MethodsX 14 (2025) 103313

Contents lists available at ScienceDirect

MethodsX

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

Level-Set field re-initialization: A computational model with finite element method on complicated domains ☆

Umer Siddiqui, Fahim Raees*

Department of Mathematics, NED University of Engineering and Technology, Karachi, Pakistan

ARTICLE INFO

Method name:
Re-initialization scheme for the Level-Set (LS) method

Keywords:
Lagrange multiplier method
Non-uniform grids
Level-set (LS) method
Finite element method (FEM)
Re-initialization, and multiphase flow

ABSTRACT

This paper introduces a proficient re-initialization method for the Level-Set (LS) field with the Finite Element Method (FEM) framework on unstructured meshes. The technique can retain the LS field's signed distance (SD) property but exploits the Eulerian-Lagrange multiplier technique. The scheme is based on geometric re-initialization and integrated with the FEM to higher-degree polynomials on complex domains. Numerical benchmark tests indicate the effectiveness and efficiency of the presented method, which also victoriously preserves the LS field's mass effectively, demonstrating its high performance.

- Enriched LS method simulations using efficient re-initialization with the FEM in complex domains.
- Based on geometry, the re-initialization scheme efficiently preserves the LS mass and applies to the higher-degree polynomials of the complex domains.
- The efficiency and effectiveness of the method revealed by the benchmark trials.

Specifications table

Subject area:	Mathematics and Statistics
More specific subject area:	Computational Fluid Dynamics (CFD), Numerical Analysis.
Name of your method:	Re-initialization scheme for the Level-Set (LS) method.
Name and reference of original method:	Not Applicable
Resource availability:	Not Applicable

Background

Multiphase flow refers to the action of more than one phase. Examining this phenomenon is an essential branch of fluid mechanics that applies to diverse industries. It involves the accompanying flow of materials with different characteristics, utilizing two or more thermodynamic phases. These phases may consist of a single chemical element, such as the diamond and graphite, liquid and solid mercury, or multiple chemical elements that interrelate, such as the steel (an alloy of multiple chemical elements) and salt. This phase naturally forms a continuous medium, either in liquid or gaseous form. The dispersed phase can include liquids, solids, and gases. This study highlights its broad relevance and significant consequence in multiple industrial fields, incorporating power generation,

☆ **Related research article:** Fahim Raees, Duncan Roy van der Heul, C. Vuik. A mass-conserving LS method for simulation of multiphase flow in geometrically complicated domains, *Int. J. Num. Meth. in Fluids* 81 (2015) 399–425. https://doi.org/10.1002/fld.4188

* Corresponding author.
E-mail address: fahimned@neduet.edu.pk (F. Raees).

https://doi.org/10.1016/j.mex.2025.103313
Received 10 December 2024; Accepted 11 April 2025
Available online 12 April 2025

food processing, medical science, and beverage production. The well-known and the famous methods for representing the multiphase flows are (i) the Eulerian Eulerian and (ii) the Eulerian-Lagrangian.

The Eulerian-Eulerian and Eulerian-Lagrangian strategies are pivotal in studying physics, particularly in the interchange of stages regarding impulse, mass interaction, heat, and turbulence.

A single domain acclimate two media with varying effects and phases in a two-phase flow model. Similar flows are prominent in industrial sectors, comprising the petroleum industry, chemical reactors, and medical sciences. The mechanism of two-phase flow may evolve problematic due to the motion of the interface and the distinguishable material attributes that vary with space and time. Similarly, the constricted flow does not encourage the interface's movement in numerous circumstances. Nevertheless, the defective intermolecular exchanges amid the two channels at the interface are probably accountable for the constricted accelerating flow.

