Journal of Computational Physics 566 (2026) 115282

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A predictor-corrector scheme for approximating signed distances using finite element methods

Amina El Bachari [a,b,*], Johann Rannou [c], Vladislav Yastrebov [b], Pierre Kerfriden [b], Susanne Claus [a]

[a] DTIS, ONERA, Université Paris-Saclay, Palaiseau, 91120, France
[b] Centre de Matériaux, MINES Paris-PSL, CNRS UMR 7633, Satory, 78000, France
[c] DMAS, ONERA, Université Paris-Saclay, Châtillon, 92320, France

ARTICLE INFO

Keywords:
Signed distance function
Reinitialization
Finite element method
Level set
Parallelization

ABSTRACT

In this article, we introduce a finite element method designed for the robust computation of approximate signed distance functions to arbitrary boundaries in two and three dimensions. Our method employs a novel prediction-correction approach, involving first the solution of a linear diffusion-based prediction problem, followed by a nonlinear minimization-based correction problem associated with the eikonal equation. The prediction step efficiently generates a suitable initial guess, significantly facilitating convergence of the nonlinear correction step.

We validate the robustness of our approach using complex initial level set functions featuring arbitrary steep or flat regions in both two and three dimensions, as well as transient test cases like the vortex test. We compare our results with a standard finite difference Hamilton-Jacobi scheme, demonstrating improved interface preservation. Furthermore, we illustrate the method's scalability for large-scale 3D problems using distributed memory parallelism (MPI), establishing its accuracy, efficiency, and broad applicability for reinitializing diverse level set functions.

1. Introduction

The computation of signed distance functions (SDFs) plays a critical role in numerous computational methods, particularly within computational mechanics, computer graphics, and numerical simulations involving interface capturing and tracking [1,2]. Accurate and efficient methods to compute SDFs are crucial for applications such as shape optimization [3,4], multi-phase flow [5–7] or image processing [8,9]. The signed distance function is typically defined as the solution of the eikonal equation, a nonlinear partial differential equation whose numerical solution poses significant computational challenges [10].

Traditional methods to compute SDFs generally fall into two main categories: geometric algorithms and PDE-based methods. Geometric algorithms include fast marching methods [2,10–12], which compute the distance by efficiently advancing distance values outwards from the boundary, and fast sweeping methods [13–15], which solve the problem by iteratively updating distance values across the grid in alternating directions. These methods are highly efficient and robust but are challenging to parallelize [16] and to extend to unstructured grids [17,18] or to higher-order geometric representations. PDE-based methods, which solve variants of the eikonal equation directly using finite element, finite volume or finite difference discretizations, offer greater flexibility for complex domains but require careful treatment to ensure stability and accuracy [1].

* Corresponding author.
E-mail address: amina.el_bachari@minesparis.psl.eu.com (A. El Bachari).

https://doi.org/10.1016/j.jcp.2026.115282
Received 17 July 2025; Received in revised form 2 July 2026; Accepted 3 August 2026
Available online 7 August 2026
0021-9991/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Among these, the time-dependent Hamilton-Jacobi formulation introduced by Sussman et al. [5] is the most effective and widely used technique. This approach evolves the level set function to a steady state via the redistancing equation

$$\partial_t \phi + \mathrm{sgn}(\phi_0)(|\nabla \phi| - 1) = 0. \tag{1}$$

While robust, standard discretizations may suffer from mass loss or interface displacement [19]. To address these issues, various improvements have been proposed, such as higher-order ENO/WENO schemes or the subcell fixation method by Russo and Smereka [20,21], which preserves the interface location by modifying the numerical flux near the zero level set.

Solving the time-dependent Hamilton-Jacobi equation works well for conservative methods such as Finite Volume and Finite Difference schemes [5]. However, its application using finite element methods presents several difficulties due to the hyperbolic nature of the equation. Standard Galerkin formulations are unstable and require consistent stabilization techniques such as Streamline Upwind Petrov-Galerkin (SUPG), Galerkin Least Squares (GLS) or Discontinuous Galerkin formulations [22] to prevent spurious oscillations. These stabilization techniques introduce artificial diffusion, which often leads to interface shifting and significant mass loss during the reinitialization process [23]. To overcome these issues and avoid the complexities of stabilizing hyperbolic PDEs, alternative formulations have been explored.

In this article, we propose a PDE-based finite element approach that builds upon the minimization-based reinitialization framework established in [23–25]. These pioneering works formulated the reinitialization problem as a least-squares minimization of the deviation of the gradient norm from unity. While effective, the original formulation faces challenges in regions where the gradient norm vanishes (flat regions), leading to singularities. To address this, previous studies proposed modifying the diffusion coefficient or the potential function to regularize the problem [23,24]. However, these modifications can alter the nature of the equation being solved or fail to fully resolve convergence issues for arbitrary initial conditions.

Our contribution is to propose a robust predictor-corrector strategy that preserves the original, consistent minimization problem while ensuring convergence through a novel initialization step. Instead of modifying the eikonal equation itself, we introduce a linear diffusion-based predictor with a sign-based source term and sign-based Neumann boundary condition. Our formulation is motivated by deriving the analytical solution for a simple 1D problem and making appropriate choices for the right hand side of the diffusion problem as well as the Neumann boundaries. Our predictor problem efficiently generates a suitable initial guess that eliminates flat and steep regions, thereby providing a robust starting point for the unmodified minimization problem. Furthermore, we extend this framework to unfitted meshes using the CutFEM method [26], allowing for the accurate reinitialization of level set functions on complex, non-conforming geometries without the need for remeshing. We derive the minimization (corrector) problem in the weak formulation for the cut finite element method including Nitsche boundary conditions for the zero interface.

Our work shares structural similarities with the recent study by Hashemi et al. [27], who proposed an optimization-based reinitialization approach utilizing a similar two-step strategy involving a primary initialization followed by a secondary optimization. Both methods aim to preserve the interface and use elliptic formulations to avoid the instabilities associated with hyperbolic Hamilton-Jacobi equations. Furthermore, both approaches address the challenge of equidistant points where the gradient vanishes. However, distinct differences exist in the formulation of the first step and the interface preservation strategy. Hashemi et al. employ a geometric reconstruction scheme that relies on curvature and complex boundary condition estimations. In contrast, we propose a purely diffusive predictor with a sign-based source term that avoids explicit geometric calculations. Regarding interface preservation, while Hashemi et al. utilize an efficient element splitting technique, our method leverages Nitsche's method combined with runtime quadrature on the unfitted interface within a CutFEM framework [26] without splitting the element. The implementation of our method is done in CutFEMx [28] which extends FEniCSx [29] to immersed finite element methods. All numerical results in this article can be reproduced using the source code at [30].

We validate our method through a comprehensive numerical study. To motivate the design of our predictor step, we first analyze a simple 1D example and compare our approach with existing nonlinear diffusion regularizations [23,24]. We demonstrate that while previous methods struggle to recover the signed distance property in flat regions where the gradient vanishes, our predictor-corrector scheme robustly eliminates these singularities. Additionally, we provide a comparative benchmark against a standard finite difference Hamilton-Jacobi solver to highlight the improved interface fidelity of our proposed method. The method is then thoroughly benchmarked in Section 5 against shapes in 2D and 3D, including a circle, a discontinuous step function, a noisy interface, a star-shaped domain with sharp corners, and a 3D torus. Furthermore, we demonstrate the robustness of the scheme on the 2D Rider-Kothe vortex test, which assesses the method's performance under interface stretching and topological deformation. These experiments confirm the method's optimal convergence order, its ability to handle non-smooth and noisy data, and its scalability to large-scale 3D problems utilizing distributed memory parallelism (MPI).

The remainder of this article is organized as follows. In Section 2, we introduce the mathematical formulation of the signed distance function and corresponding signed distance problem. In Section 3 we discuss the minimization-based eikonal problem and introduce its finite element formulation. Section 4 presents the prediction-correction algorithm and its finite element discretization. In Section 5, we validate our predictor-corrector scheme on a range of numerical experiments in both two and three dimensions.

2. Level set method

2.1. Level sets and signed distance functions.

Let us consider a fixed domain $D \subset \mathbb{R}^n$ containing domain Ω defined by a level set function, $\phi : D \to \mathbb{R}$ as:

$$\begin{cases} \phi(x) < 0 & \text{if } x \in \Omega, \\ \phi(x) = 0 & \text{if } x \in \Gamma, \\ \phi(x) > 0 & \text{if } x \in D \setminus \Omega, \end{cases} \tag{2}$$

with $\Gamma = \{x \in D, \text{ such that } x \in \partial\Omega \setminus \partial D\}$.
A level set function with signed distance property with respect to $\phi(x) = 0$ is defined as:

$$\phi(x) = \begin{cases} -d(x,\Gamma) & \text{if } x \in \Omega, \\ d(x,\Gamma) & \text{if } x \in D \setminus \{\Omega \cup \Gamma\}, \end{cases} \tag{3}$$

where d is the euclidean distance function defined as:

$$d(x,\Gamma) = \inf_{y \in \Gamma} d(x,y). \tag{4}$$

If the domain $\Omega \subset \mathbb{R}^n$ has a piecewise smooth boundary, then the signed distance function to domain boundary Γ is differentiable almost everywhere, and the equivalent signed distance property (SDP) holds:

$$|\nabla\phi| = 1, \text{ almost everywhere in } D, \tag{5}$$

where $|\cdot|$ denotes the Euclidean norm, defined by:

$$|\nabla\phi| = \left(\sum_{i=1}^{n} \left(\frac{\partial \phi}{\partial x_i} \right)^2 \right)^{1/2}. \tag{6}$$

The normal to the contour lines of a locally smooth level set function is given by:

$$\mathbf{n}_\phi = \frac{\nabla\phi}{|\nabla\phi|}, \tag{7}$$

which can be simplified as $\mathbf{n}_\phi = \nabla\phi$, almost everywhere, for a level set function with signed distance property.

2.2. Need for approximate redistancing

In general, the level set method is employed in the context of moving boundary problems. In this framework, an advection equation is used to advect the domain boundary Γ in the normal direction, with velocity given by $\theta = v\mathbf{n}_\phi$. Regardless of the method used for the advection of the level set, after one advection increment, the signed distance property $|\nabla\phi| = 1$ is generally not preserved. Nevertheless, maintaining the signed distance property is essential to

(i) compute normals on Γ, $\mathbf{n}_\phi = \nabla\phi/|\nabla\phi|$, which becomes inaccurate when $|\nabla\phi|$ becomes too small or large;
(ii) stabilize level set advection, as the convection term $v\mathbf{n}_\phi \cdot \nabla\phi = v|\nabla\phi|$ becomes very large for steep ϕ;
(iii) reconstruct the interface by linear interpolation, which becomes inaccurate for flat ϕ as small changes in ϕ values yield large differences in determining the location of the boundary.

The extensive use of the level set method for moving boundary problems, such as shape optimization, fluid simulation, and propagation phenomena, has driven active research into reinitialization methods tailored to the specific requirements of each application.

However, the need for redistancing extends beyond moving boundary problems. For example, in the context of solving partial differential equations on domains derived from image segmentation, it is often more efficient to define the domain boundaries using a zero level set rather than relying on pixel or voxel edges [31]. In such cases, the level set function is typically initialized as a constant value inside and outside the domain, with a piecewise polynomial transition in a narrow band around the interface. Consequently, it lacks the signed distance property. Similarly, in mesh generation, signed distance functions are essential for representing geometry and guiding adaptive mesh refinement [32]. More broadly, signed distance functions are ubiquitous in computer graphics applications [33].

2.3. Signed distance problem

Starting from a level set function ϕ_0 that does not have the signed distance property, we seek to determine the corresponding signed distance function denoted by ϕ that satisfies the following properties:

$$\begin{cases} |\nabla\phi| &= 1 & \text{in } D, \\ \phi(x) &= 0 & \text{on } \Gamma, \\ \operatorname{sgn}(\phi) &= \operatorname{sgn}(\phi_0) & \text{in } D \setminus \Gamma, \end{cases} \tag{8}$$

where sgn denotes the sign function, defined as:

$$\operatorname{sgn}(\phi) = \begin{cases} 1 & \text{for } \phi(x) > 0, \\ -1 & \text{for } \phi(x) < 0, \\ 0 & \text{for } \phi(x) = 0. \end{cases} \tag{9}$$

To solve this problem, reinitialization methods are used. A wide range of methods are proposed in the literature; in the following Section, we will present the used minimization based method.

3. Minimization based reinitialization

3.1. Least squares minimization

Our redistancing method is inspired by Basting and Kuzmin [23] who formulated the eikonal equation, $|\nabla\phi| = 1$, as a least squares minimization problem

$$\min\left(\int_D \frac{1}{2}(|\nabla\phi| - 1)^2\, dx\right). \tag{10}$$

Setting

$$\mathcal{F}(\phi) = \left(\int_D \frac{1}{2}(|\nabla\phi| - 1)^2\, dx\right) \tag{11}$$

and taking the Gâteaux derivative of $\mathcal{F}(\phi)$ in direction ψ, we obtain

$$d\mathcal{F}(\phi;\psi) = \int_D (|\nabla\phi| - 1)\frac{\nabla\phi}{|\nabla\phi|}\nabla\psi\, dx \tag{12}$$

$$= \int_D \left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right)\nabla\psi\, dx. \tag{13}$$

Therefore, the minimization problem becomes

$$\int_D \left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right)\nabla\psi\, dx = 0. \tag{14}$$

We can derive the strong form by integration by parts

$$-\int_D \nabla\cdot\left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right)\psi\, dx + \int_{\partial D}\left(\left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right)\cdot\mathbf{n}\right)\psi\, ds = 0, \tag{15}$$

and setting

$$\left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right)\cdot\mathbf{n} = 0 \text{ on } \partial D, \tag{16}$$

which yields

$$\begin{cases} -\nabla\cdot\left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right) &= 0 \quad \text{in } D, \\ \left(\nabla\phi - \frac{\nabla\phi}{|\nabla\phi|}\right)\cdot\mathbf{n} &= 0 \quad \text{on } \partial D. \end{cases} \tag{17}$$

Here, $\mathbf{n}$ denotes the outward normal to the boundary ∂D. To obtain the distance to the boundary defined by Γ, we impose the following Dirichlet condition:

$$\phi = 0 \text{ on } \Gamma. \tag{18}$$

This problem is a non-linear diffusion problem and hence needs to be linearized and solved iteratively. These iterative schemes suffer from convergence issues because the diffusion term $1/|\nabla\phi| \to \infty$ at points and regions where $\nabla\phi \approx 0$. In particular, Newton-Raphson schemes diverge. Picard iterations display slow convergence near these points but deliver a very good signed distance approximation elsewhere, which is analyzed in detail in the following Sections. Hence, we reformulate the problem (17) as a Picard iteration

$$\begin{cases} -\nabla\cdot\nabla\phi^{n+1} &= -\nabla\cdot\dfrac{\nabla\phi^n}{|\nabla\phi^n|} \quad \text{in } D \\ \nabla\phi^{n+1}\cdot\mathbf{n} &= \dfrac{\nabla\phi^n}{|\nabla\phi^n|}\cdot\mathbf{n} \quad \text{on } \partial D. \end{cases} \tag{19}$$

Here, n denotes the n-th iteration. The Picard iteration does not diverge globally but it does not converge in points with

$$|\nabla\phi^n| \approx 0. \tag{20}$$

One way to prevent this is to reformulate the original problem as suggested by [23–25] and detailed in the next Section. In [25], it is proposed to solve the Laplace equation in a neighborhood of the interface. The conditions on this neighborhood are chosen so as to preserve the sign of the initial level set. Nevertheless, this method cannot be extended to the entire domain D. In this article, however, we propose to address this shortcoming by carefully designing a predictor problem which we solve to obtain a suitable initial guess ϕ^0 without large flat regions. Before discussing our method, we will briefly review the reformulations of [23–25] and show why we needed to design a predictor problem instead.

