cases} \tag{7}$$

where $\hat{\phi}$ is the reinitialized LS function. All indirect methods are iterative. Some of them, however, solve the Eikonal equation locally and propagate corrections progressively, element-by-element or cell-by-cell. Remaining methods solve a partial differential equation globally or layer-by-layer.

For many methods, moreover, discretization points actually carrying the interface are treated separately. These points have a different definition depending on the numerical approximation method being used (FE, FD, FV, DG, etc). For an FE method, let $\mathcal{T}_h$ be the set of elements, and $\mathcal{N}_h$ be the set of nodes. Then the set of nodes which actually carry the interface is simply the set of all nodes of edges intersected by Γ_h:

$$\mathcal{N}_\Gamma = \{n \in \mathcal{N}_h, \exists m \in \mathcal{N}_h, \exists K \in \mathcal{T}_h, n \in K, m \in K, \phi(\boldsymbol{x}_n)\,\phi(\boldsymbol{x}_m) \leqslant 0\}. \tag{8}$$

The set of elements containing such edges is also important:

$$\mathcal{T}_\Gamma = \{K \in \mathcal{T}_h, \exists \boldsymbol{x} \in K, \exists \boldsymbol{y} \in K, \phi(\boldsymbol{x})\,\phi(\boldsymbol{y}) \leqslant 0\}. \tag{9}$$

Similar definitions can be stated for other approximation methods (FD, FV, DG, etc).

2.1. Direct methods

Given the definition of the LS function in equation (1), it seems straightforward to recompute distances directly in order to achieve reinitialization. While it is technically simple when a first-order interpolation is used for the LS function, several improvements are possible to reduce computation times. With a higher-order interpolation, the implementation becomes more complex. Mass preservation, finally, can also be integrated into a DRM. These different aspects are reviewed in the sequel.

2.1.1. Basic first-order methods. The authors of [91] used a DRM for unstructured FE meshes composed of linear triangular elements in two-dimensions (2D). It should be noted that they used a narrow band LS method which consisted in using equation (1) only for discretization points that were close enough to the interface, and truncating the LS function elsewhere. Their approach started by classifying nodes of the FE mesh as layer zero if they belonged to elements of $\mathcal{T}_\Gamma$. They then recursively classified nodes as layer $L+1$ if they belonged to elements which shared nodes with elements of layer L. They chose to compute distances only

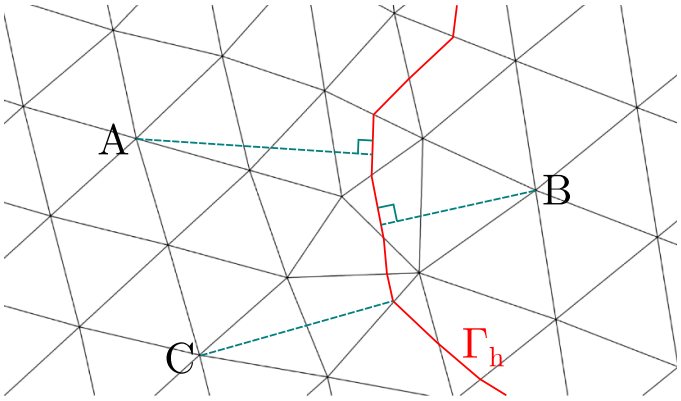

Figure 1. LS reinitialization with a DRM and a linear approximation showing the interface Γ_h in red, and the closest-point projections of three discretization points in blue: the projection is orthogonal for A and B, but it falls onto a line segment extremity for C.

for nodes up to layer four. They also assembled and solved the LS transport equation only for these nodes in order to reduce the computational cost.

In [135], they compared their method with a hyperbolic reinitialization method (HRM), and found that it could be more accurate and faster on both academic problems and two-phase flow problems. The same researchers extended their developments to the three-dimensional (3D) case of linear tetrahedral elements [92]. In 2D, indeed, Γ_h can be reconstructed as a set of line segments. It is easy, therefore, to project each mesh node up to layer four on the closest line segment to find the correct distance value. It should be mentioned that, for a given node, the closest point of the closest line segment might be one of the segment's extremities, and not necessarily an orthogonal projection [91]. This is illustrated in figure 1. Once the distance is computed, the sign can be copied from the input non-reinitialized LS function: $\hat{\phi}_h(\boldsymbol{x}) = \text{sign}\,(\phi_h(\boldsymbol{x}))\,\text{dist}(\boldsymbol{x}, \Gamma_h)$. In 3D, the intersection between Γ_h and a tetrahedron can either be a triangle or a quadrangle, so both cases should be dealt with, and computing distances becomes more technical [92]. More applications of this DRM to two-phase flows were presented in [94].

An important aspect mentioned in [92] is that the narrow band should be large enough to contain all points of Ω_ϵ. Since they defined ϵ as twice the element size, they found that including four layers (i.e. 2ϵ) in the narrow band provided similar results, while with three layers they observed perturbations of the interface.

In [93], the same authors improved their numerical framework again by integrating adaptive mesh refinement. It is interesting, moreover, that they switched to a second-order FE interpolation for the velocity, hence increasing the order as compared to their previous works, but they kept a first-order FE interpolation for the LS function. A similar approach with quadrangular elements was used in [19] for the 2D case. A possible reason is the difficulty of reconstructing Γ_h and computing distances for higher-order LS functions. They tackled this issue in [95], which is presented in section 2.1.3.

Similarly, other researchers used a DRM with a piece-wise linear interface reconstruction and a narrow band approach [61, 97]. Extension to unstructured polyhedral meshes has also been considered in the literature [60].

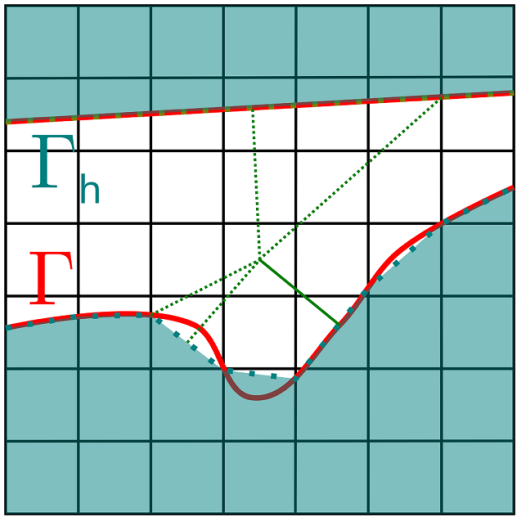

Figure 2. Ray-tracing technique for directly computing distances on a Cartesian grid with a 7×7 cells stencil.

2.1.2. Space partitioning techniques. DRMs entail a significant computational cost, even with a narrow band approach. As shown in figure 1, the problem is that each discretization point should be projected onto all segments. Computational complexity is hence of $\mathcal{O}(N_p N_s)$, where N_p is the number of discretization points in the narrow band, and N_s the number of segments resulting from the reconstruction of Γ_h. Several researchers have proposed solutions to attenuate this complexity.

2.1.2.1. Cartesian grids. For Cartesian grids [11], it is possible to define a stencil of a few neighboring cells around each discretization point, and compute distances only for interface segments lying in those cells. One has to be careful to choose a stencil that is large enough so that reinitialization is ensured for all points of Ω_ϵ. The authors of [11] defined ϵ as $\frac{3}{2}$ times the spatial discretization step, while they used a stencil of 7×7 cells (i.e. $\frac{7}{3}\epsilon$) for LS reinitialization. This stencil is illustrated in figure 2.

Another possibility for Cartesian grids with a piece-wise linear interface reconstruction is the ray-tracing algorithm proposed in [31]. This algorithm fixed LS function values of cells of $\mathcal{T}_\Gamma$. For each of these cells, the normal vector to the interface segment was computed. LS function values at discretization points aligned with this normal vector were then reinitialized by direct distance computation. Distances were updated based on the smallest one (a discretization point may be aligned with multiple normal vectors). Applications to academic and two-phase flow problems showed the efficiency of the proposed approach in terms of computation time as compared to an HRM. This was confirmed by another study applying this method to liquid composite molding [117]. The authors acknowledged, however, that the extension of their method to higher-order accuracy was technically complex, and that another reinitialization method was necessary for cells of $\mathcal{T}_\Gamma$ [31].

2.1.2.2. K-d tree. Some strategies from section 2.1.2.1 could theoretically be implemented for unstructured grids, but their generalization seems difficult, especially when adapted or

anisotropic meshes are used. A more general space partitioning technique has been proposed in [124] to efficiently compute distances and also restrict these computations to the narrow band.

Instead of defining the narrow band as a number of layers, it was defined as a length. The authors of [124], in fact, used automatic anisotropic FE mesh refinement close to the interface with a different prescribed mesh size depending on the orientation with respect to the interface (i.e. orthogonal or tangent). They hence defined the thickness of the narrow band as 20 times the mesh size prescribed in the orthogonal direction.

For each node of the FE mesh, they searched for the closest interface segment using a parallel k-d tree space partitioning technique. Due to their distributed parallel computing paradigm, the FE mesh was first partitioned among the different processes, where each process only owned and knew about a restricted number of elements of the mesh. Then, during reinitialization:

- Each process only reconstructed the part of Γ_h which corresponded to intersections of Γ_h with its own elements.
- Interface segments were organized in a k-d tree where, at the root, a partitioning plane was defined in order to divide the segments into two equally sized branches. For each of these branches, a new partitioning plane with a different orientation was defined to redivide the remaining segments into two branches. When, for some branch, no partitioning plane could be found (e.g. only one segment in the branch), then this branch was changed to a leaf. This is illustrated in figure 3(a).
- For a given point of the mesh, the k-d tree's structure could be exploited to optimize the search for the closest segment. At the bifurcation between two branches, it was indeed possible to decide based on the distance to the partitioning plane to browse only one of the branches. Comparing the distance to the partitioning plane with the narrow band's thickness was also helpful to eliminate points of the mesh which were out of the narrow band. This is illustrated in figure 3(b). The authors of [124] report a computational complexity of searches reduced to $\mathcal{O}(N_p \log(N_s))$.
- Due to the partitioning of the mesh among multiple processes, a bounding box technique with an iterative ring topology was used. Each process computed a bounding box enclosing all its interface segments. After computing their distances locally for their own mesh nodes and their own interface segments organized in a local k-d tree, processes were positioned in a ring topology. Each process interrogated its successor in the ring with search queries for its own mesh nodes. Many queries could be eliminated for partitions which were too far from each other based on distances to bounding boxes. For remaining queries, the successor interrogated its local k-d tree and returned the resulting distances. The ring topology was iteratively changed to ensure each process could interrogate all other processes.

The k-d tree space partitioning technique's performance was assessed for various academic and material science problems, including grain and void growth [124]. LS reinitialization was observed to be up to 20 times faster as compared to a DRM without partitioning. The proposed method was also compared to an HRM, with results showing superior accuracy and inferior computation times. The parallel implementation's efficiency, however, revealed to be highly dependent on mesh partitioning. Because the latter did not consider the interface but only the volume mesh, uneven partitions (e.g. a few partitions owning all interface segments while others do not own any) led to a poor scalability.

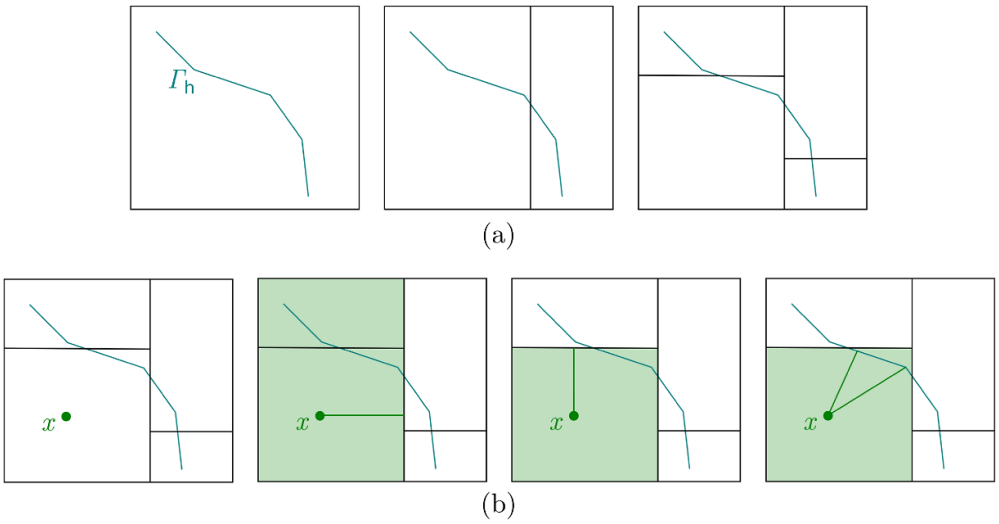

Figure 3. Example of space partitioning tree construction (a), and distance computation (b).

2.1.2.3. Oc-tree. A space partitioning technique was also presented in [181]. The authors chose oc-trees (quad-trees in 2D), which differ a little from k-d trees but also lead to a computational complexity of searches reduced to $\mathcal{O}(N_p \log(N_s))$. They reconstructed, moreover, an actual surface mesh of Γ_h. In most DRMs, indeed, Γ_h is only reconstructed as a set of independent interface segments. In 2D, for instance, no effort is usually put into detecting common extremities between line segments. A triangulation was implemented in [181] in order to construct a surface mesh where nodes are intersections between the volume mesh of Ω and the interface Γ, while elements are interface segments.

A normal smoothing technique was then proposed to compute normal vectors at nodes of the surface mesh, as shown in figure 4. These surface nodes were placed in the oc-tree instead of interface segments. Distances were hence computed by looking for the closest surface node, and then only interface segments containing this node were considered. The closest projection among these interface segments was used for distance computation. The smoothed normal vector defined at the surface node was used for all projections. The authors of [181] reconstructed this actual surface mesh and implemented this normal vector smoothing technique because it was helpful elsewhere in their physical model. They did not analyze the effect of this smoothing on reinitialization accuracy.

2.1.3. Higher-order methods. The generalization of DRMs to higher-order interpolation of the LS function is not straightforward. In a first-order 2D approach, for instance, Γ_h can be reconstructed as a set of line segments $\mathcal{S}_\Gamma, \bigcup_{S\in\mathcal{S}_\Gamma} S = \Gamma_h$, and distances can be computed by looking for the closest line segment: $\text{dist}(\boldsymbol{x}, \Gamma_h) = \min_{S\in\mathcal{S}_\Gamma} \text{dist}(\boldsymbol{x}, \mathcal{S}_\Gamma)$. Space partitioning techniques are also easier to implement because the bounding box of a line segment or a triangle is easy to compute. When higher-order interpolation is used, the intersection of Γ_h with each cell is not a line segment anymore but a complex curve which is quite difficult to reconstruct in a parametric form, as illustrated in figure 5. Without such a form, computing distances becomes challenging.

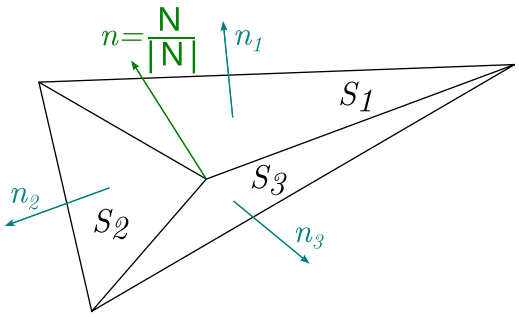

Figure 4. Smoothing technique to compute normal vectors at surface mesh nodes, with $N = \sum_{i=1}^{3} S_i n_i$.

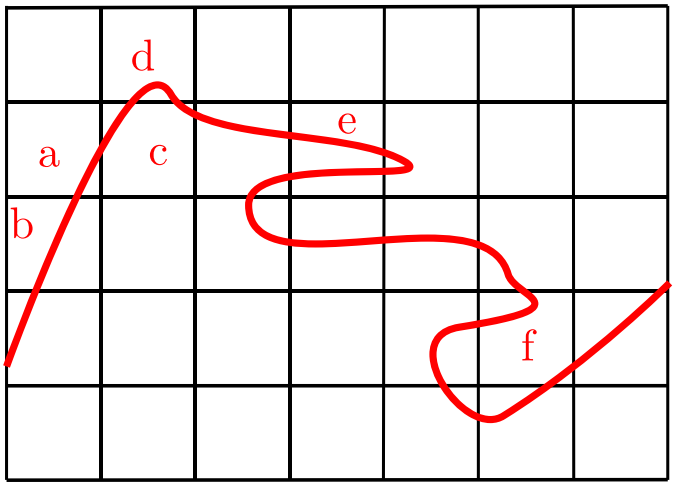

Figure 5. Example of higher-order intersections of Γ with a cell: (a) and (b) are simple cases because there are only two intersections of Γ with the cell's boundaries, (d) is similar but the same cell boundary is intersected twice, (c), (e) and (f) are much more complex because there are two interface segments within the same cell.

2.1.3.1. Quadratic interpolation. The lowest level of higher-order interpolation of the LS function is certainly second-order FE interpolation. It has been used in 2D with triangles [28, 123], and in 3D with tetrahedra [123]. Both [28, 123] used the same strategy consisting in subdividing each quadratic element into smaller linear elements, hence approximating the quadratic interface segment into smaller linear interface sub-segments. This was done by directly creating all sub-elements and then looking for intersections [123], or by subdividing recursively only sub-elements which actually contained a portion of the interface [28], as illustrated in figure 6.