Multifarious models have been created in past decennaries to simulate two-phase flow phenomena. Eminent techniques comprise computational fluid dynamics (CFD model), slip models, mixed-flow models, two-fluid models, and Euler-Euler models. Amid these, the best extensively employed and widely accepted approaches for biphasic flow simulations are (i) The mass-conserving LS method [1], (ii) The Smooth-Particle Hydrodynamics method [2], (iii) The volume of fluid (VOF) method [3], (iv) The Level-Set (LS) method [4], (v) Lattice Boltzmann method [5], (vi) The modified LS (MLS) method [2], and (vii) The marker particle method [6]. The LS method is a numerical method for detecting the moving interface. This method is simple for interface detection, but mass-conservation issues emerge, so it has to be re-initialized at diverse time steps so that it can conserve the LS mass efficiently. In this research, we introduce the optimum LS field re-initialization approach depending on the FEM for complex geometries. The re-initialization method is admiringly productive and efficacious for conserving the LS mass. Also, the introduced method depends upon the Eulerian-Lagrange multiplier technique. The introduced approach is based on geometry and conjoined with a finite element method applied to polynomials of higher degrees on complex geometries: the introduced method's efficiency and mass-conservation of the LS field exhibited by the benchmark trials.

Method details

In this study, we performed the multiphase flow simulation using the LS method with the volume of fluid fraction. This research presents the re-initialization scheme based on geometry for the LS method because mass-conservation complexities appear in the LS method.

The presented re-initialization scheme applies the enriched LS approach to simulate the complex fluid-fluid interfaces accompanied by proficient re-initialization strategies. The accurate capturing of the interface is permitted through this re-initialization method, including complex geometries. The proposed re-initialization method combined with the FEM preserves the LS mass efficiently and maintains simulation accuracy; also, this integration facilitates consistent and vigorous simulations for the complex interface challenges. The devised re-initialization method embraces the complex domain geometry and effectively assures the LS field's mass preservation. This re-initialization method permits applying the higher-degree polynomials to capture the exact complex interface. The benchmark trials reveal the accuracy, efficiency, and competence of the devised re-initialization method, which authenticates the consistency of precession and simulation. For the simulation of the complex fluid-fluid interfaces, the method's execution determines its capability in different engineering areas.

The level-set method

In 1987, two American mathematicians, *Stanley Osher* and *James Seithan*, presented the LS method to capture moving interfaces. The LS method is vigorous, effective, and proficient at correctly tracking a moving interface. The detection of moving an interface by applying the LS method is straightforward, though a mass preservation complexity has emerged. Commonly, the solution to a geometric partial differential equation characterizes an interface. The preliminary circumstance characterizes the moving interface effectively employing zero-LS, illustrated in two three-dimensional Euclidean spaces.

In the two-dimensional system, we examine the interface, illustrated by a moving curve described by a function $\Gamma(t)$ (or surface in 3D system) with a determinate region $\Omega \in \Re^2$ (this determinate region does not have to be open). We analyze the interface's velocity vector field $\mathbf{U} = (u_1, u_2)$, which is affected by the interface's position, temporal evolution, and geometric features.

The above notion defines the LS method, employing the LS function $\phi = (x, y, t)$ in one-dimensional and higher-dimensional systems. The LS method describes the positiveness above the interface, negativeness below one, and equality to zero alongside the contour indicated as $\phi = (x, y, t)$. This interface, a moving curve, invariably characterizes the current interface's position $\Gamma(t) = \{(x, y)|\phi(x, y, t)\}$ at zero and its contour. Mathematically, the LS equation read as follows:

$$\phi_t + \mathbf{U} \cdot \nabla\phi = 0 \tag{1}$$

Where

ϕ_t is the variation in an interface with time *t*.

U demonstrates the velocity vector field with zero divergence i.e. $\nabla \cdot \mathbf{U} = 0$.