3.2. Modified minimization and non-linear diffusion formulation

In order to prevent $1/|\nabla\phi| \to \infty$, [23] and [24] propose to use another function, $p(|\nabla\phi|)$, in

$$\mathcal{F}(p,\phi) = \int_\Omega p(|\nabla\phi|)\, dx. \tag{21}$$

For which the Gâteaux derivative is given by:

$$d\mathcal{F}(\phi;\psi) = \int_D \frac{p'(|\nabla\phi|)}{|\nabla\phi|}\nabla\phi\cdot\nabla\psi\, dx. \tag{22}$$

Looking for the zero of this Gâteaux derivative can be interpreted as a non-linear diffusion problem

$$\int_D \widetilde{d}(|\nabla\phi|)\nabla\phi\cdot\nabla\psi\, dx = 0, \tag{23}$$

where the non-linear diffusion term is given by:

$$\widetilde{d}(|\nabla\phi|) := \frac{p'(|\nabla\phi|)}{|\nabla\phi|}. \tag{24}$$

The strong form for this non-linear diffusion problem becomes:

$$\begin{cases} -\nabla\cdot\left(\widetilde{d}(|\nabla\phi|)\nabla\phi\right) &= 0 \text{ in } D, \\ \left(\widetilde{d}(|\nabla\phi|)\nabla\phi\right)\cdot n &= 0 \text{ on } \partial D. \end{cases} \tag{25}$$

Basting et al. [23] introduced the following double well potential inspired function for $|\nabla\phi| \le 1$:

$$p(|\nabla\phi|) := \begin{cases} \frac{1}{2}(|\nabla\phi|-1)^2 & \text{if } |\nabla\phi| > 1, \\ \frac{1}{2}|\nabla\phi|^2(|\nabla\phi|-1)^2 & \text{if } |\nabla\phi| \le 1. \end{cases} \tag{26}$$

Note that the original function for $|\nabla\phi| > 1$ is kept. The diffusion term for this function becomes

$$\widetilde{d}(|\nabla\phi|) := \begin{cases} 1-\frac{1}{|\nabla\phi|} & \text{if } |\nabla\phi| > 1, \\ 1-(3|\nabla\phi|-2|\nabla\phi|^2) & \text{if } |\nabla\phi| \le 1. \end{cases} \tag{27}$$

Adams et al. [24] pointed out that for $|\nabla\phi| < 0.5$ this diffusion term becomes positive meaning that an already flat functional profile is pushed towards $|\nabla\phi| = 0$ in any iterative scheme. In the original formulation the diffusion term is positive for $|\nabla\phi| > 1$, which causes the function profile to flatten and it is negative for $|\nabla\phi| < 1$ which causes the function profile to steepen through the iterations. Adams et al. [24] suggested the following modification of the function:

$$p(|\nabla\phi|) := \begin{cases} \frac{1}{2}\left(1-\frac{1}{|\nabla\phi|}\right) & \text{if } |\nabla\phi| > 1, \\ \frac{(|\nabla\phi|)^3}{3} - \frac{(|\nabla\phi|)^2}{2} + \frac{1}{6} & \text{if } |\nabla\phi| \le 1, \end{cases} \tag{28}$$

and the corresponding diffusion term becomes:

$$\widetilde{d}(|\nabla\phi|) := \begin{cases} 1-\frac{1}{|\nabla\phi|} & \text{if } |\nabla\phi| > 1, \\ 1-(2-|\nabla\phi|) & \text{if } |\nabla\phi| \le 1. \end{cases} \tag{29}$$

Authors of [23,24] solve this nonlinear problem by Picard iteration, which for the strong form can be written as:

$$\begin{aligned} -\nabla\cdot\nabla\phi^{n+1} &= -\nabla\cdot\left(\widetilde{d}^*(|\nabla\phi^n|)\nabla\phi^n\right) && \text{in } D, \\ \nabla\phi^{n+1}\cdot\mathbf{n} &= \left(\widetilde{d}^*(|\nabla\phi^n|)\nabla\phi^n\right)\cdot\mathbf{n} && \text{on } \partial D. \end{aligned} \tag{30}$$

For the original problem (17), as well as for the modified formulations (27) and (29), the diffusion coefficient $\widetilde{d}^*(|\nabla\phi|)$ is defined piecewise depending on the gradient magnitude $|\nabla\phi|$. For $|\nabla\phi| > 1$, the diffusion term is given by:

$$\widetilde{d}^*(|\nabla\phi|) = \frac{1}{|\nabla\phi|}. \tag{31}$$

For $|\nabla\phi| \le 1$, the form of $\widetilde{d}^*$ depends on the chosen approximation. For the method defined by Eq. (27), the diffusion term becomes

$$\widetilde{d}^*(|\nabla\phi|) = 3|\nabla\phi| - 2|\nabla\phi|^2, \tag{32}$$

while for the alternative formulation (29), it reads:

$$\widetilde{d}^*(|\nabla\phi|) = 2 - |\nabla\phi|. \tag{33}$$

In the following Section, we will discuss the finite element formulation of the non-linear diffusion problem (17), as well as for the modified formulations (27) and (29).

3.3. Finite element formulation for the non-linear diffusion problem

Let $\phi^0 : D \to \mathbb{R}$ be a function which split our domain D into two parts: a negative part Ω^- and a positive part Ω^+ given by the sign of the function ϕ^0, and an interface between Ω^- and Ω^+, denoted by Γ, given by the zero contour line of ϕ^0. This means

$$\phi^0(x) > 0 \quad \text{for } x \in \Omega^+, \tag{34}$$

$$\phi^0(x) = 0 \quad \text{for } x \in \Gamma, \tag{35}$$

$$\phi^0(x) < 0 \quad \text{for } x \in \Omega^-. \tag{36}$$

We do not make any further assumptions on ϕ^0. Let us now presume that D is decomposed into mesh elements (denoted K) whose faces coincide with the discrete interface Γ_h, which is a piecewise linear approximation of the continuous interface Γ. We denote this mesh by $\mathcal{K}_h$ and the discretized domain by $D_h = \Omega_h^+ \cup \Omega_h^-$. In later Sections in this article, we will discuss how this scheme can be extended to meshes that do not coincide with Γ_h.

Furthermore let V_h^0 denote the continuous piecewise linear finite element space which includes the zero Dirichlet boundary condition on ϕ along Γ_h, i.e.

$$V_h^0 := \left\{ v \in H^1(D) \cap C^0(D) \mid v_{|K} \in \mathbb{P}_1(K) \text{ for all } K \in \mathcal{K}_h,\ v_{|F} = 0 \text{ for all } F \in \Gamma_h \right\}. \tag{37}$$

Our finite element formulation to solve the non-linear diffusion problem (30) then reads: For $n = 0, \dots, N-1$, find $\phi_h^{n+1} \in V_h^0$ such that for all $v_h \in V_h^0$

$$\int_{D_h} \nabla \phi_h^{n+1} \nabla v_h \, dx = \int_{D_h} \left(\widetilde{d^*}\left(\left| \nabla \phi_h^n \right| \right) \nabla \phi_h^n \right) \nabla v_h \, dx. \tag{38}$$

Note that the Neumann boundary condition from (30) is implicitly enforced through first integrating by parts and then setting the boundary term to zero. And the Dirichlet condition is strongly enforced through the finite element space. The number of iterations, N, is either fixed or we are iterating until the following inequality is satisfied:

$$\left| \left\| \left(|\nabla \phi^{n+1}| - 1 \right) \right\|_{L^2(D)} - \| (|\nabla \phi^n| - 1) \|_{L^2(D)} \right| < \varepsilon, \tag{39}$$

with a small positive real value for ε. The following Section shows a comparison of the results obtained for different diffusion schemes for a one-dimensional test case.

3.3.1. Comparison of different non-linear diffusion schemes for one-dimensional test case

Let us assume an initial function ϕ^0 in a domain $D = [0, 1]$ defined as:

$$\phi^0(x) = \begin{cases} 0.6x - 0.25, & \text{if } x \in \left[0, \frac{1}{3}\right], \\ 0.3x - 0.15, & \text{if } x \in \left[\frac{1}{3}, \frac{2}{3}\right], \\ 0.05, & \text{if } x \in \left[\frac{2}{3}, 1\right], \end{cases} \tag{40}$$

displayed in Fig. 1a. This initial function features three different slopes and will be used to compare the results obtained for different potentials proposed by Kuzmin and Basting [23], Adams et al. [24], and for our prediction-correction method introduced in this article. The results are shown in Fig. 1b–d for three different methods, after 10 fixed point iterations.

First, the simulation using function (26) is conducted. In the subdomain $\left[\frac{1}{3}, 1\right]$, where the level set is initialized with a function whose gradient is less than 0.5, the solution iterates towards a function with a zero gradient norm, as shown in Fig. 1b.

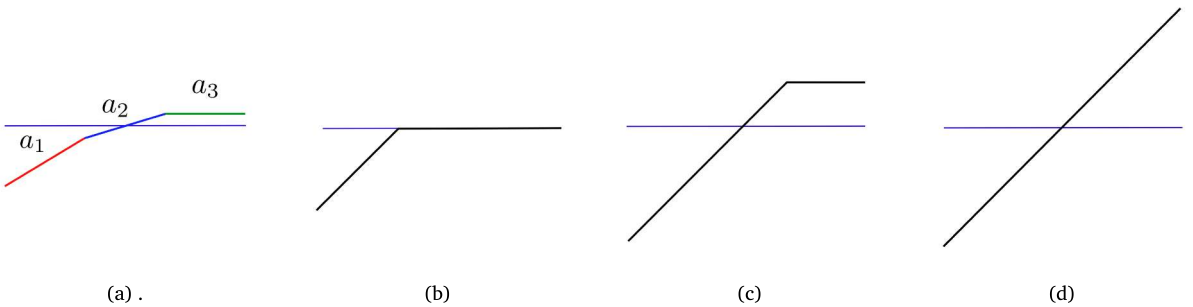
Secondly, the simulation using function (28) is performed. The level set function converges to a signed distance function for any initial condition with a non-zero gradient norm, as shown in Fig. 1c. However, for regions where the gradient norm is close to zero, the method does not converge to the expected signed distance function but remains unchanged. As a result, regions with a gradient norm very close to zero tend to retain their value.

Note that in these two methods, the diffusion function is changed to avoid an infinite diffusion for small gradient norms, however, this means we are no longer minimizing the distance between the gradient norm and 1 for these regions and hence the signed distance function is not recovered.

This is why in this article, we propose another alternative. Instead of changing the non-linear diffusion problem, we are designing a predictor problem such that we obtain a good initial guess for ϕ^0 that removes all steep or flat areas from the function to be redistanced. We then use the original unmodified method (30)-(31) to minimize the distance between $|\nabla \phi|$ and 1. The predictor-corrector scheme, proposed in this article, is able to cope with flat areas and the signed distance can be recovered for the given test case as shown in Fig. 1d. We will now proceed to introducing our predictor-corrector scheme.

4. Prediction-correction method

In Section 4.1, the prediction-correction method is explained and detailed. The weak imposition of Dirichlet boundary conditions using the Nitsche's method is discussed, and a simple 1D example of the prediction problem is provided. In Section 4.2, the finite element discretization is explained, and the prediction-correction scheme is discretized, with specific details provided for the interface integration.



(a) . (b) (c) (d)

Fig. 1. Level set after 10 fixed point iterations, the horizontal line shows the problem domain: (a) Initial value ϕ^0: $a_1 = 0.6$, $a_2 = 0.3$ and $a_3 = 0$; (b) Solution $\phi^{(10)}$ with (26) ; (c) Solution $\phi^{(10)}$ with (29) ; (d) Solution $\phi^{(10)}$ with our method.

4.1. Predictor

Let us consider an initial level set function ϕ^0 associated with an interface of arbitrary shape, decomposing the domain D into positive and negative parts such that:

$$\Omega_+ := \{x \in D : \phi^0(x) > 0\}, \tag{41}$$

$$\Omega_- := \{x \in D : \phi^0(x) < 0\}, \tag{42}$$

$$\Gamma := \{x \in D : \phi^0(x) = 0\}. \tag{43}$$

We then formulate the following predictor problem: Find $\phi_p : D \to \mathbb{R}$, such that:

$$\begin{cases} -\Delta\phi_p = \operatorname{sgn}(x) & \text{in } D, \\ \phi_p = 0 & \text{on } \Gamma, \\ \nabla\phi_p \cdot \mathbf{n} = \operatorname{sgn}(x) & \text{on } \partial D. \end{cases} \tag{44}$$

Here, we have carefully chosen sgn for the right hand side of a linear diffusion problem and as a Neumann boundary condition on ∂D. The choice of sgn in the linear form of the predictor problem ensures the maintenance of the sign of the original function. Moreover, close to the interface, there is a conservation of $|\nabla\phi_p|$. This also means that there is no strong kink in the solution at Γ as there would be if we were choosing the RHS as 1. Why this is a good first approximation for a signed distance will become even clearer in the 1D example we give below.