Because of the subdivision technique, the number of interface sub-elements could be quite large and drastically increase the computational cost of LS reinitialization as compared to the linear case. Although none of [28, 123] used space partitioning, a GPU-accelerated parallel algorithm was proposed in [123] to reduce the computation time, along with a narrow band

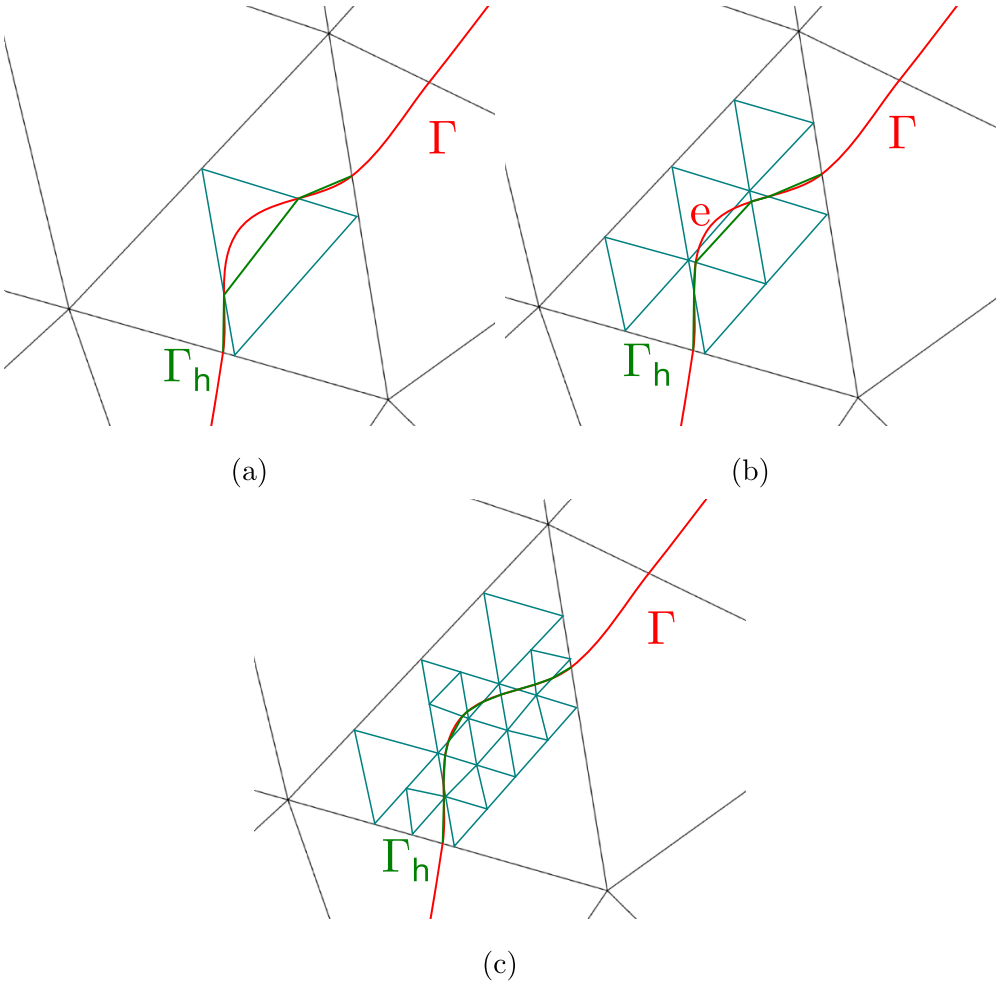

Figure 6. Example of subdivision of an element to compute distances showing the interface Γ in red, the sub-elements in blue and the linear interface sub-elements in green: (a) first sub-division, (b) second sub-division, (c) third sub-division. With the second sub-division, there is a portion of the interface $e \subset \Gamma$ which does not lead to an interface sub-element because a linear algorithm would only consider edges with opposite LS function signs at its extremities.

approach. This study showed, moreover, that optimal convergence rates could be obtained for two-phase flow simulations, especially using automatic anisotropic mesh refinement.

2.1.3.2. Cubic interpolation. Cubic Hermite FE interpolation can also be used to achieve higher-order accuracy [68]. This particular type of FE interpolation consists in storing the value of the LS function and its gradient at each discretization point. The authors of [68] combined a finite difference (FD) method with cubic Hermite FE interpolation for the LS function. To compute the distance from each discretization point of the narrow band to the interface, they used Newton's method. The objective was to find, for each point $\boldsymbol{x}$ of the narrow band, the

projection $\boldsymbol{y}$ such that:

$$\begin{cases} \phi_h(\boldsymbol{y}) = 0, \\ \nabla\phi_h(\boldsymbol{y}).(\boldsymbol{y}-\boldsymbol{x}) = 0, \end{cases} \tag{10}$$

where the first equation enforces the projection to lie on the interface, and the second equation enforces the projection to be orthogonal. To solve this system of equations with Newton's method and find the closest point, it was necessary to provide a very good initial guess. The authors of [68] provided as initial guess a linear approximation of the closest point. They acknowledged, however, that they encountered some convergence issues, especially near small features (e.g. a discretization point with a positive LS value but negative values at all neighboring points), and that more research was required to solve them. A pre-processing operation was introduced later to eliminate these issues [154].

2.1.3.3. Polynomial reconstruction. The combined FD–FE framework proposed in [68] was quite unique in the sense that the LS function and its gradient were maintained throughout the simulation by solving transport equations for both of them. Other studies introducing a higher-order polynomial interpolation of the LS function into an FD method consisted in transporting only the LS function with an FD scheme, and reconstructing a polynomial interpolation whenever LS reinitialization was required [49, 95, 108, 172]. Many studies, indeed, relied on higher-order FD schemes for solving the LS transport equation. Higher-order accuracy, nevertheless, cannot be achieved if these schemes are combined with first-order DRMs as in section 2.1.1.

In [108], the authors proposed to start with a first-order search (i.e. assuming piece-wise linear interface reconstruction) to find not only the closest interface segment, but also the second and third closest ones (figures 7(a) and (b)). They then used the projection points and the normal vectors from each of these segments to reconstruct a higher-order polynomial approximation of the interface (figures 7(c) and (d)). There are, of course, a few particular cases that should be dealt with, for instance to ensure that all interface segments found during the search are contiguous. Treatments for these particular cases for both 2D and 3D grids can be found in [108]. Through simulations on academic and two-phase flow problems, they showed that their method could converge in L^1 norm with a third-order accuracy regarding the LS function, second-order accuracy regarding normal vectors to the interface, and first-order accuracy regarding curvatures.

The authors of [108] also compared their higher-order method with the first-order method (section 2.1.1), and with an even simpler method consisting in using the distance to the closest intersection point between Γ_h and grid edges. They showed that the point method did not provide accurate results, especially regarding curvatures computation. There was hence a great improvement when switching to the first-order method. Going for the higher-order method, however, did not improve the convergence order, and improved accuracy only slightly [108].

The procedure developed to compute distances to a higher-order reconstruction of Γ_h was not detailed in [108]. A piece-wise higher-order polynomial approximation of Γ_h was proposed in [172]. A stencil including neighboring cells was created around each cell of $\mathcal{T}_\Gamma$. Intersections of Γ_h with edges of this stencil were used to fit a higher-order polynomial approximation of Γ_h by solving a least-squares problem. The stencil size was chosen appropriately depending on the order of the polynomial approximation and hence the number of coefficients to fit. The cell itself was then subdivided into a sub-grid, and the center of each cell of the sub-grid was projected onto the higher-order interface approximation using Newton's method. Finally, when reinitializing the LS function for some discretization point $\boldsymbol{x}$, the distance was computed by

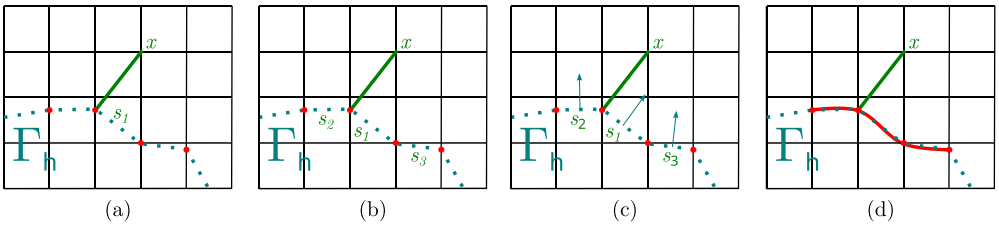

Figure 7. Higher-order polynomial approximation of the interface: (a) search of closest interface segment s_1 to a point $\boldsymbol{x}$, (b) identification of second s_2 and third s_3 closest interface segments, (c) normal vectors from each interface segment and (d) polynomial interface approximation.

first looking for the closest sub-grid projection point using a k-d tree, and then using this point as initial guess for an iterative distance computation algorithm relying on Newton's method. Similarly to [68], indeed, the authors of [172] encountered convergence issues depending on the initial guess and the local regularity of the LS function. They presented, moreover, alternative forms of equation (10):

$$\min_{\boldsymbol{y}\in\Gamma_h} (\boldsymbol{y}-\boldsymbol{x})^2, \tag{11}$$

or, using the method of Lagrange multipliers:

$$\min_{\boldsymbol{y}\in\Omega} \frac{1}{2} (\boldsymbol{y}-\boldsymbol{x})^2 + \lambda p_0(\boldsymbol{x}), \tag{12}$$

where p_0 is the second-order polynomial approximating Γ_h within the cell containing $\boldsymbol{y}$. Through applications on academic problems, they showed that they could obtain a convergence order in L^1 norm of $q+1$ for the LS function and an order of $q-1$ for curvatures when using a polynomial approximation of order $q = 2,3,4$.

An extension of this approach relying on a polynomial approximation to uniform unstructured triangular meshes can be found in [95]. The authors used the concept of layers introduced in section 2.1.1 to get enough points for the solution of the least-squares problem. Instead of creating a sub-grid for initial guesses, they proposed to reconstruct a linear approximation of Γ_h, sample points on the subsequent line segment, and then project these points on the higher-order approximation. They used Newton's method as well for this projection. Similarly as in [172], they implemented distance computation into two steps: first a search for the closest sample point projection using a k-d tree, and then Newton iterations from this initial guess to get the correct distance to the higher-order approximation. Similarly as in [172], they proved that they could increase the convergence order on LS function values, normal vectors and curvatures using higher-order polynomials.

An important issue raised in both [95, 172] is that there are some particular cases where it is difficult to reconstruct a polynomial approximation: two discontinuous portions of the interface crossing the polynomial fitting stencil (figure 8(a)), or the interface folding on itself within the stencil (figure 8(b)). Although specific treatments for these issues can be found in these studies, the authors of [95] acknowledged that they did not deal with all particular cases, and they did not extend their developments to 3D. A thorough study of these issues and how they can be solved for structured grids in 2D and 3D can be found in [49].

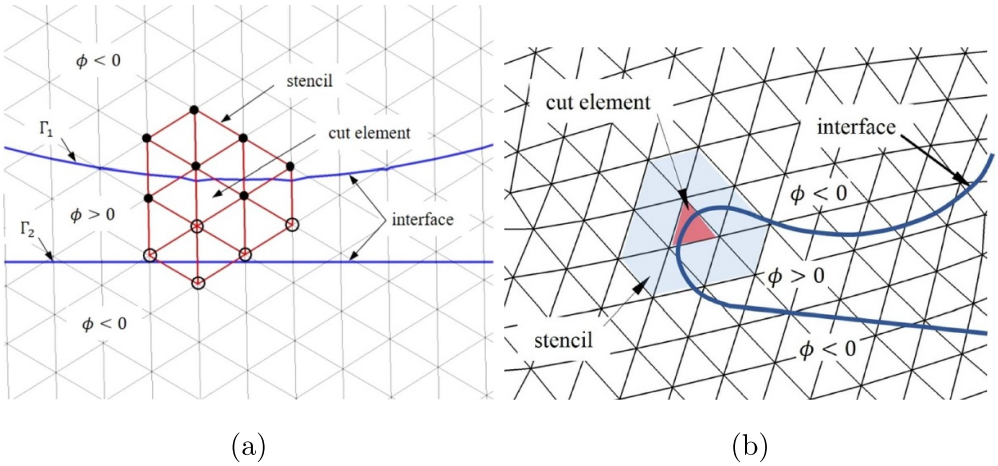

Figure 8. Polynomial reconstruction of the interface in singular regions: (a) two interface portions crossing the least squares fitting stencil, (b) interface folding on itself within the least squares fitting stencil. Reprinted from [95], Copyright (2022), with permission from Elsevier.

2.1.3.4. Discontinuous Galerkin (DG) method. The DG method is an interesting approach to improve convergence orders on LS function values, normal vectors and curvatures. Similarly to the FE method, and as opposed to the FD method, the DG method has a built-in polynomial approximation of the LS function. The difference with the FE method is that each cell has its own polynomial approximation, and there is no built-in continuity between neighboring cells. Similarly to the approach presented in [95, 108, 172], direct LS reinitialization can be implemented in a higher-order DG method using sampling points search with a k-d tree to get a good initial guess for a distance computation using Newton's method [120, 121]. The main difference is that there is no need to reconstruct the polynomial approximation on some stencil by solving a least-squares problem, because this approximation is built-in with the DG method. All issues related to this step with folded interfaces and so on are hence avoided. Other issues arise, however, because interface reconstructions do not respect the continuity of Γ_h, which requires some treatments [120].

2.1.4. Mass preservation. A well-known issue with all LS reinitialization methods is that they might slightly shift the position of the interface, hence leading to mass change errors, as shown in figure 9. An algorithm can be found in [89] to attenuate these errors within a linear FE method. The authors proposed to track local mass changes during direct reinitialization. A mass correction value was then defined at each node of the FE mesh, and direct reinitialization was finally operated with ensured global mass conservation. They proved the accuracy of their method in terms of mass change on both academic and two-phase flow problems with a narrow band approach [89].

2.1.5. Summary. DRMs consist in directly projecting each discretization point onto a cell-wise or element-wise reconstruction of the interface. This reconstruction can be explicitly performed if first-order interpolation is used for the LS function, but not for higher-order interpolation.

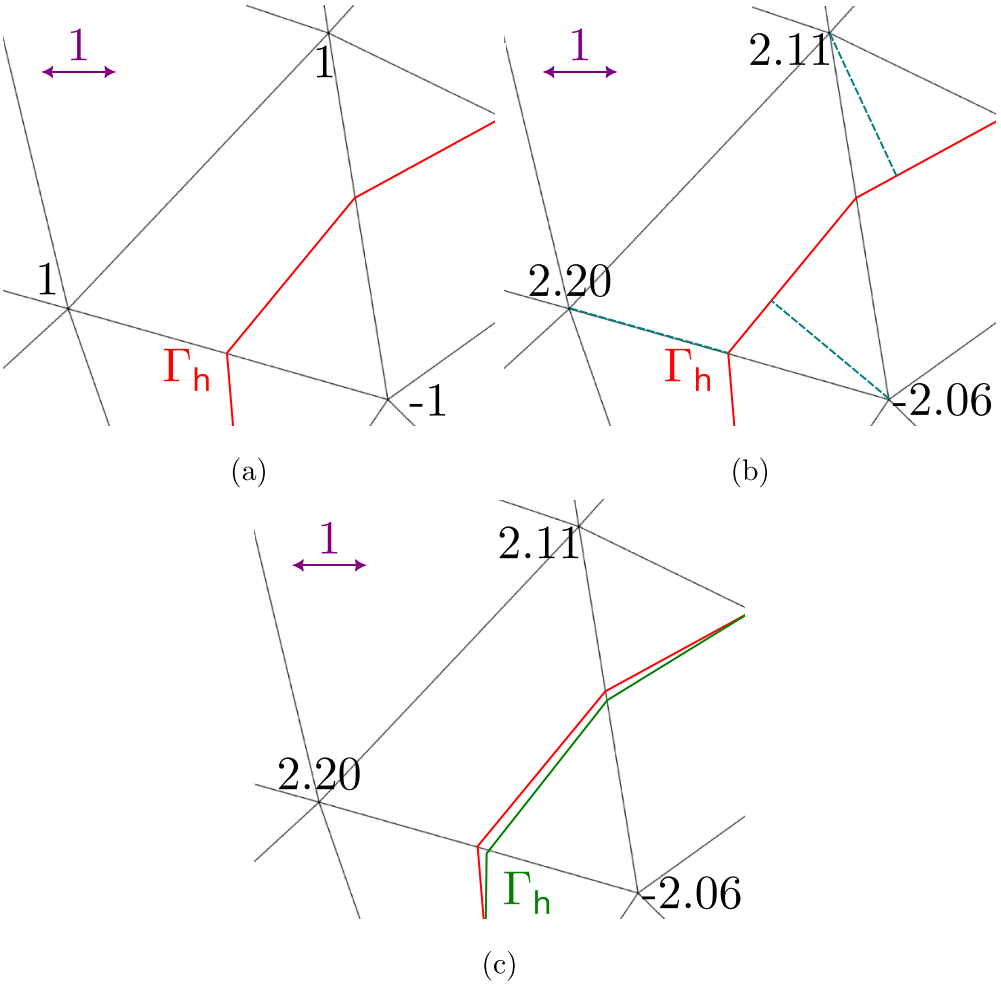

Figure 9. Example of mass change error due to LS reinitialization with a DRM and a linear approximation showing the initial interface Γ_h in red and the length scale in purple: (a) LS function values at three discretization points with a balance so that the interface always intersects the element's edges at their middles even though the distance property is not respected, (b) LS reinitialization with the DRM and correction of distance values with projections shown in blue, (c) new interface in green with a spurious shifting due to reinitialization because values on each intersected edge are no longer balanced.

For second-order interpolation, element sub-division has been proposed to approximate quadratic interface segments through smaller linear interface sub-segments and ease distance computation. For interpolation of order higher than two, distances have been computed with a local nonlinear solver such as Newton's algorithm. Difficulties have been mentioned for the convergence of this algorithm depending on the provided initial guess and the regularity of the interface. With FE and DG methods, moreover, higher-order polynomial approximation can be built-in, while with FD methods a least-squares fitting step is necessary. This fitting step also raises difficulties depending on the regularity of the interface.

For each discretization point, the computational cost of looking for the projection onto the closest interface segment is significant. Two ingredients can be combined to attenuate this cost. LS reinitialization can be conducted only in a narrow band of discretization points located close enough to the interface. Space partitioning techniques, in addition, can help reduce the computational complexity of closest-point searches.

The parallel implementation of DRMs has received less attention in the last decade. A few studies, nevertheless, have shown that it is possible on both GPUs and distributed computing environments. Combined with a narrow band approach and space partitioning, parallel computing can help reduce drastically the computational cost of a DRM.

The accuracy of these DRMs has been demonstrated in terms of errors on LS function values, normal vectors and curvatures. It has been shown to be improved with automatic adaptive refinement on unstructured meshes and higher-order interpolation. These results have been obtained mostly on academic and two-phase flow problems, although some studies have considered problems related to microstructure evolution.

2.2. Local indirect methods

Local indirect methods do not directly attempt to compute distances but instead focus on locally restoring the unit norm of the LS function gradient by solving equation (7). Because equation (7) involves the initial condition $\hat{\phi} = 0$ on Γ, local indirect methods require an initialization step for discretization points of $\mathcal{N}_\Gamma$. This initialization can be achieved by a DRM. LS values for remaining points, meanwhile, are computed by solving a local problem. For a structured 2D grid of discretization steps $(\Delta x, \Delta y)$, this problem can be formulated as [109, 161]:

$$\max\left(D_{i,j}^{-x}, -D_{i,j}^{+x}, 0\right)^2 + \max\left(D_{i,j}^{-y}, -D_{i,j}^{+y}, 0\right)^2 = 1, \tag{13}$$

with first-order forward and backward FD operators:

$$D_{i,j}^{-x} = \frac{\hat{\phi}_{i,j} - \hat{\phi}_{i-1,j}}{\Delta x}, D_{ij}^{+x} = \frac{\hat{\phi}_{i+1,j} - \hat{\phi}_{i,j}}{\Delta x}, D_{i,j}^{-y} = \frac{\hat{\phi}_{i,j} - \hat{\phi}_{i,j-1}}{\Delta y}, D_{ij}^{+y} = \frac{\hat{\phi}_{i,j+1} - \hat{\phi}_{i,j}}{\Delta y}. \tag{14}$$

Here $\hat{\phi}_{i,j}$ is the unknown LS function value at the center of cell (i,j) that can be obtained by solving equation (13). This value will depend on LS values at neighboring cells which might be on the left, right, bottom or top depending on the direction of the LS function gradient. It is important, indeed, to guarantee propagation only in the upwind direction, which is also called the causality condition.