$\nabla\phi$ is the interface's gradient function

The LS function sustains the signed distance (SD) property, mathematically, $|\nabla\phi| = 1$, and ϕ illustrates the interface's SD function. Researchers can probe the computational value of the SD function in both 2-D and 3-D problems of the LS method as,

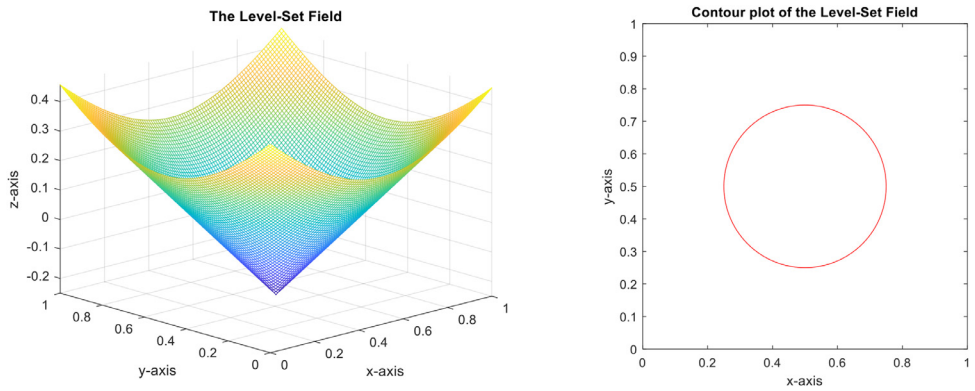

Fig. 1. Instance of the Level-Set method with its contour.

$$\sqrt{\phi_x^2 + \phi_y^2} \tag{2}$$

$$\sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2} \tag{3}$$

Where *x, y* represent the LS dimension an illustration of the LS method, besides its contour depiction, is exemplified in Fig. 1.

In examining image processing, the Chan-Vese segmentation model in [7] is generally recognized as an "active contour free from the edges," eradicating the need for edge detection. The LS method verifies that it is sufficient to address the Chan-Vese model. Researchers simplified the Chan-Vese segmentation model using the LS approach with different strategies to address this problem efficiently. The LS method desires to illustrate the LS function based on the contours. The LS function categorizes the segmented areas into three parts: (i) the starting interface, (ii) the exterior interface, and (iii) the interior interface. At the interface, the LS value is zero; inside the interior interface, the LS value is positive, and the LS negative value is inside the exterior interface. Involving the LS method in resolving the Chan-Vese segmentation model to illustrate the LS function is a substantial complexity. Over the past decades, researchers have submitted numerous approaches to specifying the LS function; the checkerboard method, as seen in [8], is one of those methods that count by thresholding.

The LS method is a computationally efficient solver that provides the proficient solution for solving the Hamilton-Jacobi (HJ) equations numerically; the LS method utilizes the HJ equations to refine the LS function as in [9], enabling steep gradients to appear near the edges. Regardless, the restricted exactitude of LS functions may result in inadequate boundary estimation. One strategy to enrich this is to maintain the exactitude of slopes when re-initializing the LS function to the SD function after numerous iterations in the advancement process. In this level, the FEM shape functions resemble the zero LS (isocontour), which authorizes the estimation of the discretized isocontour distances. The LS method restrains the intricate topological transitions over the advancement process; however, the structure of initial geometry depends on the final and best solution because of the conventional LS method; researchers have used this method throughout the evolution of two-dimensional structural optimization problems.

Introduced re-initialization scheme

This section presents an unexplored re-initialization procedure for the LS method and examines its significance in such an investigation. Through the LS method, re-initialization challenges appear related to mass preservation. To cope with this challenge, researchers have presented numerous techniques in recent decennaries, incorporating the mass-conserving initialization method based on geometry in [10] and the re-initialization method relies on the partial differential equation (PDE), see in [11], among others.