For the choice of the Neumann boundary condition, let us revisit the Neumann boundary condition of the problem we wish to solve (17), i.e.

$$\nabla\phi_p \cdot \mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \cdot \mathbf{n}. \tag{45}$$

With $\mathbf{n}_\phi$ the unit normal to the contour lines of function ϕ, the following holds:

$$\frac{\nabla\phi}{|\nabla\phi|} \cdot \mathbf{n} = \mathbf{n}_\phi \cdot \mathbf{n}. \tag{46}$$

Here, $\mathbf{n}_\phi$ points from Ω_- towards Ω_+ and

$$0 \le \mathbf{n}_\phi \cdot \mathbf{n} \le 1 \text{ for } x \in \Omega_+, \tag{47}$$

$$-1 \le \mathbf{n}_\phi \cdot \mathbf{n} \le 0 \text{ for } x \in \Omega_-. \tag{48}$$

We assume the "steepest" case

$$\frac{\nabla\phi}{|\nabla\phi|} \cdot \mathbf{n} \approx \operatorname{sgn}(x), \tag{49}$$

which yields

$$\nabla\phi_p \cdot \mathbf{n} \approx \operatorname{sgn}(x). \tag{50}$$

In addition to giving us a first approximation for the Neumann condition of the problem we want to solve, we also ensure that ϕ_p is not flattened near the boundary of D (see the following 1D example for more details). The purpose of the predictor ϕ_p is not to recover the exact signed distance function, but to provide a robust initial guess for the corrector step. By enforcing $\nabla\phi_p \cdot \mathbf{n} = \operatorname{sgn}(x)$, we explicitly prevent the gradient from vanishing at the boundary. While this choice may introduce a modeling error with respect to the exact distance function, it guarantees that $|\nabla\phi_p| \approx \mathcal{O}(1)$ near ∂D. This ensures that the subsequent corrector step has a valid descent direction and can effectively recover the signed distance property.

4.1.1. Illustration of prediction problem in 1D

In one dimension the prediction problem becomes: Find a continuous and differentiable function $\phi_p : [-A, A] \to \mathbb{R}$ such that:

$$-\frac{\partial^2 \phi_p(x)}{\partial x^2} = \operatorname{sgn}(x) \ \forall x \in [-A, A], \tag{51}$$

$$\phi_p(x) = 0 \text{ for } x \in \{-T, T\}, \tag{52}$$

$$\frac{\partial \phi_p(x)}{\partial x} = -1 \text{ at } x = -A, \tag{53}$$

$$\frac{\partial \phi_p(x)}{\partial x} = 1 \text{ at } x = A, \tag{54}$$

with $0 < T < A$, and sgn defined as the following step function:

$$\operatorname{sgn}(x) := \begin{cases} 1 & \text{if } x \in [-A, -T) \cup (T, A], \\ -1 & \text{if } x \in (-T, T), \\ 0 & \text{if } x \in \{-T, T\}. \end{cases} \tag{55}$$

By integrating twice per interval, $[-A, -T], [-T, T], [T, A]$, and using the boundary conditions (52)–(54), we obtain:

$$\phi_p(x) = \begin{cases} -\frac{1}{2}x^2 - (A+1)x + \frac{1}{2}T^2 - (A+1)T & \text{if } x \in [-A, -T], \\ \frac{1}{2}x^2 - \frac{1}{2}T^2 & \text{if } x \in [-T, T], \\ -\frac{1}{2}x^2 + (A+1)x + \frac{1}{2}T^2 - (A+1)T & \text{if } x \in [T, A]. \end{cases} \tag{56}$$

Fig. 2 shows a plot of this 1D solution (56). The solution has the same sign as the initial function and non-zero gradient norm except in the equidistant point $x = 0$, which is the point that is discontinuous in the gradient in the signed distance function.
Note that,

$$\frac{\partial \phi_p}{\partial x} = -x - (A+1) \quad \text{for } x \in [-A, -T], \tag{57}$$

$$\frac{\partial \phi_p}{\partial x} = x \quad \text{for } x \in [-T, T], \tag{58}$$

$$\frac{\partial \phi_p}{\partial x} = -x + (A+1) \quad \text{for } x \in [T, A]. \tag{59}$$

Thus

$$\frac{\nabla \phi_p}{|\nabla \phi_p|} = \begin{cases} -1, & x \in [-A, -T) \cup (T, A], \\ 1, & x \in (-T, T) \setminus \{0\}, \end{cases} \tag{60}$$

with $\frac{\nabla \phi_p}{|\nabla \phi_p|}$ undefined at $x = 0$, exactly like the signed distance function. In the corrector problem taking (38), we have

$$\int_{D_h} \nabla \phi_h \nabla v_h \, dx = \int_{D_h} \frac{\nabla \phi_p}{|\nabla \phi_p|} \nabla v_h \, dx \quad \forall v_h. \tag{61}$$

With $\nabla v_h = 1$ for piecewise linear shape functions in 1D

$$\int_{D_h} \frac{\partial \phi_h}{\partial x} \, dx = \int_{D_h} \frac{\nabla \phi_p}{|\nabla \phi_p|} \, dx, \tag{62}$$

which after integrating becomes:

$$\phi(x) = \frac{\nabla \phi_p}{|\nabla \phi_p|} x - T \text{ as } \phi(-T) = 0. \tag{63}$$

This is the signed distance function:

$$\phi(x) = \begin{cases} -x - T, & x \in [-A, 0), \\ x - T, & x \in (0, A], \end{cases} \tag{64}$$

and hence the exact solution see Fig. 2. This shows that by solving the predictor problem followed by one fixed point iteration, we obtain the exact solution for our simple test case. In practice, more iterations are needed, as the boundary of ∂D is usually not aligned with $\mathbf{n}_\phi$ and we introduce numerical errors for the points with $|\nabla \phi| \approx 0$. Nevertheless it motivates our choice for this predictor problem to be a very good first initial guess for the Picard iteration scheme (38).

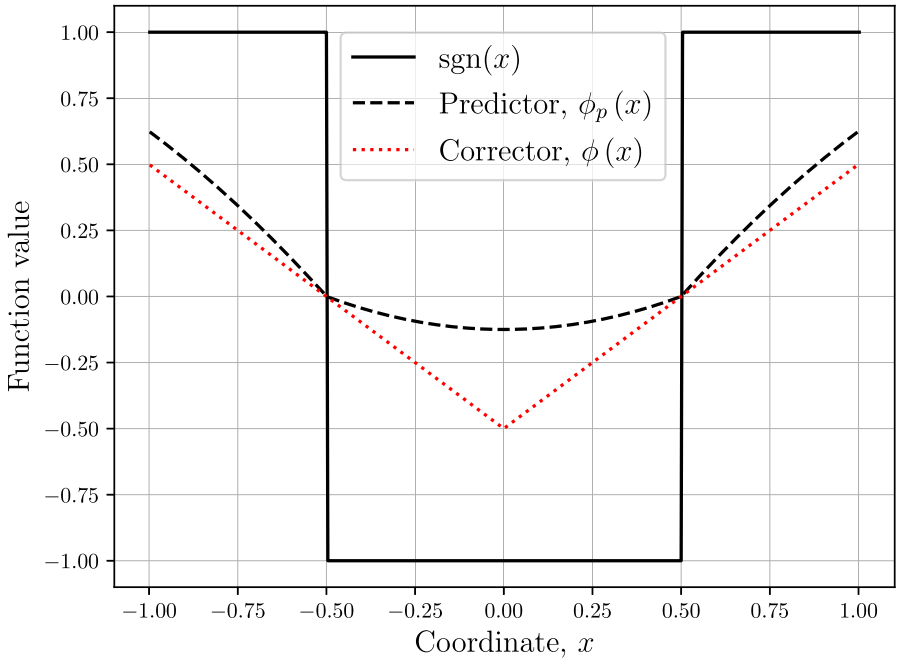


Fig. 2. Analytical solution of the 1D predictor problem (Eq. (56)) with parameters $T = 0.5$ and $A = 1$. The solution ϕ_p maintains the sign of the distance function and has a non-zero gradient everywhere except at the singularity $x = 0$, effectively regularizing the initial guess for the corrector step.

4.2. Brief discussion on the gradient norm close to zero

As mentioned above, the minimization-based (re)-distancing method suffers from one major drawback, which is the division by the norm of the gradient. This norm of the gradient will inevitably become close to zero for a signed distance function at points that are equidistant to multiple points on the boundary. For example, for a circular boundary this is the point in the center of the circle. For other geometries these might be points along equidistant lines or surfaces. The signed distance function is not differentiable at these points and piecewise polynomial approximations around these points can yield a gradient norm close to zero. Our predictor problem also features these "regularized" gradient norms close to zero around equidistant points. We show below that setting a minimal value for the gradient norm in these cases works well in the iteration scheme. Concretely, we use the expression $\max(|\nabla\phi|, \epsilon)$ in the denominator, where $\epsilon > 0$ serves purely as a numerical stability constant. The threshold ϵ is chosen to be of the order of the machine precision or slightly larger (e.g., 10^{-14} to 10^{-12}) to prevent arithmetic singularities (NaN or Inf) arising from division by zero or near-zero values. This choice of ϵ is entirely independent of physical or geometric parameters like the mesh size or the local curvature, serving only as a robust guard against numerical instability. The problem is less prevalent for narrow band approaches in which redistancing only takes place in a narrow band around the boundary. However, for any application with high curvature or in which interfaces merge, equidistant points will occur even in the narrow band. This needs to be considered in any redistancing scheme.

In the next Section, we detail the finite element discretization of the proposed prediction-correction scheme.

4.3. Finite element discretization

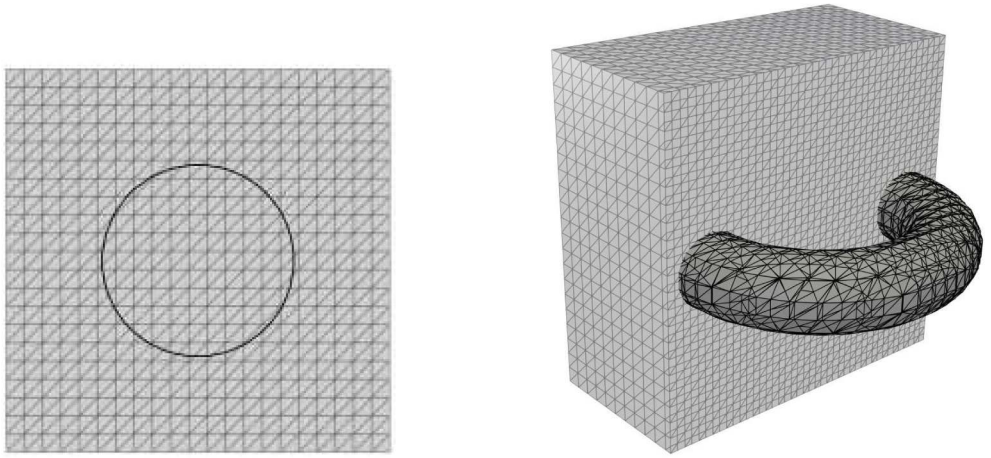
In this Section, we will first introduce the finite element discretization for a mesh that conforms to the interface Γ. We will extend this formulation to the case in which the mesh is not aligned with Γ.

4.3.1. Interface-fitted mesh

Let D_h denote the discretized domain D into a mesh conforming to Γ_h. Using the space V_h^0 previously defined by (37), the finite element problem to solve the predictor-corrector scheme reads as follows.

Predictor. Find $\phi_{p,h} \in V_h^0$ such that for all $v_h \in V_h^0$

$$\int_{D_h} \nabla\phi_{p,h} \cdot \nabla v_h \, dx = \int_{D_h} \operatorname{sgn}(x) v_h \, dx + \int_{\partial D_h} \operatorname{sgn}(x) v_h \, ds. \tag{65}$$



(a) . (b)

Fig. 3. Unfitted mesh configuration: (a) 2D interface Γ crossing background mesh elements; (b) 3D cut elements for the torus level set function.

Corrector. With $\phi_h^0 := \phi_{p,h}$ the initial value of the iterative scheme, the finite element formulation to solve the corrector problem becomes: For $n = 0, \dots, N-1$, find $\phi_h^{n+1} \in V_h^0$ such that for all $v_h \in V_h^0$

$$\int_{D_h} \nabla \phi_h^{n+1} \nabla v_h \, dx = \int_{D_h} \frac{\nabla \phi_h^n}{\left|\nabla \phi_h^n\right|} \nabla v_h \, dx. \tag{66}$$

We now extend this formulation to a finite element formulation for meshes that do not conform to the interface Γ. This choice is motivated by the fact that re-distancing is often applied in the context of moving boundary problems in which re-meshing can become challenging and costly.

4.3.2. Unfitted finite element formulation

The principle of an unfitted mesh approach, where the interface Γ_h arbitrarily cuts through the elements of the background mesh $\mathcal{K}_h$, is illustrated for both 2D and 3D geometries in Fig. 3. Let us now presume that the faces of the triangular or tetrahedral mesh $\mathcal{K}_h$ don't coincide with Γ_h. A new finite element space is constructed, which does not incorporate the boundary conditions on the interface. Within the continuous Galerkin framework, this space is defined as:

$$V_h := \left\{ v \in H^1(D) \cap C^0(D) \mid v_{|K} \in \left[\mathbb{P}_1(K) \text{ for all } K \in \mathcal{K}_h \right] \right\}, \tag{67}$$

where K denotes the elements of the mesh. Now that the Dirichlet boundary conditions are no longer included in the finite element space, we enforce them weakly through additional terms in our finite element formulation.

Weak enforcement of boundary conditions. Three main methods exist to enforce boundary conditions weakly: i) the penalty method, ii) Lagrange multiplier, iii) Nitsche's method. In [23], the penalty method is employed. While it is relatively straightforward to formulate, it requires careful selection of a penalty parameter suitable for the numerical problem. If the parameter is chosen too low, the solution to the problem is not unique. Although the algorithm converges well to a reinitialized level set, insufficient penalization on the interface can lead to significant interface displacement. Conversely, choosing a penalty parameter that is too high results in excessive constraints, generating matrix ill-conditioning and subsequent boundary oscillations. This phenomenon is known as locking.

In [24], the authors utilize the Lagrange multiplier method to impose Dirichlet conditions on Γ_h. This method introduces a new unknown parameter, λ. At each iteration, the Lagrange multiplier λ is updated by solving a PDE for the constraint. The choice of the Lagrange multiplier space is not always straightforward as it has to fulfill inf-sup stability conditions. In this work, Nitsche's method [34] is used. Similar to the penalty method, it requires the selection of a penalty parameter. The choice of this parameter leads to difficulties akin to those previously mentioned for the penalization method. However, unlike the penalty method, Nitsche's method offers a stable and effective numerical formulation for moderate penalty parameters. The Nitsche parameter γ_D is independent of the mesh size. The optimal value is not strictly defined; in the literature, it is usually chosen empirically between 10 and 100. The value must be sufficiently large to guarantee coercivity and convergence, but not so large as to impair the conditioning of the system.