The well-known fast marching method (FMM) builds upon the causality condition to classify and sort cells in a specific order to improve the efficiency. This sorting is removed in the fast iterative method (FIM) in order to ease parallel implementations, while alternative directions are combined in the fast sweeping method (FSM) to reduce the computational complexity. This is reviewed in the sequel.

2.2.1. FMM. In the FMM, the local problem in equation (13) is solved in a specific order. As described in [161], discretization points which have been reinitialized during the initialization step, for instance using a DRM, are tagged as *KNOWN*, while their neighbors (discretization points belonging to cells which also contain a *KNOWN* point) are tagged as *BAND*. Remaining points are tagged as *FAR*. All *BAND* points are subjected to a first reinitialization by solving

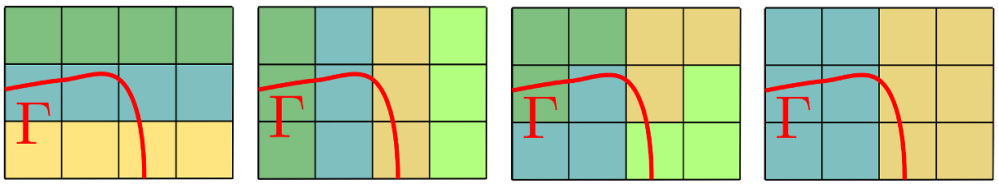

Figure 10. Parallel partitioning strategies, with cells colored by partition: (a) and (b) interface ignored during partitioning, (c) and (d) interface considered during partitioning in order to improve load balancing.

equation (13). For *FAR* points as well *BAND* points for which no solution to equation (13) could be found, for instance if the causality condition could not be ensured, the distance value is set by default to be very large ($+\infty$).

The iterative part of the FMM consists, at each iteration, in reinitializing all neighbors of the *BAND* point with smallest current distance value by solving equation (13). This chosen *BAND* point's tag is switched to *KNOWN*. Tags of neighbors for which a smaller distance value (as compared to the current one) could be found are switched to *BAND*. To efficiently find the point with smallest distance value, a specific data structure such as a heap is often used, which ensures a computational complexity of $\mathcal{O}(N_p \log(N_p))$. The algorithm terminates when the heap is empty and all points are *KNOWN*.

2.2.1.1. Oc-tree grids. An extension of the FMM to oc-tree grids (locally refined Cartesian grids) has been used in [96]. Different DRMs were proposed for the initialization step, including first-order and higher-order ones. Academic and two-phase flow problems were considered.

2.2.1.2. Parallel implementation. A parallel implementation of the FMM was proposed in [161] for Cartesian grids. The authors chose to use the analytical expression of the LS function for initialization, because their study focused on the distributed parallel implementation of the FMM within a narrow band. This method is quite complex and is not detailed herein. The main challenge that was tackled by the authors was to find a suitable replacement for the sequential heap data structure used by the FMM. They proposed appropriate data structures and synchronization operations. They demonstrated the parallel efficiency of their algorithm on academic problems featuring up to billions of discretization points partitioned among up to 65 536 processes. Similarly as the parallel DRM approach reviewed in section 2.1.2.2, they used a partitioning which ignored the interface. Consequently, they also noticed that the parallel efficiency was deteriorated when a few partitions contained the whole narrow band, while other partitions were too far from it.

This challenge was addressed by the shared memory parallel implementation of the FMM proposed in [109]. The authors proposed to improve load balancing during LS reinitialization by considering the interface during partitioning, as shown in figure 10. This reduced the risk of some threads remaining idle while others were still working. They demonstrated the parallel efficiency of their approach up to 24 threads for academic problems solved on Cartesian grids. Similarly as [161], they did not address the initialization step.

2.2.1.3. Higher-order accuracy. A second-order FMM for Cartesian grids has been used in [4]. The authors used a larger stencil for the computation of derivatives, where the formulas in equations (13) and (14) were extended to more neighboring cells.

Initialization of the FMM for second-order accuracy was also addressed [4]. A DRM similar to those reviewed in section 2.1.3.3 was used for cells of $\mathcal{T}_\Gamma$: a higher-order polynomial approximation was first reconstructed, and then distances were computed directly using a nonlinear iterative scheme with good initial guesses. Applications to two-phase flows were presented.

A second-order FMM was used in [27] as well. This study mainly addressed the initialization step and its extension to quadratic FE interpolation on unstructured grids. For the marching step itself, the authors relied on element subdivision and the linear FMM for each sub-element, which resembles some DRMs for quadratic FE interpolation reviewed in section 2.1.3.1. They demonstrated second-order of convergence on academic problems with triangular an tetrahedral meshes featuring local refinement.

2.2.2. FIM. The whole point of the FIM as compared to the FMM is to ease the parallel implementation. It is not surprising, therefore, that the only paper on the FIM fitting this review's scope is about the parallel scalability of a shared memory implementation for Cartesian grids [150]. As detailed in this study, the main ingredient in the FIM which eases the parallel implementation as compared to the FMM is the absence of a sorted queue. The algorithm proposed in [150] consisted in reinitializing the LS function layer by layer from the interface. All discretization points within the same layer could be dealt with independently in parallel using the linear kernel in equations (13) and (14). Results on 2D and 3D academic problems involving millions of discretization points demonstrated a reasonable parallel efficiency up to 12 threads.

2.2.3. FSM. The FSM is an improvement over the FMM in the sense that it reduces the best-case computational complexity to $\mathcal{O}(N_p)$ for Cartesian grids. The different *KNOWN*, *BAND* and *FAR* tags and the heap data structure are replaced by a sweeping algorithm. After the initialization step, in the 2D case, distances for all non-initialized points are set to a very large value. Then, grid points are swept in four alternating directions: top left corner to bottom right corner, top right to bottom left, bottom right to top left, and finally bottom left to top right. Every time, the considered grid point's distance is updated if a solution to equation (13) can be found. Grid sweeping in the four directions is repeated until LS function values converge between two consecutive iterations.

Higher-order extensions of the FSM were considered in [74]. The authors mentioned that these extensions often relied on larger stencils resembling those reviewed in section 2.2.1.3, which had the drawback of introducing a dependence of the number of sweepings for convergence on the number of grid points. They introduced a new second-order method which, as demonstrated by their results on academic problems, did not have this drawback.

They could not establish whether their method could be extended to an order higher than two [74]. To increase the order of accuracy, therefore, they chose to combine their second-order FSM with a third-order HRM. Discretization points in a narrow band of five cells were reinitialized with the HRM in order to get first-order convergence on curvatures, while the distance property was restored away from the interface using the second-order FSM. They mentioned, moreover, that the HRM could incorporate a specific treatment to eliminate spurious motions of the interface during reinitialization and improve mass conservation [74].

2.2.4. Hopf-Lax formula. A new local indirect approach based on the Hopf–Lax formula has been proposed in the last decade [67]. This approach could be classified as a global indirect method because it consists in solving the hyperbolic equation in equation (15). The exact solution, however, is expressed locally as the minimum of a functional which can be found using a so-called split Bregman solver. This approach could finally be classified as a direct method as well because, at convergence, the solution of the minimization problem also contains an approximation of the projection of each grid node onto the interface.

The convergence of this iterative solver was observed to be sensitive to the initial guess. In [67], the authors relied on an HRM to obtain the initial guess. Their Hopf-Lax formula solver was shown to lead to errors below $10^{-3}\%$ and a convergence rate up to five, with computation times reduced by a factor of ten as compared to a higher-order FD HRM on a Cartesian grid.

This Hopf-Lax formula-based approach was improved later on [113]: a gradient-based solver was proposed instead of the split Bregman solver, a narrow band method was developed, as well as a parallel implementation on a GPU. They demonstrated an optimal scaling for sufficiently large grids.

2.2.5. Summary. Indirect local methods consist in solving the Eikonal equation using an FD stencil locally and progressively, cell by cell. Distance values at discretization points can only be computed if the causality condition is ensured. The FMM exploits this condition by propagating in the upwind direction through classification and sorting of discretization points in a specific manner. Even though this requires a sequential heap data structure, both distributed and shared memory parallel implementations of the FMM have been developed in the last decade, showing interesting parallel efficiencies. One study, in particular, addressed the issue of load balancing when only a few partitions contain the interface. Additional gains in computation time have been obtained by implementing narrow band FMMs.

Extensions to adapted grids and higher-order FD stencils have also been considered, though only for Cartesian grids. The only study considering unstructured grids focused only on the initialization step, which relied on a second-order DRM. No study on parallel implementation of the FMM for adapted, unstructured or higher-order discretizations was proposed.

The FIM removes the heap data structure from the FMM in order to ease the parallel implementation. This was shown by a study in the last decade where a shared memory parallel algorithm was proposed. The FSM also removes the heap data structure, but replaces it by an iterative sweeping algorithm. Difficulties on extending the FSM to higher-order accuracy without increasing significantly the number of sweeping iterations were discussed in a study. The latter introduced a combined method relying on an HRM to reach third-order accuracy in the narrow band. It should be noted, once again, that both FIM and FSM studies were restricted to Cartesian grids. Recent developments based on the Hopf-Lax formula are promising in reaching higher-order accuracy with optimal scaling and reduced computation times, but they are also restricted to Cartesian grids.

Most studies have addressed only academic problems with static or moving interfaces. One study applied the FMM to two-phase flow problems.

2.3. Global indirect methods

As will be shown in this review, global indirect methods are the most widely used LS reinitialization methods, with HRMs being the first choice for most studies. Alternatives to HRMs,

mostly borrowed from the field of image segmentation, include combined hyperbolic reinitialization and LS transport, and other reformulations of equation (7) for instance as a regularized elliptic problem. These different alternatives are reviewed in the sequel.

2.3.1. Hyperbolic reinitialization. Because equation (7) is difficult to solve directly, HRMs as proposed initially in [137] consist in looking for the steady state solution of:

$$\begin{cases} \frac{\partial \hat{\phi}}{\partial \tau} + S(\phi)\left(|\nabla \hat{\phi}| - 1\right) = 0, \text{ in } \Omega \times [0, T_\tau], \\ \hat{\phi}(\tau = 0) = \phi, \text{ in } \Omega, \end{cases} \tag{15}$$

where ϕ is the input LS function and $\hat{\phi}$ is the reinitialized one, while τ is a fictitious time, T_τ is the associated fictitious duration and S is the sign function:

$$S(\phi) = \begin{cases} 1, & \phi > 0, \\ 0, & \phi = 0, \\ -1, & \phi < 0. \end{cases} \tag{16}$$

In the original method of [137], this sign function was replaced by the smeared approximation:

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + h^2}} \tag{17}$$

with h the spatial discretization step. Added to the choice of S, various ways can be found in the literature to:

- define the fictitious time step $\Delta\tau$,
- impose $\hat{\phi} = 0$ on Γ,
- terminate fictitious time iterations.

One particular focus of research is that iterations of equation (15) are well-known to cause interface shifting, and hence mass change errors. The authors of [104] proposed to attenuate these errors by changing the sign function to:

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + (|\nabla \phi| h)^2}}. \tag{18}$$

They found, indeed, that interface shifting in the original method of [137] was due to local sign changes during iterations of equation (15) where ϕ was locally steep.

Mass change errors were also investigated in [136]. They introduced a different sign function:

$$S(\phi) = 2H_\epsilon(\phi) - 1, \tag{19}$$

though their proposition to preserve the interface did not rely on this new sign function but on the introduction of a Lagrange multiplier into equation (15):

$$\begin{cases} \frac{\partial \hat{\phi}}{\partial \tau} + S(\phi)\left(|\nabla \hat{\phi}| - 1\right) - \lambda\left(\phi, \hat{\phi}\right) \frac{\mathrm{d}H_\epsilon}{\mathrm{d}\phi}\left(\hat{\phi}\right) |\nabla \hat{\phi}| = 0, \text{ in } \Omega \times [0, T_\tau], \\ \hat{\phi}(\tau = 0) = \phi, \text{ in } \Omega. \end{cases} \tag{20}$$

This scalar and uniform Lagrange multiplier $\lambda \in \mathbb{R}$ was eliminated in [136] as it could be expressed as:

$$\lambda\left(\phi,\hat{\phi}\right)=\frac{\int_{\Omega}\frac{\mathrm{d}H_{\epsilon}}{\mathrm{d}\phi}\left(\hat{\phi}\right)S\left(\phi\right)\left(|\nabla\hat{\phi}|-1\right)\mathrm{d}\Omega}{\int_{\Omega}\left(\frac{\mathrm{d}H_{\epsilon}}{\mathrm{d}\phi}\left(\hat{\phi}\right)\right)^{2}|\nabla\hat{\phi}|\mathrm{d}\Omega}. \tag{21}$$

Its aim was not to preserve the local position of the interface but the total volume of sub-domain Ω_1, using the fact that $|\Omega_1| \approx \int_{\Omega} H_{\epsilon}(\phi)\mathrm{d}\Omega$. The regularized Heaviside function H_{ϵ} was defined as in equation (2). It should be noted that the HRM defined by equations (19)–(21) does not require this specific choice of regularized Heaviside function. Other forms, relying for instance on the tanh function, can also be found in the literature.

Because of the large number of studies using an HRM in the last decade, tables are used. In each table, when an information could not be found in some study, the corresponding field is left blank.

The *Termination* column indicates the type of criterion used to terminate iterations of the hyperbolic solver. Most authors indicated either no criterion or mentioned that they iterated until steady state, without defining it. Some authors also mentioned convergence or equilibrium, once again with no definition. In all cases, unless an explicit definition was provided, the exact words of the authors are reported in the *Termination* column.

The *Step* column indicates the value used for the fictitious time step. It should be mentioned that while many studies did not provide this information, it might be simply due to the fact that these studies used an explicit discretization for the fictitious time. This value, consequently, might have been determined by a stability criterion.

The *Scheme* columns indicate numerical approximation methods in space and (fictitious) time used for solving the hyperbolic equation. Combinations with other methods (for solving e.g. the LS transport equation in equation (6) or Navier–Stokes equations) have been used by multiple researchers but this is irrelevant to the present review. Only the numerical approximation used for the HRM itself is of interest, and in particular whether higher-order convergence rates in space and time can be obtained. It should be mentioned, moreover, that convergence orders are best case scenarios as they might be deteriorated locally due to upwind schemes or interface preservation constraints.

2.3.1.1. HRM of [137]. The first category of studies directly used the original HRM and hence solved equation (15) with the smeared sign function in equation (17). These studies are listed in table 1.

These studies either did not report the termination criterion or predefined a number of iterations between 10 and 20. This is consistent with the two studies which used a number of steps of the order of $\frac{\epsilon}{h}$, as they indicated that it represented around 15 steps [23, 24].

Only a few studies reported the chosen value for the fictitious time step. They used a value between $0.1h$ and $0.5h$ which, in the case of [152, 153] might be imposed by their explicit Euler time discretization.

Two studies used spatial discretizations of order five with Cartesian grids [70, 75]. All other studies used low order schemes. This might be due to the use of unstructured meshes for [23, 88, 152, 153] as higher-order FD and FV schemes are well-known to be more complex to implement for such meshes. The DG approximation used in [34] is an interesting alternative that could give access to higher-order convergence rates for unstructured meshes, but the authors chose only linear polynomials. Two studies, in addition, used Cartesian grids with local refinement [65, 66], which is another way of improving accuracy close to the interface.

Table 1. Studies using exclusively the HRM defined by equations (15) and (17) as per [137].

Reference	Termination	Step	Space scheme (order)	Time scheme (order)
[75]			FD WENO (5)	TVD RK (2)
[70]			FV WENO (5)	TVD RK (3)
[88]			FD UC (2)	RK (2)
[34]			DG (2)	Euler (1)
[20]	5-10 steps		FD	FD
[84]	20 steps		FD UC (2)	TVD RK (2)
[64]	20 steps		FD UC (2)	TVD RK (2)
[65]	20 steps		FD UC (2)	TVD RK (2)
[66]	20 steps		FD UC (2)	TVD RK (2)
[152]	10 steps	$\mathcal{O}(h)$	FV Upwind (1)	Euler (1)
[153]	10 steps	$\mathcal{O}(h)$	FV Upwind (1)	Euler (1)
[81]	$\mathcal{O}\left(\frac{\epsilon}{h}\right)$ steps	$\mathcal{O}(h)$	FV (2)	Euler (1)
[23]	$\mathcal{O}\left(\frac{\epsilon}{h}\right)$ steps	$\mathcal{O}(h)$	FV Upwind (1)	(1)
[24]	$\mathcal{O}\left(\frac{\epsilon}{h}\right)$ steps	$\mathcal{O}(h)$	FV Upwind (1)	(1)

Regarding time discretizations, TVD RK schemes of order two or three were used in most studies, both with Cartesian grids and unstructured meshes. It should be noted that the chosen convergence order for spatial discretization was always greater or equal to the one chosen for time.

Within the framework of the HRM defined by equations (15) and (17), multiple numerical developments were proposed in the studies referenced in table 1. A multiscale approach was developed in [75] to eliminate under-resolved interface segments, and increase computational efficiency, especially for two-phase flow problems involving the generation of a large number of droplets. FD and FV approximations of the gradient were derived and applied for unstructured grids in [88, 152, 153]. A parallel implementation was proposed in [84] and reused in [64–66], as discussed in section 2.3.6. A reformulation of the reinitialized LS function into a conservative form was proposed in [24] in the aim of reducing mass change errors.

2.3.1.2. HRM of [137] with a different sign function. Studies which opted for a different sign function (excluding those using the sign function in equation (18)) are listed in table 2. Three studies did not define the sign function [17, 69, 119], while two studies used the discontinuous sign function (equation (16)). Liu *et al* [73] did not include any discussion on the choice of this discontinuous sign function. The same choice was made in [170], where the authors introduced an impressive sixth order accurate FD discretization in both space and time for Cartesian grids. They also did not comment or discuss this absence of sign function smearing in their HRM. Remaining studies all used the smeared sign function defined in equation (19).