In this study, we introduced a geometric-based approach for the re-initialization strategy. We implemented it employing the FEM; the essential notion is to fit a second-degree polynomial within the domain of each triangular mesh element and determine the smallest distance from this polynomial (defining the interface) to all nodes through the triangular element with the Lagrange multiplier approach. We noticed that the values of the LS function remain uniform over the interface, with LS values being negative (exhibiting a below position from an interface) and positive (showing an above position from an interface), namely $\mathrm{sgn}(\phi)$, where ϕ characterizes the LS values and sgn reveals the LS sign that isolates negative (below) and positive (above) values from the interface. We calculate the smallest distance from the polynomial (the interface) to the individual node in the triangular mesh. Thereon, the new LS values are then updated according to a particular mathematical expression.

$$\phi_{updated} = \mathrm{sgn}(\phi) \cdot |s| \tag{4}$$

Where,

$\phi_{updated}$ represents the new updated LS values.

ϕ denotes the exact LS values.

$|\ s\ |$ indicates the absolute distance (smallest) between the interface and all triangular mesh element nodes.

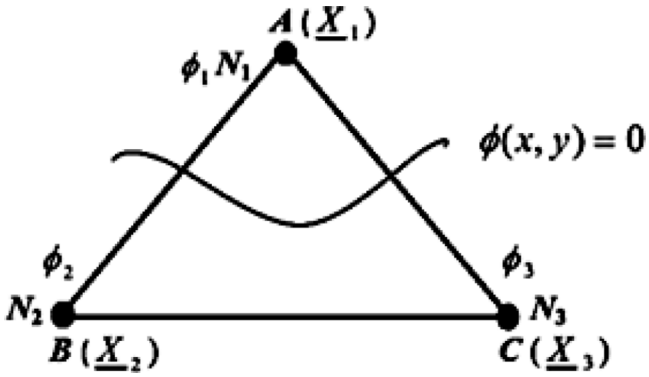

Fig. 2. Schematic diagram of triangular mesh element with nodes.

In this research, applying the finite element analysis, we can write the polynomial equation on the triangular mesh element, consequently, the subsequent form of the equation is

$$\phi(x,y) = \sum_{i=1}^{6} N_i \phi_i, \tag{5}$$

Where N_i is the shape function of the triangular mesh element with index i and ϕ_i is the exact LS value. We can now estimate the shape functions values with the triangular mesh element using the inverse of the Vandermonde matrix; the Vandermonde matrix is,

$$V = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4y_4 & x_4^2 & y_4^2 \\ 1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2 \\ 1 & x_6 & y_6 & x_6y_6 & x_6^2 & y_6^2 \end{bmatrix}, M^{-1} = \begin{bmatrix} \alpha_{01} & \alpha_{02} & \alpha_{03} & \alpha_{04} & \alpha_{05} & \alpha_{06} \\ \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} & \alpha_{15} & \alpha_{16} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} & \alpha_{25} & \alpha_{26} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} & \alpha_{35} & \alpha_{36} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} & \alpha_{45} & \alpha_{46} \\ \alpha_{51} & \alpha_{52} & \alpha_{53} & \alpha_{54} & \alpha_{55} & \alpha_{56} \end{bmatrix}$$

Where M^{-1} is the inverse of the Vandermonde matrix.
We can, as a result of this, write the elements of triangular mesh with the shape function as

$$N_i = \alpha_{0i} + \alpha_{1i}x + \alpha_{2i}y + \alpha_{3i}xy + \alpha_{4i}x^2 + \alpha_{5i}y^2, 1 \le i \le 6 \tag{6}$$

Where $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ are the constants terms in Eq. (6) and they having constant values per triangular element.
The solution of the Eq. (5) is

$$\phi(x,y) = \varepsilon_0 x + \varepsilon_1 y + \varepsilon_2 xy + \varepsilon_3 x^2 + \varepsilon_4 y^2 + \varepsilon_5 \tag{7}$$