To integrate over Γ_h, which now intersects elements in an arbitrary fashion, we use the cut integration scheme introduced in the following Section 4.3.2.

Sub-integration for interface integral. First, we reconstruct the interface in order to integrate over Γ crossing element K. A simple linear interpolation of level set values is used in this work to calculate the points of intersection between the edges of the element K and the interface and a piecewise linear interface, Γ_K, is reconstructed by connecting these points on the edges (see Fig. 4). A

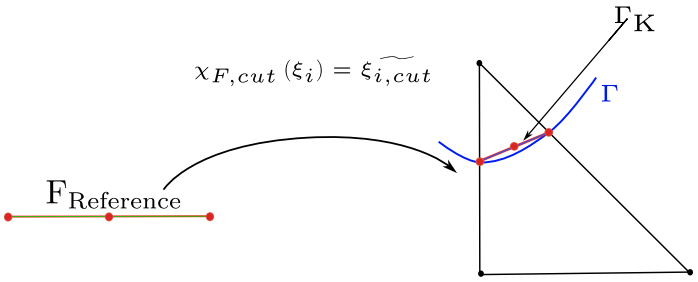


Fig. 4. Mixed-dimensional sub-integration process: the interface is reconstructed via linear interpolation and the quadrature rule is generated by mapping a reference quadrature rule on to the linear interface reconstruction.

visualization of the resulting cut mesh elements in 3D, specifically for the torus level set function, is provided in Fig. 3b. Then, to integrate over the reconstructed interface part Γ_K, we generate a quadrature rule by mapping standard integration rules onto the piecewise linear approximation of Γ_K inside the reference element. This generates a new quadrature rule with new quadrature points $\tilde{p}_i$ and new weights $\tilde{w}_i$ for each interface part. The new weights are computed by multiplying the standard weights by the Jacobian of the mapping from physical space to the reference space. The mapping to generate the quadrature rule is mixed dimensional. An example of the approximation of the Γ_K integral as part of K is given in Fig. 4.

We can now formulate our predictor-corrector scheme for interfaces Γ_h that freely intersect through a background mesh.

Predictor. Using Nitsche's method, the discretized weak formulation of the predictor problem reads: Find $\phi_{p,h} \in V_h$, such that for all $v_h \in V_h$

$$a(\phi_{p,h}, v_h) = l(v_h), \tag{68}$$

where

$$\begin{aligned} a(\phi_{p,h}, v_h) = &\int_{D_h} \nabla\phi_{p,h} \cdot \nabla v_h \; dx - \int_{\Gamma_h} \nabla\phi_{p,h} \cdot \mathbf{n}_\phi \, v_h \; ds - \int_{\Gamma_h} \nabla v_h \cdot \mathbf{n}_\phi \, \phi_{p,h} \; ds, \\ &+ \gamma_D \int_{\Gamma_h} h^{-1} \phi_{p,h} \, v_h \; ds \end{aligned} \tag{69}$$

$$l(v_h) = \int_{D_h} \operatorname{sgn}(x) \, v_h \; dx + \int_{\partial D_h} \operatorname{sgn}(x) \, v_h \; ds. \tag{70}$$

The integration term over the discretization of Γ, denoted Γ_h, is done by the cut integration scheme introduced in the previous Section 4.3.2. The rest of the terms are integrated with a standard integration scheme. A cut integration on the volume is not required as the solution to our elliptic problem is continuous at Γ despite the discontinuous RHS term $\operatorname{sgn}(x)$.

Corrector. Using Nitsche's method, and $\phi_h^0 := \phi_{p,h}$ as the initial value of the iterative scheme, the finite element formulation to solve the corrector problem reads: For $n = 0, \dots, N-1$, find $\phi_h^{n+1} \in V_h$ such that for all $v_h \in V_h$

$$a(\phi_h^{n+1}, v_h) = l(v_h, \phi_h^n), \tag{71}$$

where

$$\begin{aligned} a(\phi_h^{n+1}, v_h) = &\int_{D_h} \nabla\phi_h^{n+1} \cdot \nabla v_h \; dx - \int_{\Gamma_h} \nabla\phi_h^{n+1} \cdot \mathbf{n}_\phi v_h \; ds - \int_{\Gamma_h} \nabla v_h \cdot \mathbf{n}_\phi \phi_h^{n+1} \; ds \\ &+ \gamma_D \int_{\Gamma_h} h^{-1} \phi_h^{n+1} v_h \; ds \end{aligned} \tag{72}$$

$$l(v_h, \phi_h^n) = \int_{D_h} \frac{\nabla\phi_h^n}{\max\left(\left|\nabla\phi_h^n\right|, \epsilon\right)} \cdot \nabla v_h \; dx - \int_{\Gamma_h} \frac{\nabla\phi_h^n}{\max\left(\left|\nabla\phi_h^n\right|, \epsilon\right)} \cdot \mathbf{n}_\phi v_h \; ds \tag{73}$$

with ϵ being a very small positive scalar.

5. Numerical results

In this Section, we evaluate our prediction-correction method for four functions with increasing numerical complexity: (i) a smooth circular interface function, which enables a thorough assessment of the method (including convergence order, influence of the predictor problem, and analysis of the Nitsche parameter); (ii) a piecewise constant function; (iii) a complex example involving a sharply varying interface; and (iv) a three-dimensional case, demonstrating the ease with which our method can be extended to 3D

problems. To analyze the convergence towards the exact solution, we denote the difference between the numerical approximation, ϕ_h, and the exact solution, ϕ_{ex} , as

$$e(\phi_h(x)) = \phi_h(x) - \phi_{ex}(x), \quad x \in D_h. \tag{74}$$

We evaluate this error in the L^2 norm as follows:

$$\|e\|^2_{L^2(D_h)} = \frac{\int_{D_h} e^2(\phi_h(x))\, dx}{\int_{D_h} dx}. \tag{75}$$

Similarly, the H^1 semi-norm of the error is defined as:

$$|e|^2_{H^1(D_h)} = \frac{\int_{D_h} |\nabla e(\phi_h(x))|^2\, dx}{\int_{D_h} dx}. \tag{76}$$

Furthermore, to evaluate the method's ability to converge towards the solution of the eikonal Eq. (5), we introduce the eikonal error, defined as:

$$e_{\text{eik}}(\phi_h(x)) = 1 - \left|\nabla \phi_h(x)\right|, \quad x \in D_h. \tag{77}$$

We compute this error using the L^2 norm as follows:

$$\left\|e_{\text{eik}}\right\|^2_{L^2(D_h)} = \frac{\int_{D_h} e^2_{\text{eik}}(\phi_h(x))\, dx}{\int_{D_h} dx}. \tag{78}$$

Finally to assess the method's precision in preserving the initial interface Γ^0_h we define the following norm with respect to the solution of the prediction problem ϕ^0_h:

$$\|\phi\|^2_{L^2(\Gamma)} = \int_{\phi^0_h=0} \left(\phi_h(x) - \phi^0_h(x)\right)^2 ds \tag{79}$$

$$= \int_{\phi^0_h=0} \left(\phi_h(x)\right)^2 ds. \tag{80}$$

5.1. Classical case: circular level set

This first numerical example is used to study the order of convergence of the prediction-correction method for the three norms introduced above. We start with an initial quadratic function on the domain $D = [0, \mathrm{L}]^2$, with $\mathrm{L} = 1$, defined as:

$$\phi_{\text{init}}(x, y) = \left(x - x_0\right)^2 + \left(y - y_0\right)^2 - r^2, \tag{81}$$

with $\left(x_0, y_0\right)$ being the coordinate of the center of the circle and r being the radius. We set $\left(x_0, y_0\right) = (0.5, 0.5)$ and $r = 0.25$. As shown in Fig. 5a, this function is a parabola, and not a signed distance function. The exact solution of the reinitialized level set, denoted ϕ_{ex}, is known as a cone centered at $\left(x_0, y_0\right)$ and such that the line of level zero coincides with that of ϕ_{init}. The exact solution ϕ_{ex} is given by:

$$\phi_{ex}(x, y) = \sqrt{\left(x - x_0\right)^2 + \left(y - y_0\right)^2} - r. \tag{82}$$

Convergence study. We fix the Nitsche's parameter to 10^2 and we study the errors defined above for different mesh sizes. By observing the evolution of the L^2-errors of e and e_{eik} as a function of correction iterations (see Fig. 7a and b), we note that, regardless of the mesh size, these errors fall below 0.01 and 0.05, respectively, after just 3 iterations. The level set function after one iteration of the correction problem (see Fig. 5c) and after ten iterations (see Fig. 5d) differ only very slightly. In practice, this means that only one iteration of the correction problem gives a sufficiently accurate approximation to the eikonal equation for any application in which a strict verification of the eikonal equation is not required, and an approximation within these error margins is acceptable (for example, in shape optimization with the level set method). The cross-section displayed in Fig. 6 shows that even at the center of the domain, near the singularity, the level set shows no oscillation or locking effects.

Convergence order. In this Section, we calculate the convergence order of the prediction-correction scheme for the three errors defined in the beginning of this Section. We set $\gamma_D = 10^2$ for the convergence study. The solution is considered to have converged if

$$\left| \left\| e_{\text{eik}}(\phi^{n+1}) \right\|_{L^2(D)} - \left\| e_{\text{eik}}(\phi^{n}) \right\|_{L^2(D)} \right| < 10^{-8}, \tag{83}$$

as defined in (39). We study convergence with mesh refinement for the series of triangular meshes with size $h = \mathrm{L}/40, \mathrm{L}/60, \mathrm{L}/80, \mathrm{L}/120, \mathrm{L}/160, \mathrm{L}/240, \mathrm{L}/360$.

The order of convergence is computed between the errors obtained with $h = \mathrm{L}/40$ and $h = \mathrm{L}/360$. Numerically, we observe in Fig. 8a that the order of accuracy of our scheme at the interface is 1.99, which is very close to the optimal order of 2. In Fig. 8b, d, and c, we observe that the convergence order of the eikonal error is 0.82, the convergence order of the error e over the whole domain

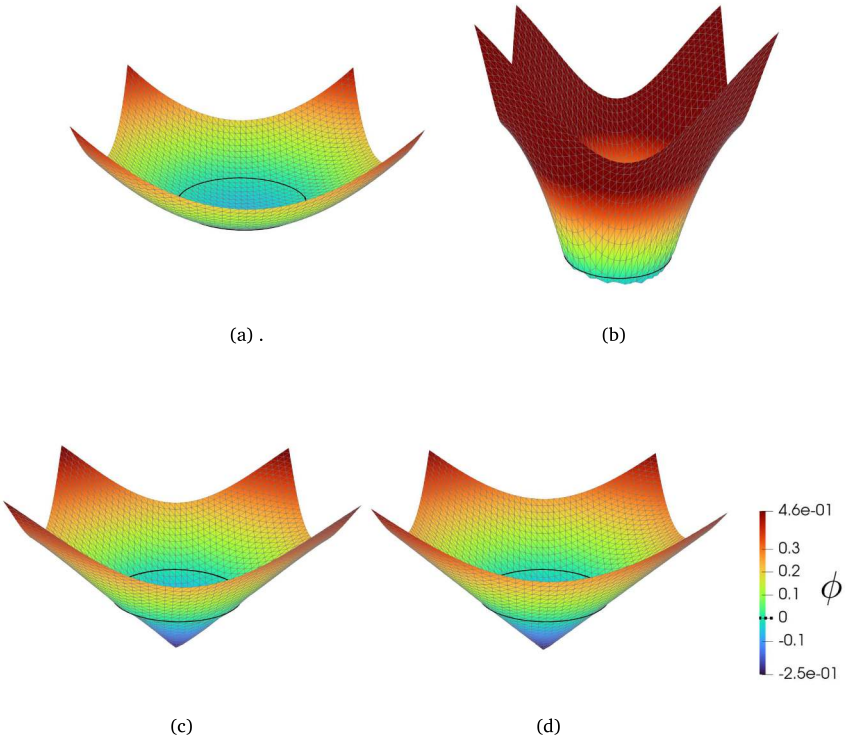


Fig. 5. Circular level set for reinitialization with prediction-correction method with $h = \mathrm{L}/40$ and $\gamma_D = 10^2$: (a) level set warped before reinitialization; (b) level set warped after prediction problem; (c) predicted level set warped after 1 iteration of correction; (d) level set warped after 10 iterations of correction.

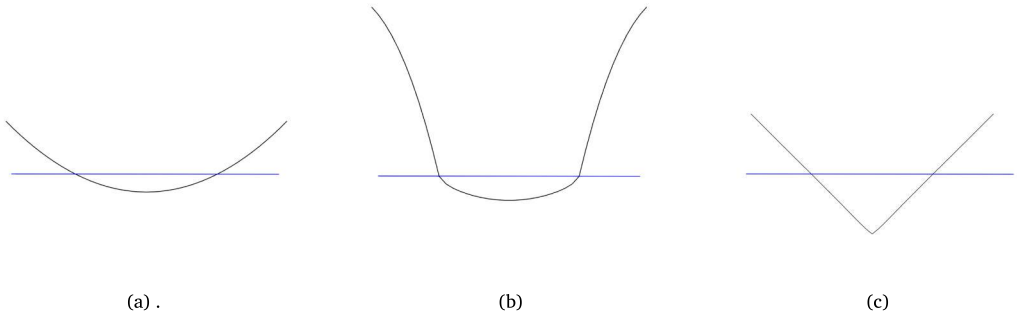
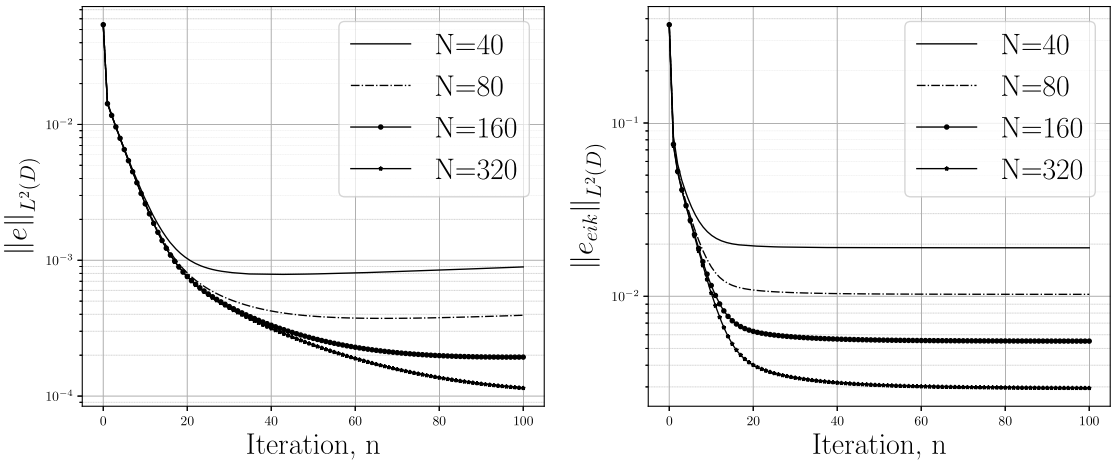


Fig. 6. View of level set function along center line: (a) initial function; (b) solution of predictor step; (c) solution of correction step (after 10 iterations).

is 1.1, and the convergence order of the H^1 error is 0.8. These rates remain below the theoretical optimal values, which are of order 2 for the L^2 error and of order 1 for both the eikonal and the H^1 errors. We have two main sources of error: (i) equidistant points with norm of the gradient close to zero (theoretically not differentiable), (ii) the splitting error in the corrector scheme from the Picard fixed point linearization. To test if the point in the center of the circle is causing the reduction in the convergence order, we solve the circular level set problem only in a narrow band around the interface. The errors are denoted as "P.C. annulus" in Fig. 8b, b and c. We observe that the optimal convergence orders are recovered for all error measures.