Remarks on termination criteria, choice of fictitious time step, and discretization schemes are the same as those made for table 1.

Regarding numerical developments, [69] proposed a three-step method to improve gradient preservation when reinitializing the LS function. A reformulation of the reinitialized LS function into a conservative form was proposed in [58] in the aim of reducing mass change errors. GPU acceleration and mesh adaption were combined in [56, 57] to improve computational efficiency and accuracy.

Table 2. Studies using exclusively the HRM defined by equation (15) as per [137], but not the sign functions defined in equations (17) or (18).

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[69]				FD ENO (2)	Gauss-Seidel (1)
[17]				FV Upwind (1)	TVD RK (2)
[119]		$\mathcal{O}\left(\frac{\epsilon}{h}\right)$ steps	$\mathcal{O}(h)$	FD ENO (2)	Euler (1)
[73]	(16)	10 steps	$\mathcal{O}(h)$	FV ENO (2)	RK (2)
[170]	(16)	Equilibrium	$\mathcal{O}(h)$	FD WENO (6)	TVD RK (6)
[58]	(19)			FV UC (2)	RK (4)
[56]	(19)	$2\frac{\epsilon}{h}$ steps		DG (3-6)	MRAB (2)
[57]	(19)	$2\frac{\epsilon}{h}$ steps		DG (3-6)	MRAB (2)
[82]	(19)	$\mathcal{O}\left(\frac{\epsilon}{h}\right)$ steps	$\mathcal{O}(h)$	FV Centered (2)	Euler (1)
[45]	(19)	Steady state		FD WENO (5)	Euler (1)
[151]	(19)	Steady state	$\mathcal{O}(h)$	FD WENO (5)	TVD RK (3)

Table 3. Studies using exclusively the HRM defined by equations (15) and (18) as per [104].

Reference	Termination	Step	Space scheme (order)	Time scheme (order)
Reference	Termination	Step	Space scheme (order)	Time scheme (order)
[46]			FV UC (2)	(2)
[149]			FV WENO (5)	TVD RK (3)
[22]	3 steps	$\mathcal{O}(h)$	FV (1)	
[155]	3 steps	$\mathcal{O}(h)$	FV	TVD RK (3)
[160]	Steady state		FD WENO (3)	TVD RK (3)
[138]	Steady state		FV	

2.3.1.3. HRM of [104]. Studies which relied on the sign function defined in equation (18) to reduce interface shifting are listed in table 3. This particular choice of sign function is discussed only in [46]. The authors mentioned that they chose this sign function to relax changes to $\hat{\phi}$ near the interface, with the hope of preventing interface shifting, and also to ease convergence in a few steps.

It is surprising that [22] only used three iterations for reinitialization. As detailed in their analysis, much more iterations were necessary depending on the level of deterioration of the input LS function. In their experience, nevertheless, since they triggered their HRM at each time step for their two-phase flow simulations, only three fictitious time iterations were enough to maintain a good accuracy in the narrow band. They mentioned, in fact, that more iterations would have been necessary in order to restore the distance property in the whole domain [22].

Numerical developments in the past decade include a detailed calculation of the $|\nabla\phi|$ term in equation (18) for unstructured meshes, and a reformulation of the reinitialized LS function into a conservative form to reduce mass change errors [149].

2.3.1.4. HRM of [136]. As shown in table 4, the HRM with a volume conservation constraint as per equations (20) and (21) has been used in many studies. Most of them used it as is. Three studies [32, 33, 171], however, implemented the volume conservation constraint in a weak sense, which means LS reinitialization was performed using equation (15) with no

Table 4. Studies using exclusively the HRM defined by equations (20) and (21) as per [136], though the sign and regularized Heaviside functions might differ.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[33]	(17)		$\mathcal{O}(\Delta t)$	DG (2)	RK (2)
[32]	(17)			DG	Euler (1)
[171]	(17)			FV WENO (5)	TVD RK (3)
[164]	(19)			FD WENO (5)	TVD RK (2)
[162]	(19)			FD WENO (5)	TVD RK (3)
[166]	(19)			FD Upwind	RK (2)
[165]	(19)			FD Upwind	RK (2)
[9]	(19)			FD WENO (5)	TVD RK (3)
[131]	(19)		$\mathcal{O}(h)$	FV WENO (5)	TVD RK (2)
[72]	(19)		$\mathcal{O}(h)$	CVFE P1 (2)	Euler (1)
[134]	(19)	2 steps	$\mathcal{O}(\Delta t)$	FD (2)	(2)
[132]	(19)	Predefined	$\mathcal{O}(h)$	FV WENO (5)	TVD RK (2)
[42]	(19)	$\frac{\epsilon}{\Delta t}$ steps	$\mathcal{O}(h)$	FD WENO (5)	TVD RK (3)
[157]	(19)	Steady state	$\mathcal{O}(h)$	FE Q1 (2)	Euler (1)
[63]	(19)	Steady state		FE P2 (3)	Characteristics (1)
[62]	(19)	Steady state		FE P2 (3)	Characteristics (1)
[41]	(19)	Steady state		FD WENO (5)	TVD RK (3)
[163]	(19)	Steady state		FD WENO (5)	TVD RK (3)
[14]	(19)	Steady state		DG (3)	TVD RK (3)

Lagrange multiplier, and the interface was shifted afterwards: $\hat{\phi} \leftarrow \hat{\phi} + c_\phi$. Correction $c_\phi \in \mathbb{R}$ was computed in [32, 33] using:

$$c_\phi = \frac{|\Omega_1| - \int_\Omega H_\epsilon\left(\hat{\phi}\right) \mathrm{d}\Omega}{\int_\Omega \frac{\mathrm{d}H_\epsilon}{\mathrm{d}\phi}\left(\hat{\phi}\right) \mathrm{d}\Omega}, \tag{22}$$

where $|\Omega_1|$ is the exact volume of Ω_1, while [171] used an empirical formula. It should be noticed that because the exact volume is used in equation (22), this correction should also eliminate mass change errors originating from the solution of the LS transport equation in equation (6). All three studies, nevertheless, reported that mass change was not totally eliminated, but that it was negligible (less than 3%) for two-phase flow simulations.

Similar results were obtained in [41, 42], which relied on the method from [136] where the volume conservation constraint is directly embedded in the HRM through equations (20) and (21). Comparisons with the initial method without a mass conservation constraint (equations (15) and (17)) demonstrated a significant improvement in terms of mass conservation, though mass change errors were not completely eliminated [42]. It was shown, in fact, that these errors also depended on the scheme used for the LS transport solver, which might indicate that mass change errors due to LS reinitialization itself were completely eliminated [41]. Tracking mass changes in various two-phase flow problems, moreover, several studies have shown that a reduction down to $10^{-3}\%$ and lower could be achieved [9, 62, 63, 164, 166]. As shown in table 4, these studies often relied on higher-order spatial discretizations.

An interesting numerical implementation of equations (20) and (21) was proposed in [72]. In their control volume FE formulation, the authors proposed to replace the uniform Lagrange multiplier in equation (21) by a discrete field. A separate Lagrange multiplier was computed

Table 5. Studies using exclusively the HRM defined by equation (20) as per [114], though the sign function might differ.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[140]	(17)		$\mathcal{O}(h)$	FV UC (2)	
[74]	(18)	Steady state $\mathcal{O}(h^3)$	$\mathcal{O}(h)$	FD WENO (5)	TVD RK (3)
[1]	(18)	Eikonal $\mathcal{O}(h^{\frac{3}{2}})$		FD WENO (5)	Euler (1)

for each control volume in the neighborhood of each FE mesh node. The objective was to eliminate interface shifting locally instead of globally. The authors, in addition, dealt with mass change errors due to LS transport by implementing a Newton–Raphson algorithm which can be seen as an iterative and exact version of equation (22). For applications involving two-phase flows, the authors demonstrated that they could reduce mass change errors down to machine precision. A similar approach has been developed using a DG approximation and a cut-cell method in [14], and with FD and FV approximations on structured grids in [9, 131, 132].

Surprisingly, two studies mentioned a fictitious time step proportional to the time step Δt used to solve the LS transport equation [33, 134]. This can be explained, however, by the fact that these studies used an explicit time scheme, which involves a dependence of Δt on h.

2.3.1.5. HRM of [114]. An alternative approach to reduce interface shifting was proposed in [114]. The authors identified as a main cause of interface shifting the wrong computation of upwind derivatives for cells of $\mathcal{T}_\Gamma$. They explained that, for each discretization point, the reinitialized LS value should be computed using only LS function values of same sign. When implementing upwind FD schemes, it was important therefore to choose the right stencil to avoid using the wrong points. The authors proposed a new HRM which involved a corrected FD upwind scheme and a corrected sign function for cells of $\mathcal{T}_\Gamma$ [114].

As reported in table 5, three studies have used this approach in the last decade. Although they were all restricted to Cartesian grids, some of them relied on extensions of the approach of [114] to higher-order spatial and temporal discretizations [74].

An interesting aspect presented in table 5, moreover, is that two studies provided detailed definitions of their termination criteria. The L^1 norm of the difference between two successive fictitious time iterations was used as criterion in [74]. The authors proposed to define the tolerance as proportional to h^3, because they expected third-order accuracy on the LS function. They postulated and observed in their simulations that this was equivalent to requiring that the L^1 norm of $|\nabla\hat{\phi}| - 1$ was lower than a tolerance proportional to h^2. This second criterion, which can be coined as Eikonal criterion, was used in [1] but with a tolerance proportional to $h^{\frac{3}{2}}$. For two-phase flow problems, it was shown that mass change errors could be reduced to less than $10^{-3}\%$ [74].

2.3.1.6. HRM of [47]. Another very popular approach to reduce interface shifting during LS reinitialization is the one proposed in [47]. Similarly to equation (20), this approach consists in introducing an additional term to the hyperbolic equation. This so-called forcing term aims at constraining the local position of the interface during fictitious time iterations for cells of $\mathcal{T}_\Gamma$. This constraint, nevertheless, is locally regularized to avoid over-determination of the problem. Studies relying on this approach are listed in table 6.

Table 6. Studies using exclusively the HRM defined by equation (15) as per [47], though the sign function might differ.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[156]			$\mathcal{O}(h)$	FD WENO (3)	TVD RK (3)
[176]			$\mathcal{O}(h)$	FD WENO (3)	TVD RK (3)
[71]	(17)			FV WENO (5)	TVD RK (3)
[26]	(17)	10 steps	$\mathcal{O}(h)$	FV WENO (3)	Explicit Euler (1)
[132]	(17)	Predefined	$\mathcal{O}(h)$	FV WENO (5)	TVD RK (2)
[85]	(17)	Steady state		FD WENO (5)	RK (3)
[177]	(17)	Steady state		FD Upwind (3)	RK (2)
[78]	(18)	Steady state		FV WENO (5)	TVD RK (3)
[79]	(18)	Steady state		FV WENO (5)	TVD RK (3)
[76]	(18)	Steady state	$\mathcal{O}(h)$	FV Upwind (2)	
[77]	(18)	Steady state	$\mathcal{O}(h)$	FV Upwind (2)	
[129]	(19)			FV Upwind	TVD RK (3)
[130]	(19)	Steady state	$\mathcal{O}(h^2)$	FV WENO (5)	TVD RK (3)

Reduction of mass change down to $10^{-3}\%$ and lower with a sufficiently fine discretization were demonstrated by some studies [76–79]. As mentioned previously, it is difficult from these studies to assess whether mass change errors are due to LS reinitialization or LS transport. A thorough analysis can be found in [132], where the authors compared the approaches from [136] (equations (20) and (21)), [114] and [47]. A few important conclusions could be drawn from their analysis. First, higher-order spatial discretization of the HRM proved essential, as the results they obtained with a first-order upwind scheme and the HRM from [114] were far worse in terms of mass conservation as compared to those obtained with a fifth-order WENO scheme, whichever the HRM. Second, higher-order temporal discretization of the HRM appeared essential. This is surprising, given that this time is purely fictitious and only the final solution at steady state is important. Results revealed, nevertheless, that not only the time discretization but also the choice of fictitious time step affected accuracy, in particular in terms of mass conservation [132]. A second-order time scheme was shown to be better than a first-order one, while no significant improvement was observed when using a third-order scheme. Finally, when applying the HRM from [114] to two-phase flow problems, parasitic currents appeared and affected accuracy and mass conservation. Similar currents with a lower magnitude were observed with the HRM from [47]. This issue was not encountered with the HRM from [136] (equations (20) and (21)).

One study combined the HRM from [47] with a global mass correction similar to equation (22), except that it involved an iterative algorithm [26]. They mentioned that the HRM from [47], in their observations, did not reduce mass change errors enough. It should also be noted from table 6 that multiple studies combined the HRM from [47] with the smeared sign function from [104] (equation (18)), which is also designed to attenuate interface shifting. This could be an indication that the HRM from [47] does not give satisfactory results in terms of mass conservation for multiple researchers.

An interesting observation is that one study related the fictitious time step to h^2 [130]. This is explained by the enforcement of a maximum principle. Because the study did not specify which value was actually used for simulations, it is possible that due to restrictions imposed by the explicit time discretization, the chosen fictitious time step was in fact proportional to h.

Table 7. Studies using the HRM defined by equation (15), though the sign function might differ, and omitting discretization points of $\mathcal{N}_\Gamma$.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[118]		30 steps	$\mathcal{O}(h)$	FD UC (5)	Euler (1)
[10]	(19)			FD WENO (5)	TVD RK (3)
[174]	(19)	Steady state		FE (2)	(2)

2.3.1.7. HRMs with fixed interface. It is clear that interface shifting and mass preservation issues with HRMs have been noticed by numerous researchers. All studies in tables 1–6 have attempted to solve these issues by adding terms to the hyperbolic equation or modifying the numerical scheme used to solve it. As presented in table 7, some researchers have chosen a much simpler approach: fixing values at discretization points of $\mathcal{N}_\Gamma$ [118, 174] or having a $|\phi|$ value close enough to zero [10] during the reinitialization process. These values were simply imposed through Dirichlet boundary conditions. These researchers, nevertheless, did not discard the importance of ensuring the distance property even for those discretization points. In [10], for instance, the authors relied on the accuracy of the scheme they used for solving equation (6). In [118], the authors proposed a modification of the LS transport velocity $\boldsymbol{v}$ for discretization points of $\mathcal{N}_\Gamma$ in order to preserve the distance property.

As shown in [10], nevertheless, fixing the values at discretization points with a very low $|\phi|$ value did not eliminate all mass preservation issues. As discussed previously, remaining mass change errors most probably originated from the transport solver used for equation (6).

2.3.1.8. HRMs with penalty. It is reminded that there is an explicit interface preservation constraint in the Eikonal equation (equation (7)). This constraint is replaced by the initial condition in the original HRM in equation (15) and, as discussed in previous paragraphs, this is not enough to avoid interface shifting. It seems appropriate, consequently, that multiple researchers have proposed to insert the interface preservation constraint back into their HRM using a penalization term of the form $\alpha(\phi - \hat{\phi})\frac{\mathrm{d}H_\epsilon}{\mathrm{d}\phi}(\hat{\phi})$, where $\alpha \in \mathbb{R}$ is the penalty coefficient and $\frac{\mathrm{d}H_\epsilon}{\mathrm{d}\phi}(\hat{\phi})$ localizes this constraint at the interface. Studies using this approach are listed in table 8. None of the studies indicated the value used for the penalty parameter α, although it is mentioned that this parameter should be mesh-independent [159].

This penalty approach, in fact, is quite similar to those reviewed in section 2.3.1.7 in the sense that reinitialization is prevented at discretization points close to the interface. It is indicated in all studies of table 8 that the penalization term is designed to prevent interface shifting or, equivalently, ensure mass conservation. Some authors, nevertheless, acknowledged that

Table 8. Studies using the HRM defined by equation (15), though the sign function might differ, with a penalty term.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[179]				FE P1/Q1 (2)	Euler (1)
[159]	(19)			FE P1 (2)	Euler (1)
[178]	(19)			FE P1 (2)	Euler (1)
[180]	(19)			FE P1/Q1 (2)	Euler (1)

Table 9. Studies using the HRM defined by equation (15), though the sign function might differ, combined with a DRM for discretization points of $\mathcal{N}_\Gamma$.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[128]		9 steps	$\mathcal{O}(h)$	FD UC (2)	Euler (1)
[127]		9 steps	$\mathcal{O}(h)$	FD UC (2)	Euler (1)
[12]		10 steps		FD Upwind (1)	Euler (1)
[112]		Convergence		FD Upwind (2)	
[101]	(16)			FD WENO (5)	RK (3)
[29]	(16)	Convergence	$\mathcal{O}(h)$	FE P1 (2)	
[168]	(17)	Convergence	$\mathcal{O}(h)$	DG (4-6)	TVD RK (3)
[175]	(17)	Steady state		FE Q1 (2)	CBS (2)
[167]	(17)	Steady state		FE Q1 (2)	CBS (2)

penalization does not solve all these issues [178–180]. Two studies [179, 180], in fact, implemented a volume conservation constraint similar to equation (22) in order to correct the total mass or volume after reinitialization.

2.3.1.9. HRMs combined with a DRM. Another quite popular way of dealing with discretization points of $\mathcal{N}_\Gamma$ is to use a DRM. Then, similarly as for methods listed in table 7, an HRM with Dirichlet boundary conditions to fix LS values at discretization points of $\mathcal{N}_\Gamma$ is used for remaining points. This strategy has been chosen by the studies listed in table 9.

On the one hand, in [127, 128], this choice is explained by the possibility to fix and maintain the sign of the LS offered by DRMs in order to avoid mass conservation issues with pure HRMs. References [29, 112, 167, 168] also chose a DRM for discretization points of $\mathcal{N}_\Gamma$ with the hope of improving mass conservation. As detailed in section 2.1, it is clear that with a DRM, the sign of the LS function is preserved. Mass conservation, however, cannot be guaranteed because it is not always possible to simultaneously restore the distance property and maintain the interface's position, as shown in figure 9. Except for the benefit regarding the sign, therefore, it is quite surprising that DRMs are chosen by some researchers to improve mass conservation along with an HRM. Multiple studies comparing a pure HRM with a DRM/HRM combination prove, nevertheless, that mass conservation is improved and can be reduced to less than 3% with the combined approach [127, 128, 167].

On the other hand, the choice of an HRM for remaining discretization points is easier to understand. As detailed in section 2.1.2, indeed, it is challenging to reduce computation times with DRMs during closest-point searches. Efficient searches are easier to implement when distances are directly computed only for discretization points of $\mathcal{N}_\Gamma$ because, in most cases, it can be assumed that the closest interface segment is located in a neighboring cell. This assumption would be incorrect for remaining discretization points, which is the reason why it may be more efficient to reinitialize these points with an HRM [12].