In Eq. (7), $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$, and ε_5 are the equation's coefficients and also the computational values are invariant for the equation's coefficients. The computational value of $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ and ε_5 are

$$\varepsilon_k = \sum_{i=1}^{6} \beta_i \delta_i, 0 \le k \le 5$$

Evolution of the scheme

This subsection implements the FEM accompanied by the Lagrange multiplier technique, employing the FEM for examining Eq. (5). Afterwards, we use the distance formula, which applies the Lagrange multiplier method to estimate the smallest distance from the interface through the triangular mesh element to each node of an element. We express the distance formula as follows,

$$|s| = \sqrt{(x_n - x_b)^2 + (y_n - y_b)^2}, 1 \le n \le 3 \tag{8}$$

In the Eq. (8)
$|\ s\ |$ represents the Euclidean distance between the nodes and the interface
(x_n, y_n) indicates the nodes coordinates of an element
(x_b, y_b) indicate the interface's placed point coordinates, namely $\phi(x_b, y_b) = 0$

Working of the scheme

This research studies the triangular mesh elements, including their coordinates $\underline{X}_i, 1 \le i \le 3$ with their various shape functions $N_i, 1 \le i \le 3$ along with the LS values $\phi_i, 1 \le i \le 3$ as shown in Fig. 2

We have specified that for node $\underline{X}_1$, the LS function value is represented as ϕ_1, node $\underline{X}_2$ as ϕ_2, and node $\underline{X}_3$ as ϕ_3, in order. We have fitted a polynomial over the triangular mesh element after for the computation of the smallest distance from the polynomial to the interface accompanied by the triangular mesh element node, where the polynomial's value equals zero, mathematically $\phi(x, y) = 0$. We apply the Lagrange multiplier method to determine the smallest distances, optimizing the function under the condition that $\phi(x_b, y_b) = 0$

Explanation and derivation for the Lagrange multiplier approach

In mathematical optimization, this examination implements the Lagrange multiplier approach for optimizing a function dependent on equality conditions. More specifically, we strive to minimize the objective function under the condition that

$$\text{Minimize } s(x, y)$$

$$\text{Satisfying the condition } \phi(x, y) = 0$$

In the above condition,

$s(x, y)$ indicates the distance metric.

$\phi(x, y)$ shows the LS function.

We assume that the two functions $s(x, y)$ and $\phi(x, y)$ of first-order partial derivatives mentioned earlier are continuous-valued. Then, for enriched understanding, we introduce a new factor (γ) referred to as the Lagrange multiplier and study it as follows,

$$L(x, y, \gamma) = s(x, y) \pm \gamma \cdot \phi(x, y) \tag{9}$$

The factor γpotentially add or subtract in Eq. (9).

Derivation

$$C = s(x_b, y_b) + \gamma\phi(x_b, y_b)$$

$$\min s(x_b, y_b) \text{ such that } \phi(x_b, y_b) = 0$$

$$\frac{\partial C}{\partial x_b} = \frac{\partial s}{\partial x_b} + \gamma\frac{\partial \phi}{\partial x_b} = 0$$

$$\frac{\partial C}{\partial y_b} = \frac{\partial s}{\partial y_b} + \gamma\frac{\partial \phi}{\partial y_b} = 0$$

$$\frac{\partial C}{\partial \gamma} = \phi(x_b, y_b) = 0$$

$$s = \sqrt{(x_n - x_b)^2 + (y_n - y_b)^2} \rightarrow R = s^2 = (x_n - x_b)^2 + (y_n - y_b)^2 1 \leq n \leq 3$$

$$\phi(x_b, y_b) = \varepsilon_0 x_b + \varepsilon_1 y_b + \varepsilon_2 x_b y_b + \varepsilon_3 x_b^2 + \varepsilon_4 y_b^2 + \varepsilon_5 = 0$$

Where

(x_b, y_b) are the interface's unidentified points even as (x_n, y_n) are indentified points. $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ and ε_5 are fitted polynomial's coefficients and incorporates negative and positive consistent values.