Sensitivity analysis of Nitsche parameter γ_D. We analyze the impact of the Nitsche penalty parameter γ_D on the method's accuracy and conservation properties. Fig. 9 illustrates the evolution of the eikonal error and mass ratio for $\gamma_D \in \{10, 100, 500\}$. As shown in Fig. 9a, the final eikonal error at convergence is independent of γ_D, confirming the robustness of the method's accuracy with respect to the penalty parameter. However, the convergence rate is affected: larger penalty parameters result in a slower convergence, requiring a higher number of iterations to reach the stopping criterion. Regarding mass conservation (Fig. 9b), we observe that the mass ratio remains close to 1, demonstrating the effectiveness of Nitsche's method in preserving volume. Interestingly, for smaller γ_D (e.g., 10), the predictor step tends to decrease the mass ratio while the corrector step increases it, leading to a compensatory effect. For larger γ_D



(a) . (b)

Fig. 7. Error evolution of circle example (prediction error is not plotted) with $\gamma_D = 10^2$ for different mesh sizes: (a) Error $\|e\|_{L^2(D)}$; (b) Error $\|e_{\text{eik}}\|_{L^2(D)}$.

(e.g., 500), the mass ratio remains consistently closer to 1 throughout the iterations, indicating stronger enforcement of the boundary condition and better volume preservation.

Comparative performance analysis. A comprehensive performance benchmark was conducted against the Elliptic method and the Hamilton-Jacobi (HJ) solver (Tables 1–3). The FD-HJ results were obtained using a typical Godunov-type finite difference scheme for the reinitialization equation, known for its stability and computational efficiency in local propagation. We emphasize that this FD-HJ approach is inherently restricted to structured (Cartesian) meshes, whereas our FEM/CutFEM formulation can handle arbitrary, unstructured meshes. We acknowledge that the total runtime of our FEM method is significantly higher than the fast Finite Difference (FD) HJ solver. This overhead stems primarily from the need to solve large, global sparse linear systems, an approach inherently more costly than the local updates of the FD-HJ method. Nevertheless, the scheme exhibits excellent scalability with an empirical complexity order of $O(\text{DoF}^{1.16})$, close to the theoretical linear complexity of optimal iterative solvers.

This comparative study is performed on a simple circular interface. For such smooth interfaces, the final errors ($\|e_{\text{eik}}\|_{L^2(D)}$, $\|e\|_{L^2(D)}$, $\|e\|_{L^2(\Gamma)}$) at convergence remain consistent regardless of whether the predictor problem is solved (Table 10b), confirming that the P.C. scheme does not degrade solutions provided by conventional minimization approaches. However, the P.C. scheme requires approximately 1.25 times more iterations than the minimization-based method (Tables 1 and 2) to reach the same convergence criterion. This increased computational cost is fundamentally justified by the scheme's unique advantages, which are particularly relevant for complex geometries.

The three methods exhibit distinct error profiles (Table 10b). While the HJ method achieves lower domain error $\|e\|_{L^2(D)} = 1.2 \times 10^{-3}$ through efficient local propagation, the P.C. method prioritizes interface accuracy, achieving $\|\phi\|_{L^2(\Gamma)} = 1.1 \times 10^{-4}$. This is nearly 60 times better than the Elliptic method and 4 times better than HJ. This trade-off is deliberate: for level set applications, interface fidelity is the critical requirement, whereas slight variations in the distance function away from the interface have negligible impact on front-tracking accuracy.

The justification for this trade-off lies primarily in two areas. First, concerning global convergence speed, the P.C. method updates the entire level set function simultaneously through a global PDE solve. As illustrated in Fig. 10a, high eikonal accuracy is attained after one predictor step followed by a few corrector iterations, where the error is propagated globally across the entire domain. Conversely, the HJ method requires a large number of iterations to reach comparable accuracy because distance information must locally propagate outward from the interface. Second, regarding geometric fidelity and robustness, the P.C. scheme ensures superior interface fidelity. As demonstrated by the comparative error data, the P.C. method achieves the lowest interface error $\|\phi\|_{L^2(\Gamma)}$ across all three methods. This is guaranteed by the use of Nitsche's method within CutFEM, leading to minimal interface displacement (of order 10^{-4}) and robust mass conservation. This high level of geometric control is essential when dealing with complex, non-smooth interfaces where traditional methods struggle.

Moreover, the least squares minimization method without predictor, the core limitation lies in its failure to converge when the initial level set function contains large, flat areas where the gradient norm is near zero. In these critical cases, the P.C. scheme's predictor problem is essential for stabilization, as demonstrated in Section 5.2 where the least squares minimization method without predictor fails to converge on a flat initial function. A more complex study case further demonstrates the benefit of the P.C. scheme in Section 5.3. Despite the higher computational runtime on simple test cases, the P.C. method's FEM-based formulation, which naturally accommodates unstructured meshes, is suited for parallel computing architectures, enabling efficient distributed memory parallelization (e.g., via MPI) and strong scalability on HPC systems, as demonstrated in Section 5.4. This makes the P.C. method a

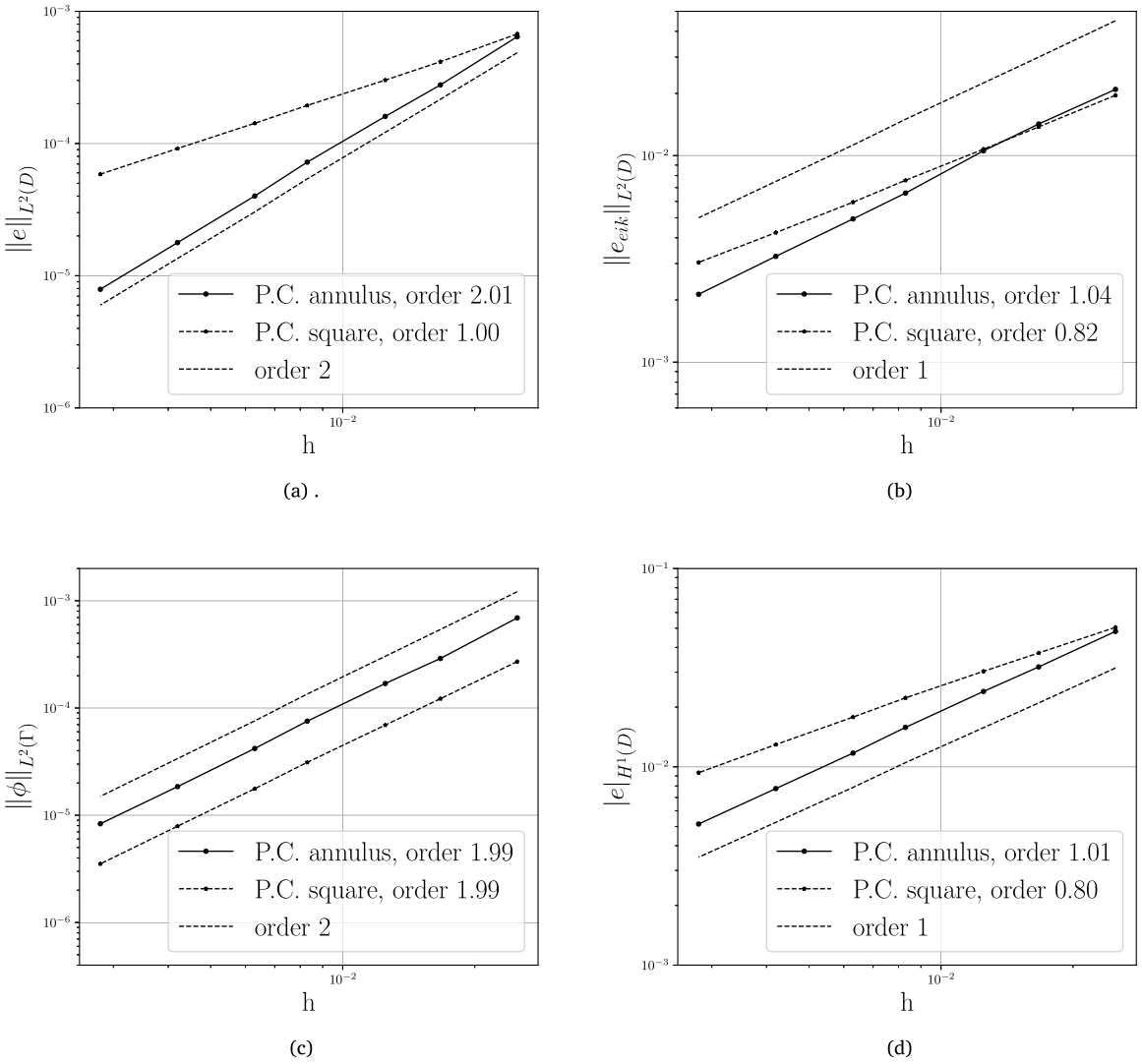


Fig. 8. The convergence order of the P.C. method for $\gamma_D = 100$ with respect to the following errors: (a) Error $\|e\|_{L^2(D)}$; (b) Error $\|e_{\text{eik}}\|_{L^2(D)}$; (c) Error $\|\phi\|_{L^2(\Gamma)}$, (d) Error $\|e\|_{H^1(D)}$.

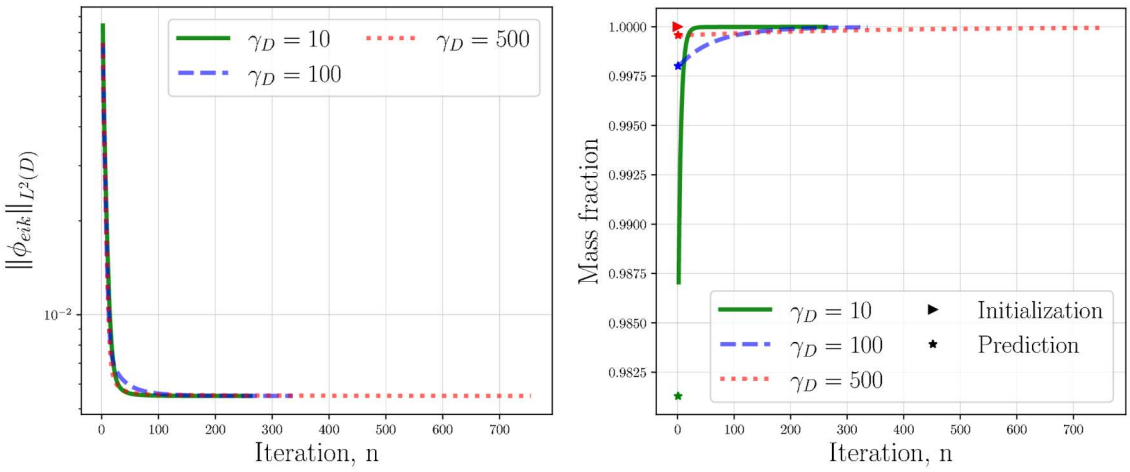
Table 1
Computational benchmark of the proposed predictor-corrector reinitialization scheme.

Mesh Size	Total DoF ($\times 10^3$)	Time (s)	Memory (MB)	Iterations
80×80	6.4	2.93	156.4	62
160×160	25.6	10.11	231.1	77
320×320	102.4	51.91	538.3	85
160×160 (1 Iter.)	25.6	0.29	–	1

valuable alternative for applications requiring signed distance functions to complex geometries, particularly when HPC resources are available.

5.2. Flat function

A second example is studied to understand the capability of the scheme to correct a flat function, i.e., a large area with $|\nabla\phi| \approx 0$. These types of flat functions are a challenge for the minimization based reinitialization scheme which includes the term $\frac{1}{|\nabla\phi|}$.



(a) . (b)

Fig. 9. Sensitivity analysis of the Nitsche parameter γ_D: (a) Evolution of the eikonal error $\|e_{\text{eik}}\|_{L^2(D)}$; (b) Evolution of the mass ratio.

Table 2
Computational benchmark of the least squares minimization method without predictor.

Mesh Size	Total DoF ($\times 10^3$)	Time (s)	Memory (MB)	Iterations
80×80	6.4	1.49	150.9	41
160×160	25.6	7.45	208.0	57
320×320	102.4	22.09	438.5	41

Table 3
Computational benchmark of the reference Hamilton-Jacobi reinitialization method (finite difference based on structured mesh).

Mesh Size	Total DoF ($\times 10^3$)	Time (s)	Memory (MB)	Iterations
80×80	6.4	0.35	282.2	2000
160×160	25.6	0.08	291.4	138
320×320	102.4	0.69	332.1	264

Let the initial level set function be a step function (see Fig. 11) on the domain $D = [0,1]^2$ defined by:

$$\phi_{\text{init}}(x,y) = \begin{cases} -1 & \text{if } x > 0.5, \\ 0 & \text{if } x = 0.5, \\ 1 & \text{if } x < 0.5. \end{cases} \tag{84}$$

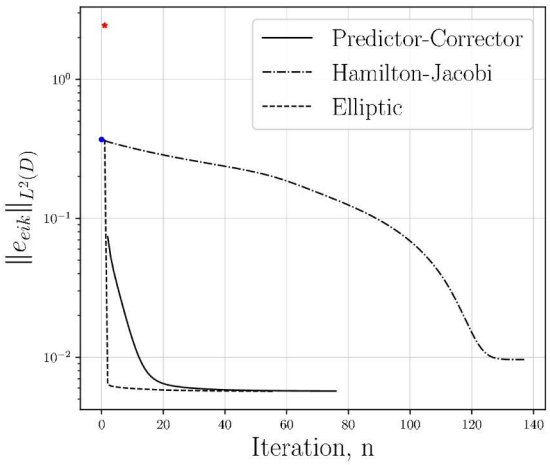
Without a predictor problem, the minimization problem's right-hand side becomes infinity for the flat regions in the Picard fixed point iteration scheme. The solution of the predictor problem is a level set with the same sign as the initial level set, while correcting the zero gradient norms so that the corrector problem is well-defined. As observed in Figs. 11 and 12 for the level set solution of the predictor problem, no element K of the mesh satisfies: $|\nabla\phi(K)| = 0$. From this predictor problem, only a few iterations are necessary to obtain a reinitialized level set function.