Due to the simplified implementation of an efficient closest-point search algorithm, combining a DRM with an HRM does not necessarily involve a significant technical challenge. One major exception concerns higher-order schemes. In [12], a cubic Hermite interpolation was used to represent the LS function, which means that both the LS function and its gradient were stored at each discretization point. The authors used a DRM similar to the one presented in section 2.1.3.2, and an HRM which also dealt with the gradient. A similar higher-order

Table 10. Studies using the HRM defined by equation (15), though the sign function might differ, combined with a particle LS method.

Reference	Sign	Termination	Step	Space scheme (order)	Time scheme (order)
[173]				FE (2)	(2)
[106]	(16)	$\frac{\epsilon}{h}$ steps	$\mathcal{O}(h)$	FE P2 (3)	BDF2 (2)
[107]	(16)	$\frac{\epsilon}{h}$ steps	$\mathcal{O}(h)$	FE P2 (3)	BDF2 (2)

polynomial interpolation was used in [101], but it first had to be reconstructed, similarly as in the DRMs reviewed in section 2.1.3.3.

Regarding the imposition of Dirichlet boundary conditions, most studies chose a strong imposition within discrete algebraic equations. Zhang and Yue [168], as an exception, imposed those conditions through a penalty approach which was more suitable for their DG approximation. In [167, 175], an original iterative approach was proposed where no constraint was imposed on discretization points of $\mathcal{N}_\Gamma$ within the HRM. After each fictitious time iteration, however, the interface was shifted back to its correct position using a DRM.

2.3.1.10. HRMs with particles. Particle LS method are an interesting solution to improve mass conservation. They consist in introducing Lagrangian marker particles which are transported simply by motion based on equation (6). A few studies have integrated particles in an HRM, as shown in table 10. These HRMs with particles are similar to the approach in [167, 175] where a DRM was used to compute small corrections at each fictitious time iteration of the HRM and reduce interface shifting. In the studies of table 10, particles are used to correct LS function values for discretization points of $\mathcal{N}_\Gamma$.

Impressive results in [107, 173] demonstrate that mass change errors can be reduced down to less than $10^{-3}\%$ with sufficiently fine discretization and enough particles. As shown in [106, 107], moreover, these approaches can be extended to ensure higher-order accuracy.

2.3.1.11. HRMs with time marching. Yet another way of preventing sign changes with an HRM is the approach proposed in [55]. Instead of solving a single hyperbolic equation, two equations were solved, one for each side of the interface. The authors mentioned fictitious time iterations were performed until steady state. They relied on higher-order DG spatial approximations and a fourth-order RK time discretization. Although they did not analyze mass conservation, they demonstrated the higher-order accuracy of their approach [55].

2.3.1.12. HRMs with the conservative LS method. A whole class of HRMs is built on the so-called conservative LS method [98, 99]. The fundamental idea of this method is to replace the LS function by $\psi = H_\epsilon(\phi)$. Two-phase flow problems can then be modeled by solving equation (6) directly for ψ instead of ϕ. Reinitialization is achieved by solving the following adaptation of equation (15):

$$\begin{cases} \frac{\partial \hat{\psi}}{\partial \tau} + \nabla.\left(\hat{\psi}\left(1-\hat{\psi}\right)\frac{\nabla\hat{\psi}}{|\nabla\hat{\psi}|}\right) = \epsilon\nabla.\left(\nabla\hat{\psi}\right), \text{ in } \Omega\times[0,T_\tau], \\ \hat{\psi}(\tau=0) = \psi, \text{ in } \Omega. \end{cases} \tag{23}$$

As summarized in table 11, the conservative LS method along with the HRM given by equation (23) has been used in numerous studies, sometimes with slightly modified versions

Table 11. Studies using the HRM defined by equation (23), or similar, with a conservative LS method.

Reference	Termination	Step	Space scheme (order)	Time scheme (order)
[44]			FE	Euler (1)
[7]			DG (5)	RK (4)
[90]			FD WENO (2)	TVD RK (2)
[87]		$\mathcal{O}(h)$	FE (2)	
[86]	2 steps		FD CD (6)	RK (3)
[53]	4 steps	$\mathcal{O}(h)$	DG (4)	RK (5)
[126]	5 steps	$\mathcal{O}(h)$	Wavelets (6)	Krylov (4)
[147]	Predefined	$\mathcal{O}(h)$	FV (2)	TVD RK (3)
[18]	Predefined	$\mathcal{O}(h)$	FV UC (5)	(5)
[141]	Predefined	$\mathcal{O}(h)$	FD UC (5)	CN (2)
[148]	Predefined	$\mathcal{O}(h)$	FV (3)	TVD RK (3)
[105]	Steady state	$\mathcal{O}(h)$	FV (4)	RK (4)
[116]	Steady state	$\mathcal{O}(h)$	FV (4)	RK (4)
[51]	Steady state	$\mathcal{O}(h)$	FV (4)	RK (4)
[102]	Steady state	$\mathcal{O}(h)$	FV UC (3)	RK (3)

of equation (23). As observed for other HRMs, there is a tendency for a higher-order space and time discretization, and a fictitious time step value of the order of mesh size. The number of fictitious time iterations is very often small, mostly because narrow band approaches are used.

Chiodi and Desjardins [18] proposed a modified version of equation (23), but they obtained mass change errors of a few percent for both versions. They demonstrated a reduction of errors below one percent with finer grid sizes. Similar results were obtained in [102, 126]. Moghadam *et al* [86] demonstrated mass change errors could be reduced down to machine precision and attributed this performance to their FD scheme.

2.3.2. Combined hyperbolic reinitialization and transport. Some authors have proposed combined approaches to avoid solving an additional hyperbolic reinitialization problem to maintain the distance property. In [144], LS transport and reinitialization were combined through:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{v}.\nabla\phi = \lambda S(\phi)\left(1 - |\nabla\phi|\right), \text{ in } \Omega, \tag{24}$$

where $\boldsymbol{v}$ is the velocity field (see equation (6)), S is the sign function (see equations (16)–(19)), and λ is a parameter proportional to the velocity that controls the amount of reinitialization. According to [39, 145], λ should be large enough to maintain the distance property, but not too large to avoid altering the interface shape. Equation (24), moreover, is to be solved in place of equation (6), which is the reason why it does not involve any fictitious time. It involves a nonlinear term which can be dealt with using Picard type iterations [38] or an explicit time discretization [39]. Other forms of equation (24) have been proposed in the literature, mainly because some authors chose to replace the LS function by a regularized Heaviside function, similarly as in the conservative LS method (see section 2.3.1.12).

All studies used a narrow band LS method restricting both transport and reinitialization to a small number of discretization points around the interface. Most studies combining reinitialization and transport used linear FEs and a first-order time scheme [13, 39, 80]. Still with linear FEs, some authors opted for a second-order time scheme [38, 40, 143]. Second-order spatial discretization was addressed by one study using quadratic FE interpolation [43]. A higher-order WENO scheme was developed in another study where it was combined with a higher-order explicit time scheme [115]. Except for this last study, this extensive use of low order space and time discretizations is surprising as compared to the literature on HRMs reviewed in section 2.3.1. A possible reason is that equation (24) involves the physical time instead of a fictitious one. It may reduce the freedom regarding the choice of time step and hence limit the applicability of higher-order explicit time schemes. Another possible reason is simply the popularity of these combined hyperbolic reinitialization and transport methods among adepts of the FE method.

2.3.3. Elliptic reinitialization. Most approaches since [137] have consisted in using HRMs because the Eikonal equation in equation (7) cannot be solved directly. This equation is not regular enough to be solved by conventional numerical methods. Recently, regularizations of equation (7) have been proposed in the form of an elliptic problem [15]:

$$\begin{cases} \nabla.\left(d\left(|\nabla\hat{\phi}|\right)\nabla\hat{\phi}\right)=0, & \text{in } \Omega, \\ \hat{\phi}=0 & \text{on } \Gamma. \end{cases} \tag{25}$$

where the so-called diffusion rate d is chosen appropriately to ensure regularity (existence and uniqueness of solutions) and consistency with equation (7). Solutions of equation (25), in particular, should be signed distance functions. As discussed in [15], the single-well diffusion rate

$$d(s)=1-\frac{1}{s} \tag{26}$$

is the simplest choice but raises issues when $s\to 0$, which can be solved by using the double-well diffusion rate

$$d(s)=\begin{cases} 2s^2-3s+1, & s\leqslant 1, \\ 1-\frac{1}{s} & s>1. \end{cases} \tag{27}$$

Regarding the boundary condition $\hat{\phi}=0$ on Γ, it can be enforced using penalization. Since the elliptic problem in equation (25) is nonlinear, moreover, an iterative nonlinear solver is required (e.g. Picard iterations). The termination criterion is usually defined as steady state, i.e. the norm of the difference between two successive iterations reaching a sufficiently low value [2, 158].

This elliptic reinitialization approach has been implemented using a DG approximation and a narrow band method in [142]. They demonstrated that their DG solver was more accurate than the FE method, but only if the penalization coefficient fixing the interface position was large enough. It could not be too large, however, as that deteriorated conditioning and increased iteration numbers. Utz *et al* [142] discussed the choice of penalty coefficient and demonstrated convergence rates up to five using higher-order polynomials and both structured and unstructured grids for academic problems. They encountered some issues with both diffusion rates in

equations (26) and (27) in the vicinity of singular or flat interface portions which led to the divergence of their elliptic reinitialization method.

A new diffusion rate was proposed in [2] to solve these issues. The appropriate choice of a penalty coefficient to fix the interface position was also discussed in this study, and the authors ended up exploring other possibilities as they could not find a systematic way of determining this coefficient depending on the problem at hand, the polynomial degree of their DG approximation, and mesh size. They proposed the use of Lagrange multipliers to enforce the constraint on interface position, with one Lagrange multiplier by element to avoid under- or over-constraining the elliptic problem. They demonstrated that convergence issues were eliminated and that they could get higher-order accurate results up to an order of five with a narrow band method.

The new diffusion rate from [2] was used in another study, which confirmed that convergence issues were eliminated for both structured and unstructured meshes [158]. Instead of Lagrange multipliers the so-called shifted boundary method was used to enforce the interface position.

Yet another diffusion rate was proposed in [48], with the hope of improving Picard iteration numbers. The authors demonstrated the accuracy of their approach for both academic and two-phase flow problems using an FE approximation, a narrow band method and Nitsche's scheme to enforce the interface position. In a recent study [103], a neural network was developed to solve the elliptic problem.

An alternative elliptic reinitialization approach was proposed in [5]. The author's idea was to multiply the LS function by some coefficient to get the gradient norm closer to unity. By playing on different approximation spaces for the scaling coefficient, an elliptic reinitialization approach was derived where the coefficient could be obtained by solving an elliptic problem [5]. Isogeometric analysis was proposed to obtain a regular scaling coefficient, and a global volume correction algorithm was implemented to improve mass conservation. The new approach was tested on unstructured meshes. Optimal convergence rates could not be obtained, which was attributed to the global volume correction algorithm.

2.3.4. Combined elliptic reinitialization and transport. Combining LS transport and elliptic reinitialization consists in introducing the diffusion term from equation (25) into equation (6). This idea has been exploited in the so-called meshfree interface finite element (FE) method [35–37], where a meshfree approximation is used to represent the LS function. This approach improves mass conservation by ensuring there are always meshfree interpolation nodes on the interface, similarly to particle LS methods. It also benefits from the higher-order interpolation capabilities of meshfree methods. The main application in the literature has been microstructure evolution [35–37].

A combined LS transport and elliptic reinitialization approach has also been proposed in [139], where the authors emphasized the need to define a diffusion coefficient to establish the balance between reinitialization and transport. They mentioned that this coefficient should depend on transport velocity and mesh size. They obtained sub-optimal convergence rates using a quadratic FE method, which they attributed to stabilization terms. Their mass change errors, moreover, were of a few percents and could be reduced to less than one percent using a fine mesh. Similar approaches relying on combined elliptic reinitialization and transport with an FE approximation and a narrow band method were developed in [16, 111, 146]. Some authors have preferred the FD method [83]. Similarly as for elliptic reinitialization, various diffusion rates have been investigated in the literature [8, 54]. Some authors have considered incorporating a volume conservation constraint [8], similarly as in equation (20). Applications

of combined elliptic reinitialization and transport methods include microstructure evolution [54] and two-phase flows [8, 50, 110, 125].

2.3.5. Parabolic reinitialization. Equation (25) can be reformulated as a parabolic equation [15]:

$$\begin{cases} \frac{\partial \hat{\phi}}{\partial \tau} + \nabla . \left(d \left(|\nabla \hat{\phi}| \right) \nabla \hat{\phi} \right) = 0, & \text{in } \Omega \times [0, T_\tau], \\ \hat{\phi} = 0 & \text{on } \Gamma. \end{cases} \tag{28}$$

where τ is a fictitious time, similarly as in HRMs.

Only one study employed parabolic reinitialization [3]. Spatial discretization was achieved using a DG approach while different firs-order time discretizations were tested. A narrow band LS method was adopted. They concluded from their tests that an implicit time discretization was preferable over explicit ones, because the number of fictitious time iterations could blow up for some problems due to the stability condition. As compared to the elliptic reinitialization approach based on equation (25), moreover, they observed that the parabolic approach required less iterations to reach steady state. They demonstrated optimal convergence rates using higher-order polynomials up to a degree of three.

2.3.6. Parallel computing. Unless they are combined with a DRM (section 2.3.1.9), global indirect methods do not raise any additional difficulty for parallel computing. If LS reinitialization is part of a framework which involves a transport solver, a flow solver or any other physical model, then any parallel implementation of these solvers can also be used for LS reinitialization. This is even easier and more interesting, in fact, if an explicit (fictitious) time scheme is used with an HRM.

The most used paradigm for parallel implementations of global indirect methods was distributed memory [22, 39, 50, 51, 62, 63, 72, 84, 146, 165, 166], although parallel efficiency was not systematically investigated. Since the discretization grid is partitioned among multiple processes with the distributed memory paradigm, simulations can be run on multiple computing nodes, hence increasing the amount of available memory space. Simulations on very large problems or, equivalently, very fine discretizations are hence possible. This was demonstrated for an HRM in [84], where the authors proved the good scalability of their implementation up to 4096 processes with 2D and 3D adaptive Cartesian grids involving nearly a billion points. They relied on explicit time discretizations for all equations. Good strong scalability was also demonstrated for an HRM in [72] up to 128 processes with 3D unstructured FE meshes involving three million elements, which is quite impressive since they used an implicit time discretization. Similar demonstrations for HRMs with adaptive unstructured FE meshes can be found in [51, 63]. Weak and strong scalability has also been evaluated for a combined elliptic reinitialization and LS transport method up to 25 000 processes, with results showing a nearly-optimal scalability with at least 12 500 discretization points per process [50].

Shared memory parallel implementations of HRMs have been developed [75, 106], although their efficiency has not been analyzed.

GPU-accelerated HRMs have been used in [7, 56]. Results in [56] demonstrate that GPUs can provide much more computing power than CPUs, especially when using a higher-order DG approximation with an explicit time discretization. These results, however, were limited to small unstructured FE meshes of only 7000 elements.

All three parallel computing paradigms were combined in [112]. For Cartesian grids involving up to 100 million points, the authors demonstrated a good strong scalability thanks to their explicit time discretization.

The study relying on a neural network to solve the elliptic reinitialization problem used a GPU to train the neural network [103]. Scalability was not evaluated.

2.3.7. Summary. Global indirect methods originally involved solving a hyperbolic equation with a fictitious time variable. In most studies, this equation was solved until a steady state was achieved. Researchers typically predefined either a tolerance for changes in the LS function between two fictitious time increments or a fixed number of time increments required to reach steady state. Additionally, most studies performed reinitialization only within a narrow band surrounding the interface. Regarding the fictitious time step, most studies chose a value proportional to mesh size.

Numerous studies have employed this HRM and sometimes proposed improvements. The main concern in the literature has been to improve accuracy by increasing the convergence rate and reducing mass change errors.

Convergence rate has been successfully improved with higher-order space and time discretizations. The chosen convergence order for spatial discretization was always greater or equal to the one chosen for time. Higher-order spatial discretization was often achieved with FD or FV WENO schemes for Cartesian grids, and DG schemes for unstructured grids. Higher-order time discretization was mostly achieved using explicit RK schemes.

Although many methods have been proposed to reduce mass change errors, results were not always convincing. The best results were obtained by combining a higher-order scheme with a volume conservation constraint. This constraint was either directly integrated into the hyperbolic equation, or implemented in a weak sense by shifting the interface after reinitialization to ensure mass conservation. Some studies proposed a similar approach with a local correction in order to shift the interface only where mass changes were significant, and demonstrated that mass change errors could be reduced down to machine precision. Particle LS method have also been shown to reduce mass change errors. These attempts to improve mass conservation have led to multiple options to fix the interface's position during fictitious time increments: volume conservation constraints, relying on another reinitialization method such as a DRM for discretization points of $\mathcal{N}_\Gamma$, penalty approaches or Lagrange multipliers.

Other global indirect methods have been developed in the literature. Elliptic reinitialization approaches consist in solving a regularized version of the Eikonal equation. This nonlinear problem is solved with an iterative process. Similarly to HRMs, higher-order spatial discretization is possible, but there is no fictitious time and hence no time discretization. Some techniques developed for HRMs such as Lagrange multipliers or penalty approaches to fix the interface's position during reinitialization have been extended to elliptic reinitialization.

Parabolic reinitialization approaches also rely on a regularized version of the Eikonal equation, but they still involve a fictitious time. Only one study employed parabolic reinitialization in the last decade. Comparisons with an elliptic reinitialization approach showed a small reduction in terms of the number of iterations to reach steady state.

Some researchers have developed combined reinitialization and LS transport approaches. These approaches are particularly suited for two-phase flow problems, where a transport equation is solved at each time increment for the LS function. Combining reinitialization and LS transport can be advantageous in terms of computation time and also opens the possibility to reduce mass change errors during LS transport. These approaches, however, involve a parameter controlling the amount of reinitialization.

Parallel implementations of global indirect methods have relied mostly on distributed memory paradigms. Impressive simulations involving billions of discretization points have demonstrated that nearly optimal scalability can be obtained for Cartesian, unstructured and even adaptive grids.