$$\frac{\partial C}{\partial x_b} = \frac{\partial R}{\partial x_b} + \gamma\frac{\partial \phi}{\partial x_b}$$
$$\frac{\partial C}{\partial x_b} = 2(x_b - x_n) + \gamma(\varepsilon_0 + \varepsilon_2 y_b + 2\varepsilon_3 x_b) = 0$$
$$2(1 + \varepsilon_3\gamma)x_b + \gamma\varepsilon_2 y_b = 2x_n - \gamma\varepsilon_0 \tag{10}$$

$$\frac{\partial C}{\partial y_b} = \frac{\partial R}{\partial y_b} + \gamma\frac{\partial \phi}{\partial y_b}$$
$$\frac{\partial C}{\partial y_b} = 2(y_b - y_n) + \gamma(\varepsilon_1 + \varepsilon_2 x_b + 2\varepsilon_4 y_b) = 0$$
$$\gamma\varepsilon_2 x_b + 2(\varepsilon_4\gamma + 1)y_b = 2y_n - \gamma\varepsilon_1 \tag{11}$$

$$\frac{\partial C}{\partial \gamma} = \frac{\partial R}{\partial \gamma} + \gamma\frac{\partial \phi}{\partial \gamma}$$
$$\frac{\partial C}{\partial \gamma} = \varepsilon_0 x_c + \varepsilon_1 y_b + \varepsilon_2 x_b y_b + \varepsilon_3 x_b^2 + \varepsilon_4 y_b^2 + \varepsilon_5 = 0$$
$$\varepsilon_0 x_c + \varepsilon_1 y_b + \varepsilon_2 x_b y_b + \varepsilon_3 x_b^2 + \varepsilon_4 y_b^2 + \varepsilon_5 = 0 \tag{12}$$

Multiplying Eq. (10) by 2 $(\varepsilon_4\gamma + 1)$ and Eq. (11) by $\gamma\varepsilon_2$, we get,

$$x_b = \frac{2(\varepsilon_4\gamma + 1)(2x_n - \gamma\varepsilon_0) - \gamma\varepsilon_2(2y_n - \gamma\varepsilon_1)}{(4(1 + \varepsilon_3\gamma)(1 + \varepsilon_4\gamma) - \gamma^2\varepsilon_2^2)}$$

Eq. (10) multiply by $\gamma\varepsilon_2$, and Eq. (11) by 2 $(1 + \varepsilon_3\gamma)$, gives

$$y_b = \frac{\gamma\varepsilon_2(2x_n - \gamma\varepsilon_0) - 2(1 + \varepsilon_3\gamma)(2y_n - \gamma\varepsilon_1)}{(\gamma^2\varepsilon_2^2 - 4(1 + \varepsilon_3\gamma)(1 + \varepsilon_4\gamma))}$$

$$y_b = -\frac{\big(\gamma\varepsilon_2(2x_n - \gamma\varepsilon_0) - 2(1 + \varepsilon_3\gamma)(2y_n - \gamma\varepsilon_1)\big)}{(4(1 + \varepsilon_3\gamma)(1 + \varepsilon_4\gamma) - \gamma^2\varepsilon_2^2)}$$

$$\underline{\mathbf{H}}(x_b, y_b) = \begin{pmatrix} \frac{2(\varepsilon_4\gamma+1)(2x_n-\gamma\varepsilon_0)-\gamma\varepsilon_2(2y_n-\gamma\varepsilon_1)}{(4(1+\varepsilon_3\gamma)(1+\varepsilon_4\gamma)-\gamma^2\varepsilon_2^2)}, \\ -\frac{(\gamma\varepsilon_2(2x_n-\gamma\varepsilon_0)-2(1+\varepsilon_3\gamma)(2y_n-\gamma\varepsilon_1))}{(4(1+\varepsilon_3\gamma)(1+\varepsilon_4\gamma)-\gamma^2\varepsilon_2^2)} \end{pmatrix} \tag{13}$$