To complete our study, we initialized a variant of the previous level set to assess the ability of the scheme to correct very irregular functions, containing areas with zero gradient norms and areas with very high gradient norms. The initialized level set is defined as:

$$\phi_{\text{init}}(x,y) = \arctan(10(x-0.5)) + \xi(x) \tag{85}$$

with ξ a function of small Gaussian perturbation generated randomly.

This example highlights the relevance of the predictor problem for highly irregular initial level set functions. Indeed, as shown in Fig. 15a and b, the errors obtained using the prediction-correction scheme are significantly lower than those resulting from solving the minimization problem alone. Looking at the level set function at convergence displayed in Figs. 13 and 14, we observe that the minimization problem alone is unable to smooth out significant irregularities in the vicinity of the domain boundaries while the predictor corrector problem yields a smooth approximation of the signed distance. Furthermore, it is observed that the least squares minimization scheme (elliptic scheme) requires more iterations to converge to the solution. Finally, the error on e decreases



(a) .

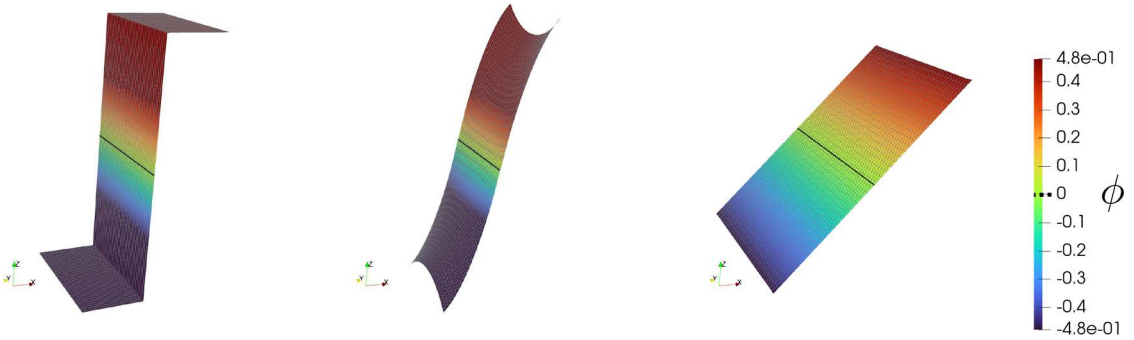
Eikonal Error ($\times 10^{-3}$)			
Metric	P.C.	H.J.	Elliptic
$\|e_{\text{eik}}\|_{L^2(D)}$	5.7	9.6	5.7

Domain Error ($\times 10^{-3}$)			
Metric	P.C.	H.J.	Elliptic
$\|e\|_{L^2(D)}$	7.2	1.2	5.1

Interface Error ($\times 10^{-4}$)			
Metric	P.C.	H.J.	Elliptic
$\|\phi\|_{L^2(\Gamma)}$	1.1	4.3	61.5

(b)

Fig. 10. Comparison between the predictor-corrector method, the least squares minimization method without predictor (Elliptic), and the Hamilton-Jacobi method for $h = L/160$: (a) convergence analysis of $\|e_{\text{eik}}\|_{L^2(D)}$; (b) final error comparison.



(a) . (b) (c)

Fig. 11. Step-shape warped for reinitialization step with P.C. method, $\gamma_D = 10$ and $h = \text{L}/41$: (a) level set function before reinitialization; (b) level set function after prediction problem; (c) level set function after P.C. reinitialization.

over iterations in the prediction-correction method, whereas this is not the case for the least squares minimization method without predictor as shown in Fig. 15b.

To illustrate the ability of our method to reinitialize level sets with a complex interface, we examine an example next that features an irregular boundary.

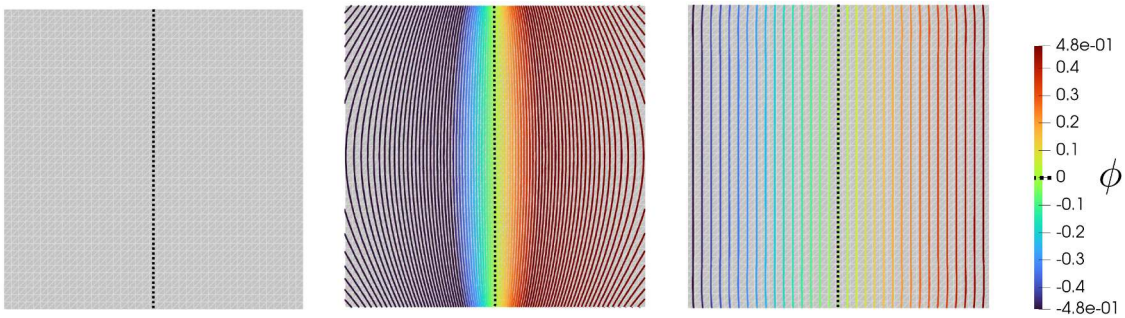
5.3. Complex interface: star level set

In this Section, we test our scheme on a star-shaped level set function (see Figs. 16a and 17a) defined on the domain $D = [0, 1]^2$ given by

$$\phi_{\text{init}}(x, y) = \frac{1}{2}\left(\sqrt{(x - x_0) + (y - y_0)} - \frac{1}{10}\cos\left(n \arctan\left(\frac{y - y_0}{x - x_0}\right) + 2\right) + \frac{1}{4}\right) \tag{86}$$

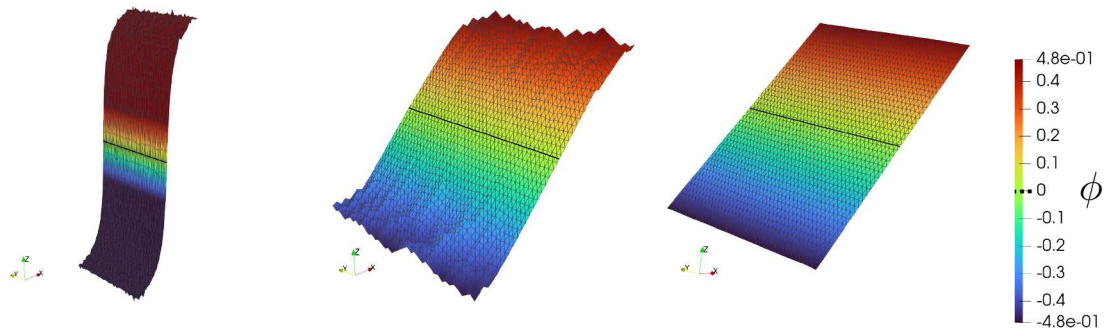
with (x_0, y_0) the center of the star and $n = 8$ the number of rays. The distance function associated with this star-shaped form is particularly complex to compute. The interface exhibits large areas with high curvature and many equidistant points along lines between the star rays for which the norm of the polynomially interpolated gradient can become zero. The preservation of the geometry as well as the large number of equidistant lines makes this a challenging test problem.

Convergence study and eikonal error. We fix the Nitsche penalty parameter to $\gamma_D = 100$ and evaluate the error metrics for various mesh sizes (h). The L^2 eikonal error $\|e_{\text{eik}}\|$ evolves rapidly with correction iterations (Fig. 18b). Regardless of the mesh size, the eikonal error falls below the threshold of 1 after only 3 iterations. This rapid convergence demonstrates the efficiency of the linear correction step in driving the level set function towards the distance property $|\nabla\phi| = 1$. The effectiveness of the reinitialization is visually confirmed in Fig. 17c, which shows well-distributed, equidistant level set contours after the corrector step.



(a) . (b) (c)

Fig. 12. Step-shape level set contours for reinitialization step with P.C. method, $\gamma_D = 10$ and $h = \mathrm{L}/41$: (a) level set contours before reinitialization; (b) level set contours after predictor problem; (c) level set contours after P.C. reinitialization.



(a) . (b) (c)

Fig. 13. Warped level set for arctan-problem with $\gamma_D = 10$ and $h = \mathrm{L}/41$: (a) level set function before reinitialization; (b) level set function after minimization based reinitialization; (c) level set function after P.C. reinitialization.

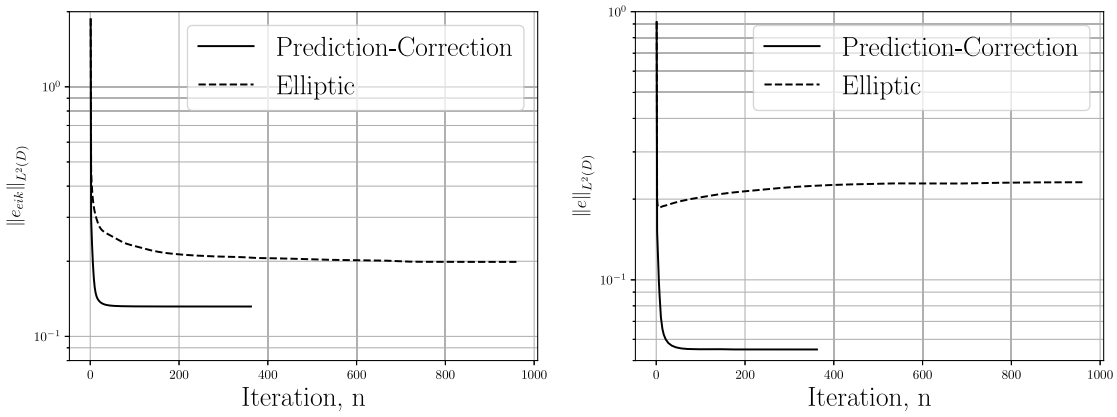


(a) . (b) (c)

Fig. 14. Perturbed arctan level set contours with $\gamma_D = 10$ and $h = \mathrm{L}/41$: (a) level set contours before reinitialization; (b) level set contours after minimization based reinitialization; (c) level set contours after P.C. reinitialization.

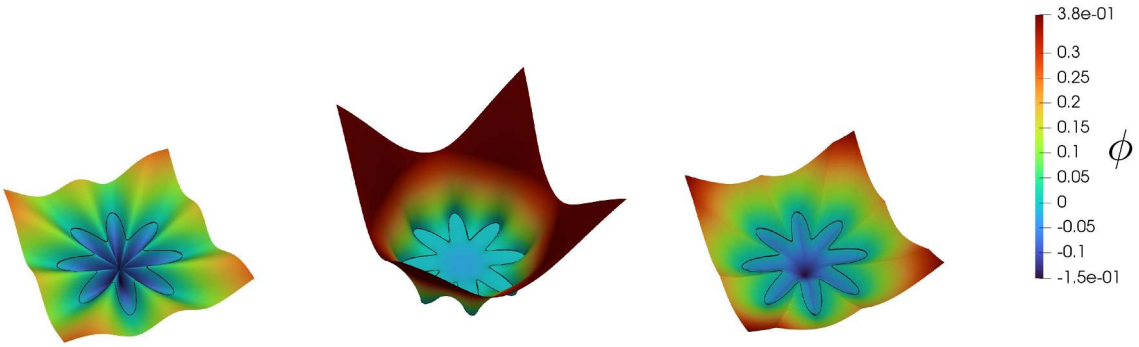
Interface analysis and mass conservation in non-smooth geometry. Having established convergence, we now examine the scheme's ability to preserve the interface geometry. The primary challenge in this test case is the highly non-smooth, sharp-angled star-shaped geometry. This complexity often leads to numerical diffusion or spurious interface displacement in reinitialization methods. We examine the evolution of the interface error and mass conservation (Fig. 18a):

- **Visual stability:** Despite the challenging geometry, Nitsche's method for weak enforcement of boundary conditions ensures that no noticeable displacement of the interface Γ is observed throughout the reinitialization process.
- **Mass displacement mechanism:** The two-step scheme manages mass conservation through compensation. The predictor step induces a slight outward interface displacement, resulting in a marginal mass increase. The corrector step then generates an equal displacement in the opposite direction, precisely compensating for this mass gain.



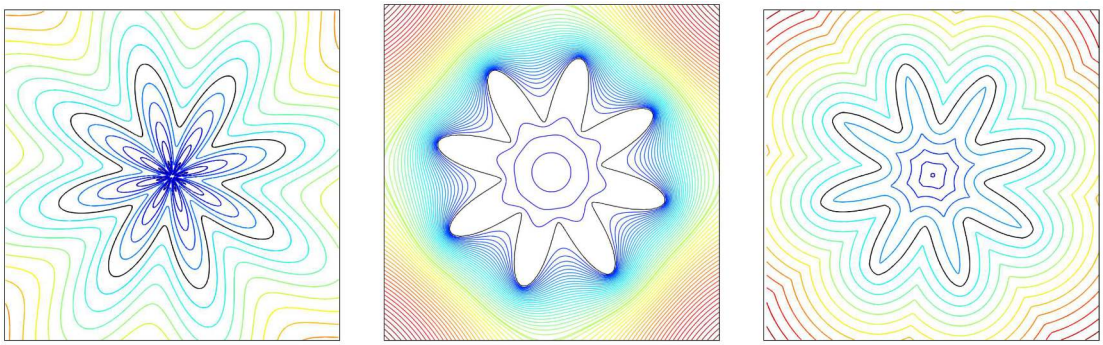
(a) . (b)

Fig. 15. Error evolution for $\gamma_D = 10$ and $h = \mathrm{L}/41$: (a) error $\|e_{\mathrm{eik}}\|_{L^2(D)}$; (b) error $\|e\|_{L^2(D)}$.