3. Discussions

The literature review in section 2 includes 161 papers from the past decade (2014–2024), among which 129 (80.1%) relied on a global indirect method, 24 (14.9%) on a DRM and 9 (5.59%) on a local indirect method (one paper combined a global and a local indirect method). The most popular global indirect methods are clearly HRMs with 100 papers (62.1% of total), and then combined elliptic reinitialization and LS transport methods with 13 papers (8.07% of total), combined hyperbolic reinitialization and LS transport methods with 10 papers (6.21% of total), elliptic reinitialization methods with 7 papers (4.35% of total), and parabolic reinitialization methods with 1 paper. Note that, although they are cited in the present paper, 20 references are not included in those statistics because they did not fit the scope of the review [6, 15, 21, 25, 30, 47, 52, 59, 98–100, 104, 114, 122, 123, 133, 136, 137, 144, 169].

Even though global indirect methods are clearly more popular, and in particular HRMs, some trends can be analyzed independently from the chosen method. Half papers, for instance, relied on narrow band LS methods. Higher-order reinitialization, in addition, is possible for all methods, as well as parallel computing. Interesting features such as the extension to unstructured grids, moreover, have been developed only for some methods. In the following, these different aspects are analyzed in order to identify limitations with current reinitialization methods and challenges which are yet to be overcome. Computational cost is discussed in section 3.1, parallel efficiency in section 3.2, approximation methods and extensions to unstructured grids in section 3.3, developments of higher-order methods in section 3.4, mass change errors reduction techniques in section 3.5, investigations on the robustness and tuning of numerical parameters (if any) in section 3.6 and finally challenges related to applications in section 3.7.

3.1. Computational cost

From a theoretical point of view, LS reinitialization techniques differ drastically in terms of computational complexity. The computational complexity of DRMs for unstructured grids can be reduced with space partitioning techniques at best to $\mathcal{O}(N_p \log(N_s))$, with N_p the number of discretization points in the narrow band and N_s the number of segments resulting from the reconstruction of Γ_h (see section 2.1.2). With the stencil-based space-partitioning technique presented in section 2.1.2.1, this complexity can be reduced for Cartesian grids to $\mathcal{O}(N_p)$. Similarly, the computational complexity of local indirect methods can be reduced with a heap to $\mathcal{O}(N_p \log(N_p))$ (see section 2.2.1) and, for Cartesian grids, to $\mathcal{O}(N_p)$ (see section 2.2.3).

Because global indirect methods rely on the solution of a partial differential equation, their computational complexity depends on the numerical method used to approximate and solve this equation. At best, if an explicit discretization is used for the fictitious time or if the linear system resulting from an implicit discretization is solved with an iterative method, then the computational complexity of each iteration is $\mathcal{O}(N_p)$. The total computational complexity finally depends on the number of iterations which, based on the literature review in section 2.3, seems to be a fixed number.

On Cartesian grids, therefore, there is no theoretical result indicating the superior efficiency of one method over another. On unstructured grids, global indirect methods can theoretically

seem more efficient. This efficiency, however, highly depends on whether an explicit discretization can be used for the fictitious time, or if an efficient linear solver can be used for the linear system resulting from an implicit discretization.

It is not surprising, in fact, that some papers show that DRMs are more accurate and reduce mass change errors and computation time as compared to HRMs [92, 124, 135].

3.2. Parallel efficiency

Only one study from the past decade developed a parallel implementation of a DRM and evaluated its scalability, as presented in section 2.1.2. This study proposed an advanced distributed space partitioning algorithm [124]. Scalability was found to be limited because mesh partitioning did not consider the interface. Some partitions, consequently, ended up with large portions of the narrow band surrounding the interface, while other partitions did not have any. This uneven distribution of distance computation operations severely deteriorated scalability. Two other studies developed a distributed algorithm [31] and a GPU-accelerated algorithm [123], but did not evaluate scalability.

This difficulty to obtain an optimal scalability was also reported for local indirect methods. Parallel implementations of the FMM and the FIM have been developed in the past decade, as presented in sections 2.2.1.2 and 2.2.2. The parallel efficiency of a shared memory FMM improving load balancing during LS reinitialization by considering the interface during partitioning was demonstrated up to 24 threads [109]. Parallel implementations of the FMM and the FIM, however, were all restricted to Cartesian grids and low order schemes [109, 150, 161].

Recent developments on Hopf-Lax formula-based reinitialization approaches are promising, as optimal scalability has been demonstrated with a parallel implementation on a GPU for sufficiently large grids [113]. Distributed parallel implementations of these approaches have not been developed in the past decade. Unstructured or adapted grids have not been addressed either.

If parallel scalability is a priority, global indirect methods offer multiple advantages. The implementation, first, is greatly simplified. Global indirect methods consist in solving partial differential equations. Since LS reinitialization methods are always integrated in a computational framework involving the solution of other partial differential equations (LS transport, flow, etc), it is quite straightforward to rely on the same framework for LS reinitialization. In other words, if the LS transport solver is implemented in a parallel framework, the same implementation can be used for the global indirect methods. This is even simpler if an HRM is used with an explicit time discretization.

Second, the most impressive and demonstrative parallel scalability results have been obtained using global indirect methods, as detailed in section 2.3.6. These results have been obtained with distributed memory paradigms for Cartesian, unstructured or adaptive grids involving more than billions of discretization points partitioned among up to 25 000 processes.

It can be concluded that to develop an efficient parallel LS reinitialization method, it is best to pick a global indirect method, and in particular an HRM with an explicit fictitious time discretization. This is especially justified for higher-order schemes.

As an alternative, local indirect methods can be interesting for Cartesian grids if higher-order accuracy is not a priority. DRMs can be interesting for unstructured and adapted grids, but more developments are required to improve load balancing and parallel efficiency.

It should be mentioned that, although shared memory or GPU-accelerated parallel algorithms can be interesting to improve scalability, distributed memory algorithms are necessary in order to deal with large grids. It is not surprising, therefore, that 16 papers relied on

a distributed memory paradigm, while only 6 papers relied on shared memory, and 5 papers relied on GPU acceleration.

3.3. Approximation method and grid

In the past decade, 52 papers (32.2%) relied on an FD approximation, 47 (29.2%) on an FE approximation, 42 (26.1%) on an FV approximation, 14 (8.70%) on a DG approximation, 5 (3.11%) on a meshfree approximation and 2 (1.24%) on other approximations. Since most papers relying on FD or FV approximations worked with Cartesian grids for which these two schemes are very similar, the corresponding numbers can be combined, leading to 94 papers (58.4%).

There were in fact 85 papers (52.8%) relying exclusively on Cartesian grids, among which only 14 papers did not rely on an FD or FV approximation. Papers relying on the FE method, on the opposite, conducted at least one simulation on an unstructured mesh, except for eight papers which only exploited Cartesian grids. Similarly, there were only three papers choosing a DG approximation but not showing any result on an unstructured mesh.

Among the studies which showed results on unstructured meshes, many actually kept a uniform size. There were only 45 papers (28.0%) exploiting adapted meshes, which were often refined close to the interface. Since the LS method is typically used to represent moving interfaces, some papers relied on automatic mesh adaption techniques to keep the mesh refined close to the interface during the simulation. This strategy was shown to be advantageous in terms of accuracy [88], convergence rate [63], and computational cost [17, 93]. These advantages were nevertheless analyzed on complex problems where LS reinitialization was integrated into frameworks which also included at least a flow solver and an LS advection solver. It is not possible to evaluate whether adaptive meshing techniques specifically improve LS reinitialization accuracy or computational cost.

The choice of approximation method, as a conclusion, seems to be intrinsically related to the choice of discretization grid. Cartesian grids offer limited possibilities when it comes to real and arbitrary geometries. They are also limited for problems involving interfaces where adaptive grids can be advantageous to refine the discretization in the narrow band. For unstructured or adaptive grids, the first option in the past decade has been an FE approximation, while the second choice has been a DG approximation.

3.4. Convergence rate

Obtaining higher-order accuracy has been critical for many researchers, not only to improve the overall accuracy but in particular to reduce mass change errors. There are in fact 91 papers (56.5%) using LS reinitialization methods with a convergence rate of at least three on the LS function value, at least two on the normal vector (equation (3)) and at least one on the mean curvature (equation (4)). In a computational framework involving a higher-order LS transport and flow solver, it is key to integrate a higher-order LS reinitialization method. If the LS reinitialization method cannot be extended to reach higher-order accuracy, then the accuracy of the whole computational framework is affected. This is problematic for physical models that depend on the mean curvature (e.g. surface tension in two-phase flow problems).

Higher-order local indirect methods, on the one hand, have only been developed for Cartesian grids. Higher-order DRMs and global indirect methods, on the other hand, have been developed for both structured and unstructured grids.

Higher-order DRMs relying on FE or DG approximations are particularly interesting because they are compatible with unstructured grids, but there are technical issues regarding

distance computation. Results obtained with Newton's algorithm are promising in that regard, though convergence of this algorithm has been shown to highly depend on the initial guess.

Higher-order global indirect methods have been developed with multiple approximations: FD, FV, FE, DG and meshfree. No technical difficulty has been mentioned in the past decade regarding these higher-order schemes. This could be explained by the fact that most higher-order global indirect methods are HRMs which exploit a hyperbolic equation solver initially developed and tested for LS transport.

3.5. Mass change errors

Mass change errors are related to the convergence rate, as studies have shown that when higher-order accuracy is obtained for distance values, then mass change errors are also reduced with a higher convergence rate (see section 2.3.1.4). There were, nevertheless, many contributions developing other methods to reduce and even eliminate mass change errors. These contributions have only been developed for DRMs and global indirect methods.

The most general and easiest way to reduce and even completely eliminate volume changes due to LS reinitialization in a global sense is to:

- compute the volume of Ω_1 before LS reinitialization,
- perform LS reinitialization,
- recompute the volume of Ω_1,
- shift the LS function by a constant to ensure mass conservation.

This algorithm can be combined with all LS reinitialization methods. This review of the past decade's literature reveals that in practice only two papers implemented this algorithm or a similar approach with a DRM. There are 12 studies combining this technique with a global indirect method, to which may be added 17 studies from table 4 which used equations (20) and (21). These equations directly enforce global volume conservation within a HRM. There are hence 31 papers (19.3%) which implemented a global volume conservation constraint. Indifferently from the LS reinitialization, this method has been shown to be very effective, reducing global volume change errors below one percent for various problems. The main critic is that the volume correction is global, as described in equation (22). It systematically shifts the whole interface for volume changes which may be very local.

To overcome this limitation, 56 papers (34.8%) employed local techniques to reduce and sometimes eliminate mass change errors. The main idea is similar as the global volume conservation constraint, only volume changes are tracked locally. Depending on the approximation, mass corrections can be computed at nodes (e.g. [89, 167]) or control volumes (e.g. [72]) for FE methods, and cells for FD and FV methods (e.g. [73, 76]). In some methods such as the conservative LS method or particle LS methods, local volume changes can be tracked because the interface's position is not determined solely by the LS function (see sections 2.3.1.10 and 2.3.1.12). In many global indirect methods, moreover, attempts are made to reduce mass change errors by modifying the numerical scheme used for discretization points of $\mathcal{N}_\Gamma$ or cells of $\mathcal{T}_\Gamma$, or adding constraints such as penalty terms (see sections 2.3.1.5, 2.3.1.6 and 2.3.1.8).

3.6. Robustness and influence of numerical parameters

Only some methods presented in this review require tuning numerical parameters. Some of these parameters have largely been investigated and criteria have been established to determine them. For some parameters, more investigation is required.

Local indirect methods do not involve any parameters and most DRMs neither. The only exception is higher-order DRMs relying on Newton's method, which require a rule regarding the proposition of the initial guess in order to ensure convergence.

Global indirect methods require some parameters. HRMs require the definition of a fictitious time step. This subject has been largely discussed in the literature, especially when an explicit fictitious time discretization is employed.

Combined hyperbolic or elliptic reinitialization and transport methods require the definition of a parameter controlling the importance of reinitialization. Regarding the control parameter for combined hyperbolic reinitialization and transport, researchers acknowledge that a balance should be found. A too small value would not ensure proper reinitialization while a too large value would alter the interface's shape and increase iteration numbers [39, 145]. The authors of [145] recognized that this parameter should be 'optimized for a given application'. The same observation has been made for combined elliptic reinitialization and transport [139], although researchers have proposed a proportionality relationship between this control parameter and transport velocity and mesh size.

Global indirect methods enforcing the interface's position through penalization require the definition of a penalty parameter. These parameters are well-known to require tuning because large values are required to ensure the constraints are enforced, but too large values lead to deteriorated conditioning and increased iteration numbers.

3.7. Applications

By far, scientific research on LS reinitialization in the past decade has focused first on two-phase flow problems with 114 papers (70.8%). Other applications are microstructure evolutions with 6 papers (3.73%) and fluid-structure interaction with 4 papers (2.48%). There are also 39 papers (24.2%) focusing on numerical developments and academic examples and not dealing with any particular application.

Microstructure evolution problems include dendritic solidification, where a curvature-dependent LS function evolution equation is solved [36]. Simulations of microstructure evolution in polycrystalline materials are particularly impressive, because the simulation domain is split into more than two parts, hence requiring many LS functions [35, 101, 124].

Two-phase flow problems rely on the very simple LS transport equation in equation (6). Complexity is actually added into the flow solver, which involves a curvature-dependent surface tension term. One particular topic of interest that has been addressed by only eight papers (4.97%) in the past decade is LS reinitialization in the presence of moving contact lines.

Two-phase flows with moving contact lines are two-phase flows where the interface between the two fluids meets solids, hence forming a triple junction: first fluid with solid, second fluid with solid and first fluid with second fluid. It is well-established that at equilibrium, due to a physical phenomenon called wetting, the interface between the two fluids Γ and the solid's boundary meet at a characteristic angle denoted static contact angle. In non-stationary conditions, there is a local force at the triple junction as the interface Γ resists the flow whenever it is forced to leave its static contact angle. Computation of the contact angle between Γ and the solid's boundary is simplified with the LS method, as well as the localization of the triple junction (see e.g. [169]).

As demonstrated in [21], some reinitialization methods systematically alter the angle between Γ and $\partial\Omega$ by generating numerical errors moving it towards 90°. Since numerical approaches for wetting often exclude solids from the simulation domain, they include the solid's boundary as a part of $\partial\Omega$. Numerical errors hence emerge as the fluid velocity computed

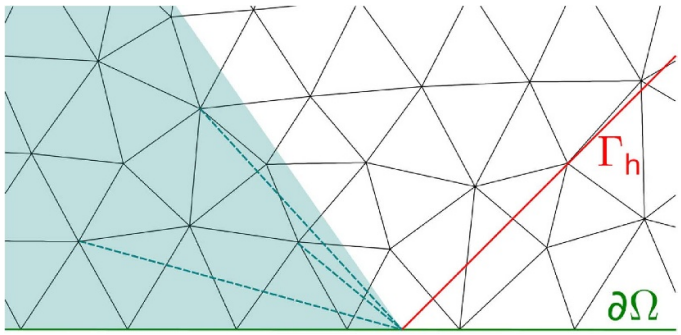

Figure 11. Example of interface Γ_h (in red) meeting the domain boundary $\partial\Omega$ (in green) with a non-orthogonal angle. The discretization points in the shaded region are in a blind spot and would be projected with a DRM onto the intersection $\Gamma_h \cap \partial\Omega$, leading to normal vectors that would not be orthogonal to Γ_h and contour lines that would not be parallel to Γ_h.

by the flow solver attempts to rotate Γ towards the static contact angle, while the LS reinitialization method moves it towards 90°. This issue is actually not specific to wetting, as it is encountered whenever $\Gamma \cap \partial\Omega \neq \emptyset$ and the angle is different from 90°. As shown in figure 11, depending on the angle between Γ and $\partial\Omega$, some discretization points end up in a blind spot. The closest point from Γ to these points is on the intersection $\Gamma \cap \partial\Omega$, which automatically leads to contour lines that are orthogonal to $\partial\Omega$. The angle between Γ and $\partial\Omega$, consequently, also shifts towards orthogonality.

Some treatments have been proposed in the literature to solve this issue, although they have not been generalized to all LS reinitialization methods yet. DRMs are severely affected because they lead to the situation shown in figure 11 where numerically *correct* distance values lead to physically *incorrect* contour lines.

The first treatment proposed in the literature can be found in [133] and addresses this issue for HRMs. It was used in three papers from the past decade [131, 156, 176]. It consists in solving a different hyperbolic equation to reinitialize LS function values for points of $\partial\Omega$ in blind spots. These values are then imposed as boundary conditions at points of $\partial\Omega$ in blind spots when solving the original hyperbolic equation (e.g. equations (15) or (20)). This correction consists in propagating the contact angle from triple junctions to all points of $\partial\Omega$ in blind spots. Xu and Ren [156] proposed to solve this new equation simultaneously with the one from the original HRM. This required defining a parameter controlling the propagation speed of the correction. Solomenko *et al* [131] proposed to eliminate this parameter by first reinitializing values at points of $\partial\Omega$ in blind spots, and then dealing with other points. This technique can be applied to other global indirect methods such as elliptic reinitialization [48].

Pertant *et al* [105] proposed a numerical treatment for DRMs. Their method consisted in identifying discretization points in blind spots, and then extrapolating distance values from the points outside blind spots using multi-dimensional Taylor series expansions. This technique consisted in starting from the points outside blind spots and using their correct LS function gradients and Hessian matrices to extrapolate LS function values for their neighbors in blind spots. This iterative technique was applied to successive neighbors, propagating the correction to all points in blind spots. It was successfully implemented in a distributed computing code, but scalability was not analyzed [105]. As shown in figure 12, it led to correct contour lines.

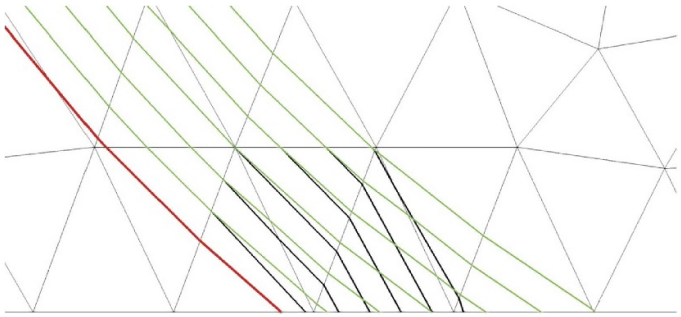

Figure 12. LS reinitialization with a DRM showing the interface in red, the reinitialization result without treatment of blind spots in black and the reinitialization result with the treatment in green. Reprinted from [105], Copyright (2021), with permission from Elsevier.

To the author's best knowledge, this issue has not been analyzed or reported for local indirect methods. Since local indirect methods do not compute distances and do not involve any boundary conditions, it is reasonable to assume they are not affected by blind spots.