Where, (x_b, y_b) indicates the point $\underline{\mathbf{H}}$ coordinates, which are positioned at the interface
Now, we replace the x_b and y_b values in Eq. (12), yields,

$$\varepsilon_0 x_c + \varepsilon_1 y_b + \varepsilon_2 x_b y_b + \varepsilon_3 x_b^2 + \varepsilon_4 y_b^2 + \varepsilon_5 = 0$$

Solving the above mathematical relation so it generates the fourth-degree polynomial for γ

$$\begin{aligned}
&\big(8\varepsilon_3\varepsilon_4\varepsilon_5 - \varepsilon_3\varepsilon_1^2 - 4\varepsilon_3^2\varepsilon_1^2\varepsilon_4 + 4\varepsilon_0\varepsilon_1\varepsilon_2\varepsilon_3\varepsilon_4 - 4\varepsilon_0^2\varepsilon_4^2\varepsilon_3 + 16\varepsilon_3^2\varepsilon_4^2\varepsilon_5 + \varepsilon_0\varepsilon_1\varepsilon_2 - \varepsilon_0^2\varepsilon_4 + \varepsilon_5\big)\gamma^4 \\
&\quad + \big(-8\varepsilon_0^2\varepsilon_4^2 + 8\varepsilon_0\varepsilon_1\varepsilon_2\varepsilon_4 + 32\varepsilon_3^2\varepsilon_4\varepsilon_5 + 8\varepsilon_4\varepsilon_5 + 32\varepsilon_3\varepsilon_4^2\varepsilon_5 - 8\varepsilon_0^2\varepsilon_3\varepsilon_4 + 8\varepsilon_3\varepsilon_5 - 8\varepsilon_1^2\varepsilon_3^2 - 8\varepsilon_1^2\varepsilon_3\varepsilon_4 + 8\varepsilon_0\varepsilon_1\varepsilon_2\varepsilon_3\big)\gamma^3 \\
&\quad + \big(-8\varepsilon_1\varepsilon_2\varepsilon_4 x_n - 16\varepsilon_2\varepsilon_3\varepsilon_4 x_n y_n + 16\varepsilon_0\varepsilon_4^2 x_n + 12\varepsilon_0\varepsilon_1\varepsilon_2 - 4\varepsilon_0^2\varepsilon_3 - 16\varepsilon_0^2\varepsilon_4 - 8\varepsilon_1\varepsilon_2\varepsilon_3 x_n - 8\varepsilon_0\varepsilon_2\varepsilon_3 y_n - 4\varepsilon_0 x_n + 16\varepsilon_4^2\varepsilon_5 \\
&\quad - 8\varepsilon_0\varepsilon_2\varepsilon_4 y_n + 16\varepsilon_3^2\varepsilon_5 - 4\varepsilon_1 y_n + 8\varepsilon_5 - 4\varepsilon_2 x_n y_n + 16\varepsilon_1\varepsilon_3^2 y_n + 16\varepsilon_3\varepsilon_4^2 x_n^2 - 16\varepsilon_1^2\varepsilon_3 + 16\varepsilon_3^2\varepsilon_4 y_n^2 + 64\varepsilon_3\varepsilon_4\varepsilon_5 + 4\varepsilon_3 y_n^2 \\
&\quad + 4\varepsilon_4 x_n^2 - 4\varepsilon_1^2\varepsilon_4\big)\gamma^2 + \big(32\varepsilon_3\varepsilon_5 + 32\varepsilon_3\varepsilon_4 y_n^2 - 16\varepsilon_0\varepsilon_2 y_n - 8\varepsilon_0^2 - 16\varepsilon_1\varepsilon_2 x_n - 8\varepsilon_1^2 + 32\varepsilon_4\varepsilon_5 + 8y_n^2 + 32\varepsilon_0\varepsilon_4 x_n + 32\varepsilon_3\varepsilon_4 x_n^2 \\
&\quad + 32\varepsilon_1\varepsilon_3 y_n + 8x_n^2\big)\gamma + 16\varepsilon_5 + 16\varepsilon_1 y_n + 16\varepsilon_2 x_n y_n + 16\varepsilon_4 y_n^2 + 16\varepsilon_0 x_n + 16\varepsilon_3 x_n^2.
\end{aligned}$$