(a) . (b) (c)

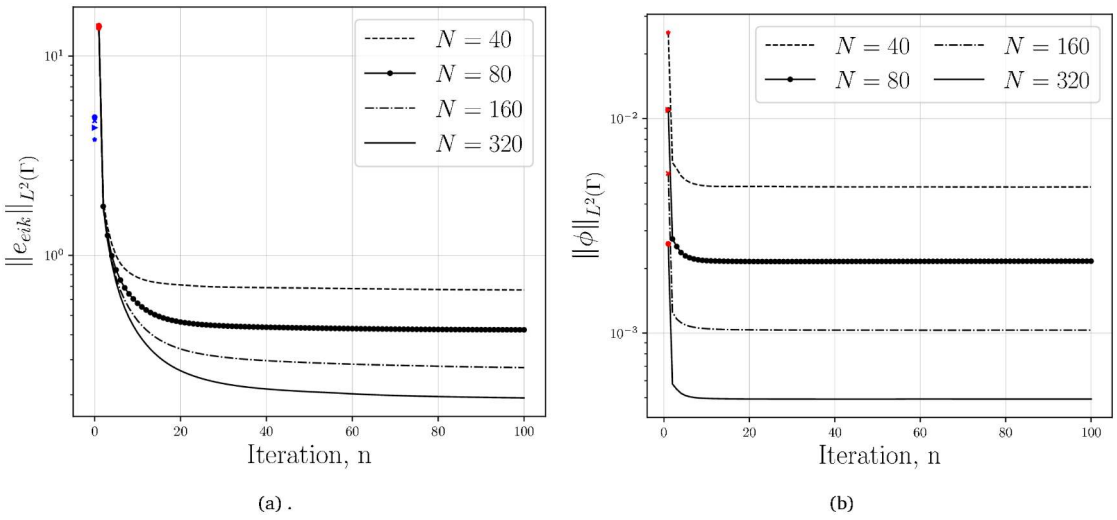
Fig. 16. Warped star level set function with $\gamma_D = 100$ and $h = \mathrm{L}/320$: (a) before reinitialization; (b) after predictor; (c) after corrector reinitialization.



(a) . (b) (c)

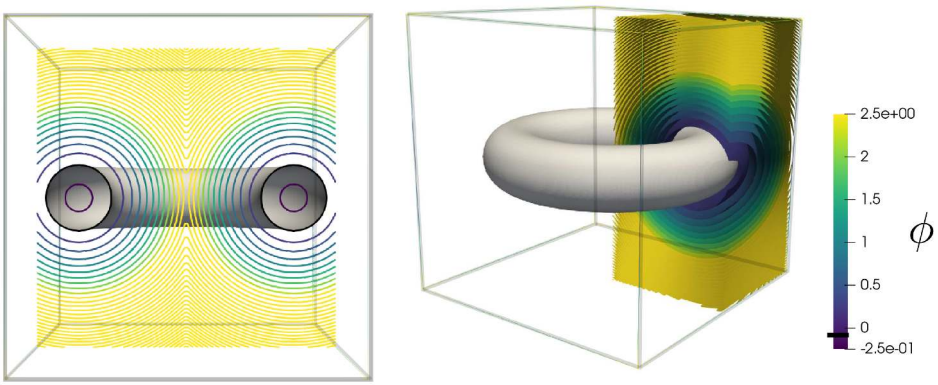
Fig. 17. Star level set contours $\gamma_D = 100$ and $h = \mathrm{L}/320$: (a) before reinitialization; (b) predictor problem; (c) after corrector reinitialization.

The overall interface displacement error remains negligible at $O(10^{-3})$ in the L^2 norm, confirming that the linear two-step scheme effectively achieves reinitialization while preserving both the position and volume of the complex, non-smooth interface. Visual inspection of Fig. 17c further confirms the geometric stability, showing no perceptible interface distortion or artificial displacement after reinitialization.



(a) . (b)

Fig. 18. Evolution of the error with respect to mesh size (h) for the star-shaped domain example ($\gamma_D = 100$): (a) L^2 interface error and (b) eikonal error $\|e_{\text{eik}}\|_{L^2(D)}$, showing initial error (blue) and predictor error (red).



(a) . (b)

Fig. 19. Torus level set contours before prediction-correction reinitialization: (a) view of a 2D cross-section of the level lines with the interface in 3D; (b) view of a cross section of the level set.

5.4. Simulation in three dimensions

As our reinitialization scheme is purely based on PDEs and a cut integration of the interface local to each cell, our method can be parallelized easily and can hence be used for complex 3D geometries. In this Section, we consider a torus-shaped level set function defined on the domain $D = \left[-\mathrm{L}/2, \mathrm{L}/2\right]^3$, with $\mathrm{L} = 2.5$ defined as follows:

$$\phi_{\text{init}}(x, y, z) = \left(R - \sqrt{x^2 + y^2}\right) + z^2 - r^2 \tag{87}$$

with, $R = 1.8$ and $r = 0.55$.

We use a tetrahedral mesh with size $h = \mathrm{L}/100$, resulting in over two million unknowns. Fig. 19 shows the initial level set which is not a signed distance function as can be seen by the densely packed surfaces close to the zero contour surface. After solving the predictor problem followed by 10 iterations of the corrector problem, Fig. 20 illustrates that the method gives a very good approximation of the signed distance function associated with the initial interface. The simulation was completed in 7 minutes and 54 seconds using 6 processors based on FEniCSx [29], CutFEMx [28] and PETSc [35].

Convergence analysis and HPC performance. The method's rapid convergence is evidenced by the evolution of errors over the correction steps. Fig. 21b illustrates that the eikonal error decreases rapidly during the initial iterations, stabilizing quickly towards a minimum. Concurrently, the interface error shown in Fig. 21a also rapidly decreases, confirming the scheme's ability to stabilize the interface position and conserve mass. For a comprehensive assessment of the asymptotic convergence behavior, the reinitialization was performed until the stringent stagnation criterion (39) ($\varepsilon = 10^{-8}$) was met. This demanding 3D benchmark required 500 iterations, consuming 117 GB of memory with a total wall clock time of 4 hours and 25 minutes using 80 processors. These results demonstrate

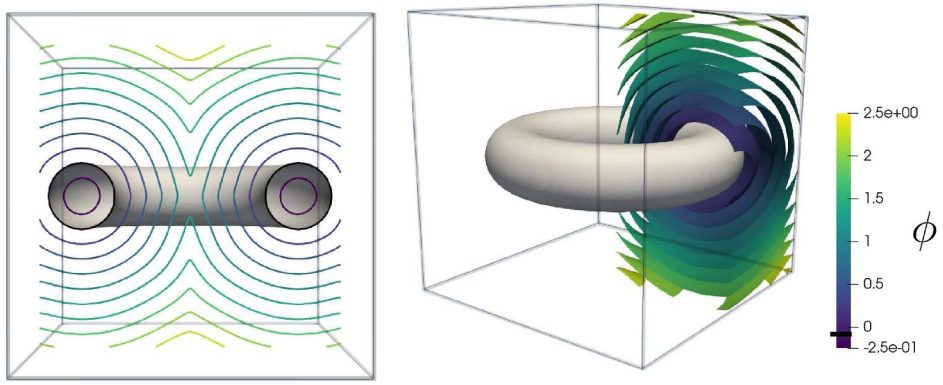


Fig. 20. Torus level set contours after prediction-correction reinitialization: (a) view of a 2D cross-section of the level lines with the interface in 3D; (b) view of a cross section of the level set.

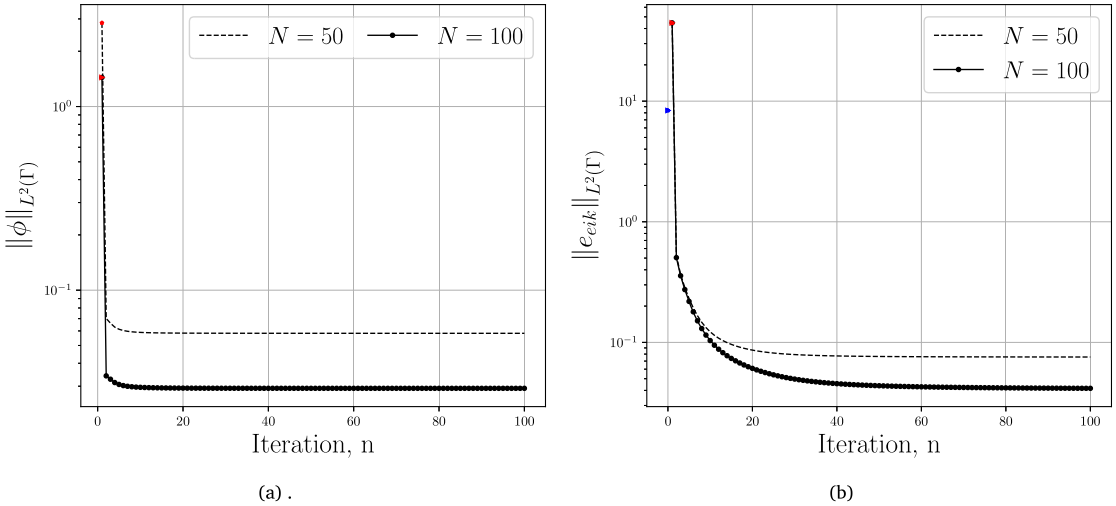


Fig. 21. Evolution of the error with respect to mesh size (h) for the torus domain example ($\gamma_D = 100$): (a) L^2 interface error and (b) eikonal error $\|e_{\text{eik}}\|_{L^2(D)}$, showing initial error (blue) and predictor error (red).

the feasibility and scalability of the proposed CutFEMx scheme for large-scale 3D problems on High-Performance Computing (HPC) architectures.

Having demonstrated our method's capability to handle complex three-dimensional geometries and its scalability on HPC architectures, we now evaluate its performance in a dynamic context. The following single vortex test subjects the algorithm to reversible deformations, thereby validating the accuracy and robustness of the reinitialization process not only on static shapes but also during complex convective transport.

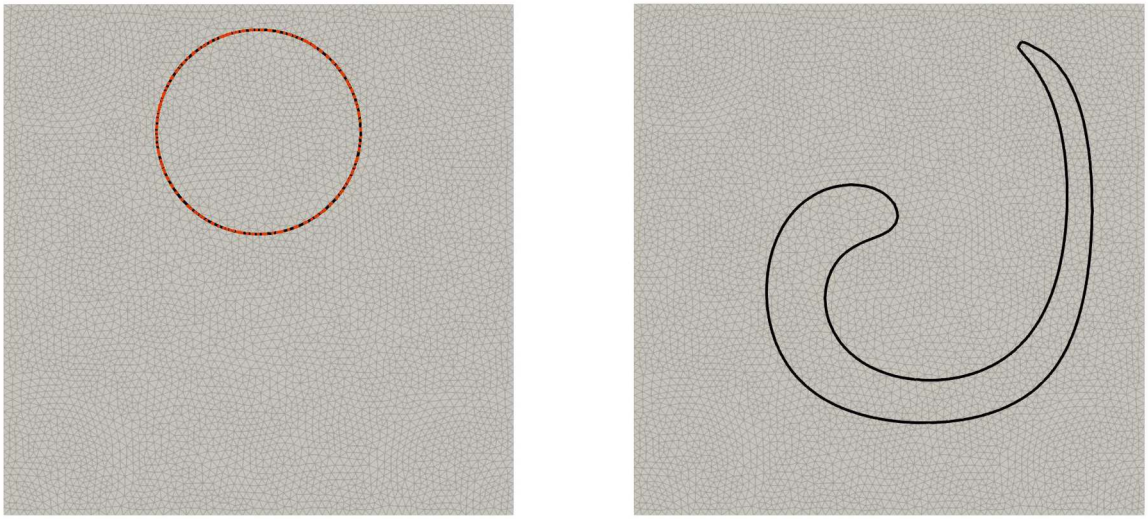
5.5. Single vortex test

The single vortex test, originally introduced by Rider and Kothe [36], is a standard benchmark used to assess the robustness of level-set methods under interface distortions. This test evaluates the ability of the reinitialization scheme to maintain the signed distance property without degrading the interface.

We consider an initial disk of radius $R = 0.2$ centered at $(x_0, y_0) = (0.5, 0.75)$ on a square domain $\Omega = [0, 1]^2$. The disk is advected by a time-dependent velocity field $\mathbf{u} = (u, v)^T$ defined as follows:

$$\begin{cases} u(x, y, t) = -\sin^2(\pi x)\sin(2\pi y)\cos\left(\frac{\pi t}{T_{inv}}\right) \\ v(x, y, t) = \sin^2(\pi y)\sin(2\pi x)\cos\left(\frac{\pi t}{T_{inv}}\right) \end{cases} \tag{88}$$

where T_{inv} represents the time at which the flow field is reversed. In this study, we set $T_{inv} = 1.5$, resulting in a total simulation time of $T = 3.0$. The inclusion of the cosine term ensures that the flow is perfectly reversible; theoretically, the interface should return to



(a) . (b)

Fig. 22. Qualitative evaluation of the single vortex test ($N = 64$): (a) interface recovery at $t = 3.0$ (black dotted line) compared to the exact interface (red dashed line); (b) maximum filament stretching at $t = 1.5$.

its initial circular shape and position at $t = T$. This property allows for a rigorous quantitative analysis of the geometric errors and mass conservation introduced by the numerical scheme.

The transport of the level-set function ϕ is governed by the pure advection equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{89}$$

where $\mathbf{u}$ is the velocity field defined above. To avoid the numerical diffusion inherent to Eulerian advection schemes and isolate the errors introduced by the reinitialization procedure, we use a semi-Lagrangian approach. For each degree of freedom in the mesh, the characteristic trajectories are integrated backward in time using a fourth-order Runge-Kutta (RK4) scheme. The advected level-set value is then determined by evaluating the exact initial signed distance function at the computed departure point. This method ensures that any deviation in mass conservation or interface geometry observed during the simulation is caused solely by the predictor-corrector reinitialization, rather than the advection step.

The reinitialization process is performed at each advection time step to ensure the level-set function ϕ remains a signed distance function throughout the simulation. In our implementation, this is achieved by applying one main prediction step followed by five additional correction steps where the penalty parameter is set to $\gamma_D = 10^2$.

During this transient test case, we observed some oscillations around the interface during the highly stretched phase of the simulation. These oscillations stem from the relatively coarse piecewise constant normal approximation used in the predictor-corrector scheme which seems insufficient for handling the distortions. We found that introducing a piecewise linear and continuous normal through an L^2 projection of the gradient defined through the following weak form:

$$\int_{D_h} \mathbf{g}_h \cdot \mathbf{v}_h \, dx = \int_{D_h} \nabla \phi_h \cdot \mathbf{v}_h \, dx \quad \forall \mathbf{v}_h \in [V_h]^d, \tag{90}$$

mitigates these oscillations and improves the overall stability of the reinitialization process. This mechanism was not visible in the static benchmarks presented earlier and seems to only manifest in the context of highly transient and distorted interfaces.