4. Conclusions

LS reinitialization methods can be classified in three categories:

- DRMs which geometrically compute distances, including methods which accelerate closest-point searches with space partitioning techniques.
- Local indirect methods which solve the Eikonal equation point-by-point, including the FMM, the FIM, the FSM, and Hopf-Lax formula-based methods.
- Global indirect methods which solve the Eikonal equation for all points at once, including HRMs, elliptic reinilialization methods, parabolic reinitialization methods and combinations of hyperbolic or elliptic reinitialization with LS transport.

This review of the literature has focused on the past decade (2014–2024), covering 161 papers mentioning the keywords *LS* and *reinitialization* or *redistancing* according to Web of Science™ and dealing with simulations of moving fronts in computational mechanics and materials science. The first choice for LS reinitialization in the past decade, clearly, is an HRM with a higher-order FD or FV spatial discretization restricted to Cartesian grids and a higher-order explicit RK time discretization. It also appears that the first application is two-phase flows. The first concern for most researchers when developing LS reinitialization methods is mass change errors, which justifies the tendency to develop higher-order schemes. The second choice is a DRM, while local indirect methods have received little attention.

From this review, it appears that several topics have been largely addressed in the literature and convincing results have been obtained:

- Parallel implementation of all reinitialization methods, especially with distributed-memory paradigms, with numerical proofs of scalability, although it was not always optimal.
- LS reinitialization through a DRM or a global indirect method with various approximation methods such as FD, finite volume (FV), FE, DG or meshfree methods, and on both structured and unstructured grids.

Table 12. Comparison of different reinitialization methods. For each line, the best result achievable is highlighted in blue, and the worst one in brown. Easiness and difficulty are measured in the sense of implementation effort, with easy meaning little additional effort is required.

Method	Direct	Local indirect	Global indirect
Computational complexity	$\mathcal{O}(N_p \log(N_s))$	$\mathcal{O}(N_p \log(N_p))$	$\mathcal{O}(N_p)$
↘ for Cartesian grids	$\mathcal{O}(N_p)$	$\mathcal{O}(N_p)$	$\mathcal{O}(N_p)$
Parallel efficiency	limited	none	optimal
↘ for Cartesian grids	limited	optimal	optimal
Unstructured grids	easy	difficult	easy
Higher-order convergence	difficult	difficult	easy
Mass change reduction	global	none	local
Numerical parameters	none	none	multiple
Blind spots elimination	difficult	not reported	difficult

- The extension of all reinitialization methods to obtain a higher-order convergence rate on LS function values, LS function gradients and even second derivatives of the LS function, although some technical difficulties were mentioned for DRMs, and developments for local indirect methods were restricted to Cartesian grids.
- Many numerical treatments to reduce mass change errors, including global interface shifting, local corrections, particle LS methods, and developments of higher-order schemes.
- Narrow band LS reinitialization methods, where computational cost is reduced by conducting reinitialization only for discretization points close enough to the interface.
- Applications of the LS method to two-phase flows.

This review also enables a factual comparison between the different reinitialization methods. As summarized in table 12, global indirect methods dominate according to many criteria, mainly due to the simplicity of their implementation, especially in computing codes which already include a transport solver. The only issue with global indirect methods is the presence of numerical parameters which have to be tuned to ensure robustness and efficiency. DRMs do not suffer from this limitation, but require more implementation effort to access higher-order convergence rates, and improve parallel efficiency. Local indirect methods, finally, are disadvantaged by a lack of attention from researchers in the past decade. Mass change error reduction techniques and blind spots, consequently, have not been investigated. Parallel algorithms for unstructured grids, similarly, have not been developed.

In the future, in the author's opinion, it would be interesting to investigate:

- Parallel implementation of DRMs considering the interface during partitioning to improve load balancing during distance computations, with the hope of improving scalability.
- Extensions of all local indirect methods (the FMM, the FIM, the FSM and the Hopf-Lax formula-based method) to unstructured grids, especially regarding the parallel implementation and higher-order accuracy.
- Robust distance computation techniques for higher-order DRMs, especially in the context of an FE or DG approximation.
- Robust criteria to determine the control parameter for combined hyperbolic or elliptic reinitialization and LS transport methods.

- Other applications of the LS method, which may raise specific issues for LS reinitialization. Among those, two-phase flows with moving contact lines raise the challenge of dealing with blind spots. Little attention has been given to this issue in the past decade, with most developments focusing on blind spots corrections for global indirect methods.

Funding

This work was supported by the French Agence Nationale de la Recherche (ANR) through the MISSA Project (ANR-22-CE46-0002).

Conflict of interest

The author has no relevant financial or non-financial interests to disclose.

ORCID iD

Modesar Shakoor https://orcid.org/0000-0001-6802-6176

References

[1] Abu-Al-Saud M O, Riaz A and Tchelepi H A 2017 Multiscale level-set method for accurate modeling of immiscible two-phase flow with deposited thin films on solid surfaces *J. Comput. Phys.* **333** 297–320
[2] Adams T, Giani S and Coombs W M 2019 A high-order elliptic PDE based level set reinitialisation method using a discontinuous Galerkin discretisation *J. Comput. Phys.* **379** 373–91
[3] Adams T, McLeish N, Giani S and Coombs W M 2019 A parabolic level set reinitialisation method using a discontinuous Galerkin discretisation *Comput. Math. Appl.* **78** 2944–60
[4] Afrasiabi M, Roethlin M and Wegener K 2018 Thermal simulation in multiphase incompressible flows using coupled meshfree and particle level set methods *Comput. Methods Appl. Mech. Eng.* **336** 667–94
[5] Akkerman I 2017 Monotone level-sets on arbitrary meshes without redistancing *Comput. Fluids* **146** 74–85
[6] Akram F, Kim J H, Lee C-G and Choi K N 2015 Segmentation of regions of interest using active contours with SPF function *Comput. Math. Methods Med.* **2015** 710326
[7] Al-Salami J, Kamra M M and Hu C 2021 A high order flux reconstruction interface capturing method with a phase field preconditioning procedure *J. Comput. Phys.* **438** 110376
[8] Alamé K, Anantharamu S and Mahesh K 2020 A variational level set methodology without reinitialization for the prediction of equilibrium interfaces over arbitrary solid surfaces *J. Comput. Phys.* **406** 109184
[9] An R, Gu Z, Zhou T and Yu C 2023 Numerical simulation of incompressible interfacial flows by a level set re-distancing method with improved mass conservation *Ocean Eng.* **290** 116428
[10] An R D and Yu C H 2020 A level set redistancing algorithm for simulation of two-phase flow *Numer. Heat Transfer B* **78** 30–53
[11] Ansari M R, Azadi R and Salimi E 2016 Capturing of interface topological changes in two-phase gas–liquid flows using a coupled volume-of-fluid and level-set method (VOSET) *Comput. Fluids* **125** 82–100
[12] Anumolu L and Trujillo M F 2018 Gradient augmented level set method for phase change simulations *J. Comput. Phys.* **353** 377–406
[13] Bahbah C, Khalloufi M, Larcher A, Mesri Y, Coupez T, Valette R and Hachem E 2019 Conservative and adaptive level-set method for the simulation of two-fluid flows *Comput. Fluids* **191** 104223

[14] Bai X and Li M 2023 A conservative sharp-interface numerical method for two-dimensional compressible two-phase flows *J. Sci. Comput.* **97** 30
[15] Basting C and Kuzmin D 2013 A minimization-based finite element formulation for interface-preserving level set reinitialization *Computing* **95** 13–25
[16] Basting C, Kuzmin D and Shadid J N 2017 Optimal control for reinitialization in finite element level set methods *Int. J. Numer. Methods Fluids* **84** 292–305
[17] Bellotti T and Theillard M 2019 A coupled level-set and reference map method for interface representation with applications to two-phase flows simulation *J. Comput. Phys.* **392** 266–90
[18] Chiodi R and Desjardins O 2017 A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows *J. Comput. Phys.* **343** 186–200
[19] Choi S, Cho M H, Choi H G and Yoo J Y 2016 A Q2Q1 integrated finite element method with the semi-implicit consistent CSF for solving incompressible two-phase flows with surface tension effect *Int. J. Numer. Methods Fluids* **81** 284–308
[20] Cui Z, Yang Z, Jiang H-Z, Huang W-X and Shen L 2018 A sharp-interface immersed boundary method for simulating incompressible flows with arbitrarily deforming smooth boundaries *Int. J. Comput. Methods* **15** 1750080
[21] Della Rocca G and Blanquart G 2014 Level set reinitialization at a contact line *J. Comput. Phys.* **265** 34–49
[22] Dianat M, Skarysz M and Garmory A 2017 A coupled level set and volume of fluid method for automotive exterior water management applications *Int. J. Multiph. Flow* **91** 19–38
[23] Duy T-N and Hino T 2020 An improvement of interface computation of incompressible two-phase flows based on coupling volume of fluid with level-set methods *Int. J. Comput. Fluid Dyn.* **34** 75–89
[24] Duy T-N, Nguyen V-T, Phan T-H and Park W-G 2021 An enhancement of coupling method for interface computations in incompressible two-phase flows *Comput. Fluids* **214** 104763
[25] Emmendoerfer H and Fancello E A 2016 Topology optimization with local stress constraint based on level set evolution via reaction–diffusion *Comput. Methods Appl. Mech. Eng.* **305** 62–88
[26] Engberg R F and Kenig E Y 2014 Numerical simulation of rising droplets in liquid–liquid systems: a comparison of continuous and sharp interfacial force models *Int. J. Heat Fluid Flow* **50** 16–26
[27] Esser P and Grande J 2015 An accurate and robust finite element level set redistancing method *IMA J. Numer. Anal.* **35** 1913–33
[28] Fahsi A and Soulaïmani A 2017 Numerical investigations of the XFEM for solving two-phase incompressible flows *Int. J. Comput. Fluid Dyn.* **31** 135–55
[29] Frey P, Kazerani D and Ta T 2018 An adaptive numerical scheme for solving incompressible 2-phase and free-surface flows *Int. J. Numer. Methods Fluids* **87** 543–82
[30] Fu J, Li H, Gao L and Xiao M 2019 Design of shell-infill structures by a multiscale level set topology optimization method *Comput. Struct.* **212** 162–72
[31] Fu L, Hu X Y and Adams N A 2017 Single-step reinitialization and extending algorithms for level-set based multi-phase flow simulations *Comput. Phys. Commun.* **221** 63–80
[32] Gao P 2020 The finite element numerical investigation of free surface Newtonian and non-Newtonian fluid flows in the rectangular tanks *Math. Probl. Eng.* **2020** 7548192
[33] Gao P, Ouyang J and Zhou W 2018 Development of a finite element/discontinuous Galerkin/level set approach for the simulation of incompressible two phase flow *Adv. Eng. Softw.* **118** 45–59
[34] Gao P, Zhao Z and Yang Y 2021 The numerical modeling and study of gas entrapment phenomenon in non-isothermal polymer filling process *J. Non-Newton. Fluid Mech.* **294** 104575
[35] Ghoneim A 2015 A meshfree interface-finite element method for modelling isothermal solutal melting and solidification in binary systems *Finite Elem. Anal. Des.* **95** 20–41
[36] Ghoneim A 2016 A new technique for numerical simulation of dendritic solidification using a meshfree interface finite element method *Int. J. Numer. Methods Eng.* **107** 813–52
[37] Ghoneim A Y 2018 The meshfree interface finite element method for numerical simulation of dendritic solidification with fluid flow *Int. J. Comput. Methods* **15** 1850057
[38] Giavaras A and Boateng E 2015 Transient filling modelling at meso-level for RTM process using a single phase LSM *Int. J. Mater. Form.* **8** 197–210
[39] Grave M, Camata J J and Coutinho A L G A 2020 A new convected level-set method for gas bubble dynamics *Comput. Fluids* **209** 104667
[40] Grave M and Coutinho A L G A 2022 Comparing the convected level-set and the Allen–Cahn phase-field methods in AMR/C simulations of two-phase flows *Comput. Fluids* **244** 105569

[41] Gu Z, Wen H, Yu C and Sheu T W H 2018 Interface-preserving level set method for simulating dam-break flows *J. Comput. Phys.* **374** 249–80
[42] Gu Z H, Wen H L, Ye S, An R D and Yu C H 2018 Development of a mass-preserving level set redistancing algorithm for simulation of rising bubble *Numer. Heat Transfer B* **74** 699–727
[43] Guermond J-L, de Luna M Q and Thompson T 2017 An conservative anti-diffusion technique for the level set method *J. Comput. Appl. Math.* **321** 448–68
[44] Habera M and Hron J 2017 Modelling of a free-surface ferrofluid flow *J. Magn. Magn. Mater.* **431** 157–60
[45] Hadidi A and Jalali-Vahid D 2016 Numerical simulation of dielectric bubbles coalescence under the effects of uniform magnetic field *Theor. Comput. Fluid Dyn.* **30** 165–84
[46] Haghshenas M, Wilson J A and Kumar R 2017 Algebraic coupled level set-volume of fluid method for surface tension dominant two-phase flows *Int. J. Multiph. Flow* **90** 13–28
[47] Hartmann D, Meinke M and Schröder W 2010 The constrained reinitialization equation for level set methods *J. Comput. Phys.* **229** 1514–35
[48] Hashemi A R, Hashemi M R, Ryzhakov P B and Rossi R 2024 Optimization-based level-set re-initialization: a robust interface preserving approach in multiphase problems *Comput. Methods Appl. Mech. Eng.* **420** 116699
[49] Henri F, Coquerelle M and Lubin P 2022 Geometrical level set reinitialization using closest point method and kink detection for thin filaments, topology changes and two-phase flows *J. Comput. Phys.* **448** 110704
[50] Jain S S, Mani A and Moin P 2020 A conservative diffuse-interface method for compressible two-phase flows *J. Comput. Phys.* **418** 109606
[51] Janodet R, Guillamón C, Moureau V, Mercier R, Lartigue G, Bénard P, Ménard T and Berlemont A 2022 A massively parallel accurate conservative level set algorithm for simulating turbulent atomization on adaptive unstructured grids *J. Comput. Phys.* **458** 111075
[52] Jiang L and Chen S 2017 Parametric structural shape & topology optimization with a variational distance-regularized level set method *Comput. Methods Appl. Mech. Eng.* **321** 316–36
[53] Jibben Z and Herrmann M 2017 An arbitrary-order Runge–Kutta discontinuous Galerkin approach to reinitialization for banded conservative level sets *J. Comput. Phys.* **349** 453–73
[54] Josyula K, Rahul, and De S 2018 A level set approach for shock-induced α–γ phase transition of RDX *Comput. Mech.* **61** 19–32
[55] Karakus A, Chalmers N and Warburton T 2022 A local discontinuous Galerkin level set reinitialization with subcell stabilization on unstructured meshes *Comput. Math. Appl.* **123** 160–70
[56] Karakus A, Warburton T, Aksel M and Sert C 2016 A GPU accelerated level set reinitialization for an adaptive discontinuous Galerkin method *Comput. Math. Appl.* **72** 755–67
[57] Karakus A, Warburton T, Aksel M H and Sert C 2018 An adaptive fully discontinuous Galerkin level set method for incompressible multiphase flows *Int. J. Numer. Methods Heat Fluid Flow* **28** 1256–78
[58] Kinzel M P, Lindau J W and Kunz R F 2018 A multiphase level-set approach for all-Mach numbers *Comput. Fluids* **167** 1–16
[59] Kumar I, Bhatt C and Singh K U 2022 Entropy based automatic unsupervised brain intracranial hemorrhage segmentation using CT images *J. King Saud Univ., Comput. Inf. Sci.* **34** 2589–600
[60] Kumar R and Premachandran B 2022 A coupled level set and volume of fluid method for three dimensional unstructured polyhedral meshes for boiling flows *Int. J. Multiph. Flow* **156** 104207
[61] Kumar Singh N and Premachandran B 2018 A coupled level set and volume of fluid method on unstructured grids for the direct numerical simulations of two-phase flows including phase change *Int. J. Heat Mass Transfer* **122** 182–203
[62] Laadhari A, Saramito P, Misbah C and Székely G 2017 Fully implicit methodology for the dynamics of biomembranes and capillary interfaces by combining the level set and Newton methods *J. Comput. Phys.* **343** 271–99
[63] Laadhari A and Székely G 2017 Fully implicit finite element method for the modeling of free surface flows with surface tension effect *Int. J. Numer. Methods Eng.* **111** 1047–74
[64] Larios-Cárdenas L A and Gibou F 2021 A deep learning approach for the computation of curvature in the level-set method *SIAM J. Sci. Comput.* **43** A1754–79
[65] Larios-Cárdenas L A and Gibou F 2022 Error-correcting neural networks for semi-Lagrangian advection in the level-set method *J. Comput. Phys.* **471** 111623
[66] Larios-Cárdenas L A and Gibou F 2022 Error-correcting neural networks for two-dimensional curvature computation in the level-set method *J. Sci. Comput.* **93** 6