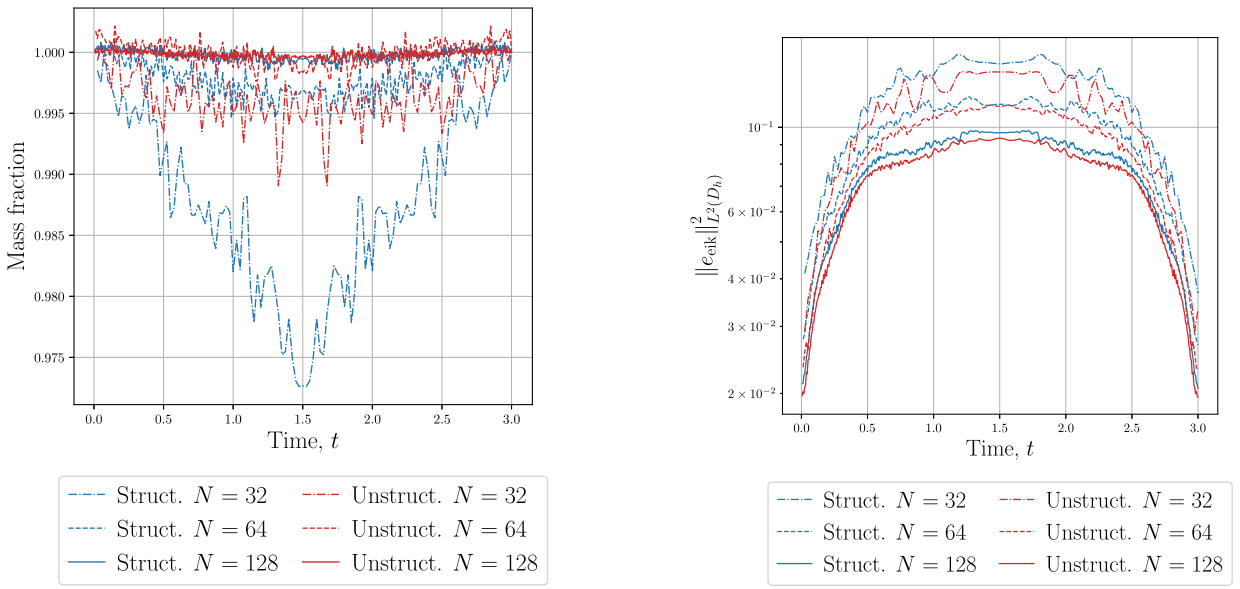
The continuous normal is then evaluated as $\mathbf{n}_\phi = \mathbf{g}_h / |\mathbf{g}_h|$ in the predictor-corrector scheme. This projection is performed rigorously before the prediction step and before each iteration of the correction step. This projection recovers robustness for the predictor-corrector scheme to robustly maintain the signed distance property even under complex distortions.

The single vortex test is evaluated on both structured and unstructured meshes.

To evaluate geometric fidelity, Fig. 22b illustrates the level-set interface at maximum stretching ($t = 1.5$) for an unstructured mesh ($h = 1/64$), where the fluid forms a thin filament. Fig. 22a displays the zero contour at the final time $t = 3.0$ in red (dashed line), superimposed on the exact initial circle in black (dotted line). The numerical solution recovers the initial geometry despite the intermediate distortion.

To quantify this geometric deviation, we define the shape error E_{shape} as the symmetric difference area between the numerical domain Ω_h and the exact analytical domain Ω_{ex}:

$$E_{shape} = \int_{\Omega} |H(\phi_h) - H(\phi_{ex})| \, d\mathbf{x} \tag{91}$$



(a) . (b)

Fig. 23. Temporal evolution of the numerical errors for the single vortex test ($N \in \{32, 64, 128\}$): (a) Relative mass error, (b) L^2-norm of the eikonal error.

Table 4
Relative shape error E_{shape} (%) at the final time $t = 3.0$ for structured and unstructured meshes.

Mesh type	$N = 32$	$N = 64$	$N = 128$
Unstructured	1.04e-3	2.46e-4	2.87e-5
Structured	1.04e-3	1.98e-4	5.68e-5

where H denotes the Heaviside function. The relative shape error is obtained by normalizing E_{shape} by the exact initial area of the disk. Table 4 reports the relative shape error E_{shape} at $t = 3.0$. Both mesh types exhibit similar accuracy for the coarse grid ($N = 32$). However, as the mesh is refined, the unstructured discretization shows a lower error magnitude compared to the structured grid. At the finest resolution ($N = 128$), the unstructured mesh error is approximately half that of the structured one. This performance improvement is attributed to the isotropic nature of the unstructured Delaunay triangulation, which mitigates the grid-aligned bias inherent to Cartesian meshes.

The temporal evolution of the relative mass error is presented in Fig. 23a for both mesh topologies. Because advection is solved via the semi-Lagrangian RK4 scheme, mass variations stem solely from the reinitialization step and sub-grid truncation. This truncation occurs when the filament thickness falls below the local cell size h. Unstructured meshes exhibit higher mass errors than structured grids. This difference arises because the Cartesian grid aligns with the primary axes of the vortex velocity field, which reduces interpolation errors during backward trajectory tracking. At finer resolutions, the mass loss stabilizes for both topologies by the end of the cycle, confirming the mass conservation capabilities of the continuous P_1 level-set method independent of grid alignment.

Fig. 23b shows the evolution of the L^2-norm of the eikonal equation error. For both mesh types, the error peaks near $t = 1.5$, corresponding to the maximum topological stretching where highly curved features become under-resolved. Consistent with the mass conservation results, unstructured grids yield a higher eikonal error magnitude due to the lack of directional alignment. However, as the velocity field reverses, the eikonal error decreases symmetrically, returning to its initial baseline magnitude at $t = 3.0$. This recovery indicates that the hybrid CutFEM predictor-corrector scheme maintains the signed distance property without accumulating numerical noise during the kinematic inversion.

6. Conclusions

In this article, we have presented a new finite element-based technique to compute the signed distance function starting from a level set function of general shape. Our method is based on a predictor-corrector formulation, where the predictor step is a linear diffusion problem with a discontinuous right-hand side and the corrector step is a nonlinear diffusion problem based on the least squares minimization of the eikonal equation. We presented both a finite element formulation for a mesh that fits to the zero contour line of the initial level set function as well as a cut finite element formulation that allows for a mesh that does not fit to the zero contour line.

We demonstrated that our predictor-corrector method is robust and converges to a signed distance function for a variety of initial level set functions, including smooth, discontinuous, and irregular functions. The method is also able to handle complex geometries in three dimensions and is easy to parallelize. We showed that the method is able to reinitialize the level set function with very few iterations, even for large meshes with millions of unknowns.

Furthermore, we have demonstrated the robustness of our scheme in a transient context through the vortex test, confirming that the method effectively maintains interface geometry and ensures mass conservation even under reversible deformations.

CRediT authorship contribution statement

Amina El Bachari: Writing – original draft, Methodology; **Johann Rannou:** Writing – review & editing; **Vladislav Yastrebov:** Writing – review & editing; **Pierre Kerfriden:** Writing – review & editing; **Susanne Claus:** Writing – review & editing, Methodology.

Declaration of competing interest

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

Acknowledgements

This research was supported by the French Aerospace Lab ONERA under internal research funding. The authors gratefully acknowledge this support.

Data availability

No data was used for the research described in the article.

References

[1] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003. https://doi.org/10.1007/b98879
[2] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Vol. 3 of Cambridge monographs on applied and computational mathematics, Cambridge University Press, 2 edition, 1999.
[3] G. Allaire, F. Jouve, A.-M. Toader, A level-set method for shape optimization, Comptes Rendus. Mathématique 334 (12) (2002) 1125–1130. https://doi.org/10.1016/S1631-073X(02)02412-3
[4] C. Dapogny, F. Feppon, Shape optimization using a level set based mesh evolution method: an overview and tutorial, Comptes Rendus. Mathématique 361 (2023) 1267–1332. https://doi.org/10.5802/crmath.498
[5] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. (1994). https://doi.org/10.1006/jcph.1994.1155
[6] F. Henri, Améliorations des méthodes Level Set pour l'impact de goutte de pluie, Theses, Université de Bordeaux, 2021. https://theses.hal.science/tel-03527520.
[7] S. Claus, P. Kerfriden, A cutFEM method for two-phase flow problems, Comput. Methods Appl. Mech. Eng. 348 (2019)185–206. https://doi.org/10.1016/j.cma.2019.01.009
[8] S. Osher, R. Tsai, Level set methods and their applications in image science, Commun. Math. Sci. 1 (2003). https://doi.org/10.4310/cms.2003.v1.n4.a1
[9] C. Li, C. Xu, C. Gui, M.D. Fox, Distance regularized level set evolution and its application to image segmentation, IEEE Trans. Image Process. 19 (2010) 3243–3254. https://doi.org/10.1109/TIP.2010.2069690
[10] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. U. S. A 93 (4) (1996) 1591–1595. https://doi.org/10.1073/pnas.93.4.1591
[11] D.L. Chopp, Computing minimal surfaces via level set curvature flow, J. Comput. Phys. 106 (1) (1993) 77–91. https://doi.org/10.1006/jcph.1993.1092
[12] R. Kimmel, J.A. Sethian, Computing geodesic paths on manifolds, Proc. Natl. Acad. Sci. 95 (15) (1998) 8431–8435. https://doi.org/10.1073/pnas.95.15.8431
[13] Y.-H.R. Tsai, L.-T. Cheng, S. Osher, H.-K. Zhao, Fast sweeping algorithms for a class of Hamilton-Jacobi equations, SIAM J. Numer. Anal. 41 (2) (2003) 673–694. https://doi.org/10.1137/S0036142901396533
[14] H. Zhao, A fast sweeping method for eikonal equations, Math. Comput. 74 (250) (2004) 603–627. https://doi.org/10.1090/S0025-5718-04-01678-3
[15] J. Qian, Y. Zhang, H. Zhao, Fast sweeping methods for eikonal equations on triangular meshes, SIAM J. Numer. Anal. 45 (1) (2007) 83–107. https://doi.org/10.1137/050627083
[16] M. Shakoor, B. Scholtes, P.-O. Bouchard, M. Bernacki, An efficient and parallel level set reinitialization method - Application to micromechanics and microstructural evolutions, Appl. Math. Model. 39 (23–24) (2015)7291–7302. https://doi.org/10.1016/j.apm.2015.03.014
[17] E. Carlini, M. Falcone, P. Hoch, A generalized fast marching method on unstructured triangular meshes, SIAM J. Numer. Anal. 51 (2013). https://doi.org/10.1137/110833610
[18] Z. Fu, R. Kirby, R. Whitaker, A fast iterative method for solving the eikonal equation on tetrahedral domains, SIAM J. Sci. Comput. : Publ. Soc. Ind. Appl. Math. 35 (2013) c473–c494. https://doi.org/10.1137/120881956
[19] M. Sussman, E. Fatemi, An efficient, interface-Preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM J. Sci. Comput. 20 (1999) 1165–1191. https://doi.org/10.1137/S1064827596298245
[20] G. Russo, P. Smereka, A remark on computing distance functions, J. Comput. Phys. 163 (2000) 51–67. https://doi.org/10.1006/jcph.2000.6553
[21] C. Min, On reinitializing level set functions, J. Comput. Phys. 229 (8) (2010) 2764–2772. https://doi.org/10.1016/j.jcp.2009.12.032
[22] A. Karakus, N. Chalmers, T. Warburton, A local discontinuous galerkin level set reinitialization with subcell stabilization on unstructured meshes, Comput. Math. Appl. 123 (2022) 160–170. https://doi.org/10.1016/j.camwa.2022.08.010
[23] C. Basting, D. Kuzmin, A minimization-based finite element formulation for interface-preserving level set reinitialization, Computing 95 (2013) 13–25. https://doi.org/10.1007/s00607-012-0259-z
[24] T. Adams, S. Giani, W.M. Coombs, A high-order elliptic PDE based level set reinitialisation method using a discontinuous Galerkin discretisation, J. Comput. Phys. 379 (2019) 373–391. https://doi.org/10.1016/j.jcp.2018.12.003
[25] T. Xue, W.C. Sun, S. Adriaenssens, Y. Wei, C. Liu, A new finite element level set reinitialization method based on the shifted boundary method, J. Comput. Phys. 438 (2021) 110360. https://doi.org/10.1016/j.jcp.2021.110360
[26] E. Burman, S. Claus, P. Hansbo, M. Larson, A. Massing, CutFEM: discretizing geometry and partial differential equations, Int. J. Numer. Methods Eng. 104 (2015). https://doi.org/10.1002/nme.4823

[27] A.R. Hashemi, M.R. Hashemi, P.B. Ryzhakov, R. Rossi, Optimization-based level-set re-initialization: a robust interface preserving approach in multiphase problems, Comput. Methods Appl. Mech. Eng. 420 (2024) 116699. https://doi.org/10.1016/j.cma.2023.116699
[28] S. Claus, CutFEMx: A Library for Unfitted Finite Element Methods on Complex Geometries, 2022, (https://github.com/sclaus2/cutfemx).
[29] I.A. Baratta, J.P. Dean, J.S. Dokken, M. Habera, J.S. Hale, C.N. Richardson, M.E. Rognes, M.W. Scroggs, N. Sime, G.N. Wells, DOLFINx: The next generation FEniCS problem solving environments, Zenodo (2023). https://doi.org/10.5281/zenodo.10447666
[30] A. El Bachari, Prediction-Correction-Implementation, 2025, (https://github.com/Amina-ELB/prediction-correction-implementation).
[31] S. Claus, P. Kerfriden, F. Moshfeghifar, S. Darkner, K. Erleben, C. Wong, Contact modeling from images using cut finite element solvers, Adv. Model. Simul. Eng. Sci. 8 (1) (2021). https://doi.org/10.1186/s40323-021-00197-2
[32] N. Molino, R. Bridson, J. Teran, R. Fedkiw, A crystalline, red green strategy for meshing highly deformable objects with tetrahedra, in: International Meshing Roundtable Conference, 2003, pp. 103–114. https://api.semanticscholar.org/CorpusID:701662.
[33] K. Erleben, H. Dohlmann, Signed distance fields using single-pass GPU scan conversion of tetrahedra, in: H. Nguyen (editor), GPU Gems 3, Addison-Wesley, 2008, pp. 741–763.
[34] J.C.C. Nitsche, Über ein variationsprinzip zur lösung von dirichlet-Problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, Abh. Math. Semin. Univ. Hambg. 36 (1971) 9–15. https://api.semanticscholar.org/CorpusID:122609590.
[35] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, PETSc Users Manual, Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015. http://www.mcs.anl.gov/petsc.
[36] W. Rider, D. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1999). https://doi.org/10.1006/jcph.1998.5906