[67] Lee B, Darbon J, Osher S and Kang M 2017 Revisiting the redistancing problem using the Hopf–Lax formula *J. Comput. Phys.* **330** 268–81
[68] Lee C, Dolbow J and Mucha P J 2014 A narrow-band gradient-augmented level set method for multiphase incompressible flow *J. Comput. Phys.* **273** 12–37
[69] Li L, Xu X and Spagnolie S E 2017 A locally gradient-preserving reinitialization for level set functions *J. Sci. Comput.* **71** 274–302
[70] Li Q 2016 Numerical simulation of melt filling process in complex mold cavity with insets using IB-CLSVOF method *Comput. Fluids* **132** 94–105
[71] Lin J-Y, Shen Y, Ding H, Liu N-S and Lu X-Y 2017 Simulation of compressible two-phase flows with topology change of fluid–fluid interface by a robust cut-cell method *J. Comput. Phys.* **328** 140–59
[72] Lin S, Yan J, Kats D and Wagner G J 2019 A volume-conserving balanced-force level set method on unstructured meshes using a control volume finite element formulation *J. Comput. Phys.* **380** 119–42
[73] Liu F, Xu Y and Li Y 2017 A coupled level-set and volume-of-fluid method for simulating axi-symmetric incompressible two-phase flows *Appl. Math. Comput.* **293** 112–30
[74] Luddens F, Bergmann M and Weynans L 2015 Enablers for high-order level set methods in fluid mechanics *Int. J. Numer. Methods Fluids* **79** 654–75
[75] Luo J, Hu X and Adams N 2016 Efficient formulation of scale separation for multi-scale modeling of interfacial flows *J. Comput. Phys.* **308** 411–20
[76] Lyras K G, Hanson B, Fairweather M and Heggs P J 2020 A coupled level set and volume of fluid method with a re-initialisation step suitable for unstructured meshes *J. Comput. Phys.* **407** 109224
[77] Lyras P, Hubert A and Lyras K G 2023 A conservative level set method for liquid-gas flows with application in liquid jet atomisation *Exp. Comput. Multiph. Flow* **5** 67–83
[78] Majidi S and Afshari A 2015 Towards numerical simulations of supersonic liquid jets using ghost fluid method *Int. J. Heat Fluid Flow* **53** 98–112
[79] Majidi S and Afshari A 2016 A ghost fluid method for sharp interface simulations of compressible multiphase flows *J. Mech. Sci. Technol.* **30** 1581–93
[80] Marioni L, Khalloufi M, Bay F and Hachem E 2017 Two-fluid flow under the constraint of external magnetic field *Int. J. Numer. Methods Heat Fluid Flow* **27** 2565–81
[81] Meng W-K, Yu C-H, Li J and An R-D 2021 Numerical simulation of gas-liquid two-phase flow impacting fixed structure by CLSVOF/IB method based on OpenFOAM *J. Hydrodyn.* **33** 1176–89
[82] Meng W-K, Yu C-H, Li J and An R-D 2022 Numerical simulations of air-water flow and rigid-body motion based on two-liquid CLSVOF/IB method and overset mesh *Ocean Eng.* **256** 111455
[83] Mirjalili S, Ivey C B and Mani A 2020 A conservative diffuse interface method for two-phase flows with provable boundedness properties *J. Comput. Phys.* **401** 109006
[84] Mirzadeh M, Guittet A, Burstedde C and Gibou F 2016 Parallel level-set methods on adaptive tree-based grids *J. Comput. Phys.* **322** 345–64
[85] Mizuno K, Asahara M, Kamiya T and Miyasaka T 2022 Finite difference and reinitialization methods with level set to interfacial area transport equations for gas–liquid two-phase flows *Int. J. Comput. Fluid Dyn.* **36** 361–83
[86] Moghadam A M, Shafieefar M and Panahi R 2016 Development of a high-order level set method: compact conservative level set (CCLS) *Comput. Fluids* **129** 79–90
[87] Molina J and Ortiz P 2019 A conservative flux-corrected continuous finite element method for fluid interface dynamics *Int. J. Numer. Methods Fluids* **91** 287–310
[88] Morgan N R and Waltz J I 2017 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement *J. Comput. Phys.* **336** 492–512
[89] Mostafaiyan M, Wießner S, Heinrich G and Hosseini M S 2021 An improved conservative direct re-initialization method (ICDR) for two-phase flow simulations *Fluids* **6** 261
[90] Neviani M, Bagnerini P and Paladino O 2021 Gas bubble dynamics in airlift photo-bioreactors for microalgae cultivation by level set methods *Fuel* **292** 120402
[91] Ngo L C and Choi H G 2016 A local level set method based on a finite element method for unstructured meshes *J. Mech. Sci. Technol.* **30** 5539–45
[92] Ngo L C and Choi H G 2017 Efficient direct re-initialization approach of a level set method for unstructured meshes *Comput. Fluids* **154** 167–83

[93] Ngo L C and Choi H G 2020 A multi-level adaptive mesh refinement for an integrated finite element/level set formulation to simulate multiphase flows with surface tension *Comput. Math. Appl.* **79** 908–33
[94] Ngo L C, Choi H G and Chang K 2021 A coupled level set/volume of fluid method for simulation of two-phase flow on unstructured grids *J. Mech. Sci. Technol.* **35** 625–34
[95] Ngo L C, Dinh Q-N and Choi H G 2022 High-order level set reinitialization for multiphase flow simulations based on unstructured grids *Comput. Math. Appl.* **120** 60–77
[96] Nikitin K D, Olshanskii M A, Terekhov K M and Vassilevski Y V 2015 A splitting method for numerical simulation of free surface flows of incompressible fluids with surface tension *Comput. Methods Appl. Math.* **15** 59–77
[97] Ningegowda B and Premachandran B 2014 A coupled level set and volume of fluid method with multi-directional advection algorithms for two-phase flows with and without phase change *Int. J. Heat Mass Transfer* **79** 532–50
[98] Olsson E and Kreiss G 2005 A conservative level set method for two phase flow *J. Comput. Phys.* **210** 225–46
[99] Olsson E, Kreiss G and Zahedi S 2007 A conservative level set method for two phase flow II *J. Comput. Phys.* **225** 785–807
[100] Osher S and Sethian J A 1988 Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations *J. Comput. Phys.* **79** 12–49
[101] Pan S, Lyu X, Hu X Y and Adams N A 2018 High-order time-marching reinitialization for regional level-set functions *J. Comput. Phys.* **354** 311–9
[102] Parameswaran S and Mandal J 2023 A stable interface-preserving reinitialization equation for conservative level set method *Eur. J. Mech. B* **98** 40–63
[103] Park Y, hoon Song C, Hahn J and Kang M 2024 ReSDF: redistancing implicit surfaces using neural networks *J. Comput. Phys.* **502** 112803
[104] Peng D, Merriman B, Osher S, Zhao H and Kang M 1999 A PDE-based fast local level set method *J. Comput. Phys.* **155** 410–38
[105] Pertant S, Bernard M, Ghigliotti G and Balarac G 2021 A finite-volume method for simulating contact lines on unstructured meshes in a conservative level-set framework *J. Comput. Phys.* **444** 110582
[106] Prieto J L 2016 SLEIPNNIR: a multiscale, particle level set method for Newtonian and non-Newtonian interface flows *Comput. Methods Appl. Mech. Eng.* **307** 164–92
[107] Prieto J L and Carpio J 2019 A-SLEIPNNIR: a multiscale, anisotropic adaptive, particle level set framework for moving interfaces. Transport equation applications *J. Comput. Phys.* **377** 89–116
[108] Qin Z, Delaney K, Riaz A and Balaras E 2015 Topology preserving advection of implicit interfaces on Cartesian grids *J. Comput. Phys.* **290** 219–38
[109] Quell M, Diamantopoulos G, Hössinger A and Weinbub J 2021 Shared-memory block-based fast marching method for hierarchical meshes *J. Comput. Appl. Math.* **392** 113488
[110] Quezada de Luna M, Haydel Collins J and Kees C E 2020 An unstructured finite element model for incompressible two-phase flow based on a monolithic conservative level set method *Int. J. Numer. Methods Fluids* **92** 1058–80
[111] Quezada de Luna M, Kuzmin D and Kees C E 2019 A monolithic conservative level set method with built-in redistancing *J. Comput. Phys.* **379** 262–78
[112] Ramanuj V and Sankaran R 2019 High order anchoring and reinitialization of level set function for simulating interface motion *J. Sci. Comput.* **81** 1963–86
[113] Royston M, Pradhana A, Lee B, Chow Y T, Yin W, Teran J and Osher S 2018 Parallel redistancing using the Hopf–Lax formula *J. Comput. Phys.* **365** 7–17
[114] Russo G and Smereka P 2000 A remark on computing distance functions *J. Comput. Phys.* **163** 51–67
[115] Sabelnikov V, Ovsyannikov A Y and Gorokhovski M 2014 Modified level set equation and its numerical assessment *J. Comput. Phys.* **278** 1–30
[116] Sahut G, Ghigliotti G, Balarac G, Bernard M, Moureau V and Marty P 2021 Numerical simulation of boiling on unstructured grids *J. Comput. Phys.* **432** 110161
[117] Sandberg M, Hattel J H and Spangenberg J 2019 Simulation of liquid composite moulding using a finite volume scheme and the level-set method *Int. J. Multiph. Flow* **118** 183–92
[118] Satheesh K, Srinivas G and Hemchandra S 2018 An advection velocity correction scheme for interface tracking using the level-set method *Comput. Fluids* **168** 232–44

[119] Sato K and Koshimura S 2023 Development of a single-phase free-surface flow model with the improved lattice kinetic scheme *Comput. Math. Appl.* **145** 275–88
[120] Saye R 2017 Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction and free surface flow: part I *J. Comput. Phys.* **344** 647–82
[121] Saye R 2017 Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction and free surface flow: part II *J. Comput. Phys.* **344** 683–723
[122] Shakoor M 2021 FEMS—a mechanics-oriented finite element modeling software *Comput. Phys. Commun.* **260** 107729
[123] Shakoor M and Park C H 2021 A higher-order finite element method with unstructured anisotropic mesh adaption for two phase flows with surface tension *Comput. Fluids* **230** 105154
[124] Shakoor M, Scholtes B, Bouchard P-O and Bernacki M 2015 An efficient and parallel level set reinitialization method—application to micromechanics and microstructural evolutions *Appl. Math. Modelling* **39** 7291–302
[125] Shao C, Yuan S, Chai M, Jin T and Luo K 2023 A generalized variational level set method without frequent reinitialization for simulations of gas-liquid flows *J. Comput. Phys.* **495** 112558
[126] Shervani-Tabar N and Vasilyev O V 2018 Stabilized conservative level set method *J. Comput. Phys.* **375** 1033–44
[127] Shi W, Xu J and Shu S 2017 An adaptive semi-Lagrangian level-set method for convection-diffusion equations on evolving interfaces *Adv. Appl. Math. Mech.* **9** 1364–82
[128] Shi W, Xu J-J and Shu S 2017 A simple implementation of the semi-Lagrangian level-set method *Adv. Appl. Math. Mech.* **9** 104–24
[129] Shiratori S, Usui T, Koyama S, Ozawa S, Nagano H and Shimano K 2021 Efficient implementation of two-phase flow solver based on THINC/SW and S-CLSVOF on unstructured meshes *Int. J. Microgravity Sci. Appl.* **38** 380301
[130] Shukla R K 2014 Nonlinear preconditioning for efficient and accurate interface capturing in simulation of multicomponent compressible flows *J. Comput. Phys.* **276** 508–40
[131] Solomenko Z, Spelt P D M and Alix P 2017 A level-set method for large-scale simulations of three-dimensional flows with moving contact lines *J. Comput. Phys.* **348** 151–70
[132] Solomenko Z, Spelt P D M, Ó Náraigh L and Alix P 2017 Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: a comparative study *Int. J. Multiph. Flow* **95** 235–56
[133] Spelt P D M 2005 A level-set approach for simulations of flows with multiple moving contact lines with hysteresis *J. Comput. Phys.* **207** 389–404
[134] Starinshak D P, Karni S and Roe P L 2014 A new level-set model for the representation of non-smooth geometries *J. Sci. Comput.* **61** 649–72
[135] Sufyan M, Ngo L C and Choi H G 2016 A comparative study of reinitialization approaches of the level set method for simulating free-surface flows *J. Mech. Sci. Technol.* **30** 1681–9
[136] Sussman M and Fatemi E 1999 An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow *SIAM J. Sci. Comput.* **20** 1165–91
[137] Sussman M, Smereka P and Osher S 1994 A level set approach for computing solutions to incompressible two-phase flow *J. Comput. Phys.* **114** 146–59
[138] Torres-Ulloa C, Berres S and Grassia P 2018 Foam–liquid front motion in Eulerian coordinates *Proc. R. Soc. A* **474** 20180290
[139] Touré M K and Soulaïmani A 2016 Stabilized finite element methods for solving the level set equation without reinitialization *Comput. Math. Appl.* **71** 1602–23
[140] Tsui Y-Y, Liu C-Y and Lin S-W 2017 Coupled level-set and volume-of-fluid method for two-phase flow calculations *Numer. Heat Transfer B* **71** 173–85
[141] Turnquist B and Owkes M 2019 multiUQ: an intrusive uncertainty quantification tool for gas-liquid multiphase flows *J. Comput. Phys.* **399** 108951
[142] Utz T, Kummer F and Oberlack M 2017 Interface-preserving level-set reinitialization for DG-FEM *Int. J. Numer. Methods Fluids* **84** 183–98
[143] Valette R, Pereira A, Riber S, Sardo L, Larcher A and Hachem E 2021 Viscoplastic dam-breaks *J. Non-Newton. Fluid Mech.* **287** 104447
[144] Ville L, Silva L and Coupez T 2011 Convected level set method for the numerical simulation of fluid buckling *Int. J. Numer. Methods Fluids* **66** 324–44

[145] Vivacqua V, Ghadiri M, Abdullah A, Hassanpour A, Al-Marri M, Azzopardi B, Hewakandamby B and Kermani B 2016 Analysis of partial electrocoalescence by Level-Set and finite element methods *Chem. Eng. Res. Des.* **114** 180–9
[146] Vukčević V, Jasak H and Malenica Š 2016 Decomposition model for naval hydrodynamic applications, part I: computational method *Ocean Eng.* **121** 37–46
[147] Wacławczyk T 2015 A consistent solution of the re-initialization equation in the conservative level-set method *J. Comput. Phys.* **299** 487–525
[148] Wacławczyk T 2021 Modeling of non-equilibrium effects in intermittency region between two phases *Int. J. Multiph. Flow* **134** 103459
[149] Wang Y, Shu C, Yang L and Yuan H 2018 On the re-initialization of fluid interfaces in diffuse interface method *Comput. Fluids* **166** 209–17
[150] Weinbub J and Hössinger A 2014 Accelerated redistancing for level set-based process simulations with the fast iterative method *J. Comput. Electron.* **13** 877–84
[151] Wen H, Sheu T W H and Yu C 2018 A combined compact finite difference scheme for predicting the evolution of a mean curvature driven interface *Comput. Fluids* **170** 299–312
[152] Xie B, Jin P, Du Y and Liao S 2020 A consistent and balanced-force model for incompressible multiphase flows on polyhedral unstructured grids *Int. J. Multiph. Flow* **122** 103125
[153] Xie B, Jin P, Nakayama H, Liao S and Xiao F 2020 A conservative solver for surface-tension-driven multiphase flows on collocated unstructured grids *J. Comput. Phys.* **401** 109025
[154] Xin J, Shi F, Jin Q and Ma L 2020 Gradient-augmented level set two-phase flow method with pretreated reinitialization for three-dimensional violent sloshing *J. Fluids Eng.* **142** 011402
[155] Xiong Y, Xiao F and Xie B 2022 A hybrid volume of fluid and level set interface capturing scheme with quartic surface representation for unstructured meshes *Int. J. Numer. Methods Fluids* **94** 1542–65
[156] Xu S and Ren W 2016 Reinitialization of the Level-Set function in 3D simulation of moving contact lines *Commun. Comput. Phys.* **20** 1163–82
[157] Xu S, Zhu Q, Fernando M and Sundar H 2022 A finite element level set method based on adaptive octree meshes for thermal free-surface flows *Int. J. Numer. Methods Eng.* **123** 5500–16
[158] Xue T, Sun W, Adriaenssens S, Wei Y and Liu C 2021 A new finite element level set reinitialization method based on the shifted boundary method *J. Comput. Phys.* **438** 110360
[159] Yan J, Deng X, Korobenko A and Bazilevs Y 2017 Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines *Comput. Fluids* **158** 157–66
[160] Yang A, Chen S, Yang L and Yang X 2014 An upwind finite volume method for incompressible inviscid free surface flows *Comput. Fluids* **101** 170–82
[161] Yang J and Stern F 2017 A highly scalable massively parallel fast marching method for the Eikonal equation *J. Comput. Phys.* **332** 333–62
[162] Yu C and Sheu T W H 2017 Development of a coupled level set and immersed boundary method for predicting dam break flows *Comput. Phys. Commun.* **221** 1–18
[163] Yu C, Wu T, An R and Li Y 2022 Numerical simulation for liquid sloshing with baffle by the CLSVOF/IB method *Ocean Eng.* **258** 111732
[164] Yu C, Ye Z, Sheu T W H, Lin Y and Zhao X 2016 An improved interface preserving level set method for simulating three dimensional rising bubble *Int. J. Heat Mass Transfer* **103** 753–72
[165] Zeng Y, Liu H, Gao Q, Almgren A, Bhalla A P S and Shen L 2023 A consistent adaptive level set framework for incompressible two-phase flows with high density ratios and high Reynolds numbers *J. Comput. Phys.* **478** 111971
[166] Zeng Y, Xuan A, Blaschke J and Shen L 2022 A parallel cell-centered adaptive level set framework for efficient simulation of two-phase flows with subcycling and non-subcycling *J. Comput. Phys.* **448** 110740
[167] Zhang H, Zhao L, Mao J and Liu X 2020 An efficient 3D iterative interface-correction reinitialization for the level set method *Comput. Fluids* **213** 104724
[168] Zhang J and Yue P 2019 A high-order and interface-preserving discontinuous Galerkin method for level-set reinitialization *J. Comput. Phys.* **378** 634–64
[169] Zhang J and Yue P 2020 A level-set method for moving contact lines with contact angle hysteresis *J. Comput. Phys.* **418** 109636
[170] Zhang T and Wolgemuth C W 2020 Sixth-order accurate schemes for reinitialization and extrapolation in the level set framework *J. Sci. Comput.* **83** 26
[171] Zhang Y, Liang B and Ni J 2020 Numerical study of bubble rising motion in a vertical wedge-shaped channel based on a modified level set method *Fluid Dyn.* **55** 241–51

[172] Zhang Y and Ye W 2016 A high-order level-set method with enhanced stability for curvature driven flows and surface diffusion motion *J. Sci. Comput.* **69** 1316–45
[173] Zhao L, Khuc H, Mao J, Liu X and Avital E 2018 One-layer particle level set method *Comput. Fluids* **170** 141–56
[174] Zhao L, Mao J, Bai X, Liu X, Li T and Williams J 2014 Finite element implementation of an improved conservative level set method for two-phase flow *Comput. Fluids* **100** 138–54
[175] Zhao L, Zhang H, Mao J, Zhu H, Peng D and Mu K 2019 A simple iterative geometry-based interface-preserving reinitialization for the level set method *Int. J. Comput. Fluid Dyn.* **33** 371–92
[176] Zhao Q, Xu S and Ren W 2022 A Level Set method for the simulation of moving contact lines in three dimensions *Commun. Comput. Phys.* **32** 1310–31
[177] Zhao X, Ma W and Wang K 2023 Simulating laser-fluid coupling and laser-induced cavitation using embedded boundary and level set methods *J. Comput. Phys.* **472** 111656
[178] Zhao Z and Yan J 2021 Variational multi-scale modeling of interfacial flows with a balanced-force surface tension model *Mech. Res. Commun.* **112** 103608
[179] Zhu L, Goraya S A and Masud A 2019 Interface-capturing method for free-surface plunging and breaking waves *J. Eng. Mech.* **145** 04019088
[180] Zhu Q, Xu F, Xu S, Hsu M-C and Yan J 2020 An immersogeometric formulation for free-surface flows with application to marine engineering problems *Comput. Methods Appl. Mech. Eng.* **361** 112748
[181] Zhu Q and Yan J 2021 A mixed interface-capturing/interface-tracking formulation for thermal multi-phase flows with emphasis on metal additive manufacturing processes *Comput. Methods Appl. Mech. Eng.* **383** 113910