Therefore, the final version is

$$\xi_0\lambda^4 + \xi_1\lambda^3 + \xi_2\lambda^2 + \xi_3\lambda + \xi_4 = 0 \tag{14}$$

Where $\xi_0, \xi_1, \xi_2, \xi_3$ and ξ_4 are constant in Eq. (13) also the Eq. (13) shows the fourth-degree polynomial for γ, and we estimate the smallest distances from the interface to the triangular mesh element nodes with the distance formula using γ values.
Where,

$$\begin{aligned}
\xi_0 &= 8\varepsilon_3\varepsilon_4\varepsilon_5 - \varepsilon_3\varepsilon_1^2 - 4\varepsilon_3^2\varepsilon_1^2\varepsilon_4 + 4\varepsilon_0\varepsilon_1\varepsilon_2\varepsilon_3\varepsilon_4 - 4\varepsilon_0^2\varepsilon_4^2\varepsilon_3 + 16\varepsilon_3^2\varepsilon_4^2\varepsilon_5 + \varepsilon_0\varepsilon_1\varepsilon_2 - \varepsilon_0^2\varepsilon_4 + \varepsilon_5 \\
\xi_1 &= -8\varepsilon_0^2\varepsilon_4^2 + 8\varepsilon_0\varepsilon_1\varepsilon_2\varepsilon_4 + 32\varepsilon_3^2\varepsilon_4\varepsilon_5 + 8\varepsilon_4\varepsilon_5 + 32\varepsilon_3\varepsilon_4^2\varepsilon_5 - 8\varepsilon_0^2\varepsilon_3\varepsilon_4 + 8\varepsilon_3\varepsilon_5 - 8\varepsilon_1^2\varepsilon_3^2 - 8\varepsilon_1^2\varepsilon_3\varepsilon_4 + 8\varepsilon_0\varepsilon_1\varepsilon_2\varepsilon_3 \\
\xi_2 &= -8\varepsilon_1\varepsilon_2\varepsilon_4 x_n - 16\varepsilon_2\varepsilon_3\varepsilon_4 x_n y_n + 16\varepsilon_0\varepsilon_4^2 x_n + 12\varepsilon_0\varepsilon_1\varepsilon_2 - 4\varepsilon_0^2\varepsilon_3 - 16\varepsilon_0^2\varepsilon_4 - 8\varepsilon_1\varepsilon_2\varepsilon_3 x_n - 8\varepsilon_0\varepsilon_2\varepsilon_3 y_n - 4\varepsilon_0 x_n \\
&\quad + 16\varepsilon_4^2\varepsilon_5 - 8\varepsilon_0\varepsilon_2\varepsilon_4 y_n + 16\varepsilon_3^2\varepsilon_5 - 4\varepsilon_1 y_n + 8\varepsilon_5 - 4\varepsilon_2 x_n y_n + 16\varepsilon_1\varepsilon_3^2 y_n + 16\varepsilon_3\varepsilon_4^2 x_n^2 - 16\varepsilon_1^2\varepsilon_3 + 16\varepsilon_3^2\varepsilon_4 y_n^2 + 64\varepsilon_3\varepsilon_4\varepsilon_5 \\
&\quad + 4\varepsilon_3 y_n^\chi + 4\varepsilon_4 x_n^2 - 4\varepsilon_1^2\varepsilon_4 \\
\xi_3 &= 32\varepsilon_3\varepsilon_5 + 32\varepsilon_3\varepsilon_4 y_n^2 - 16\varepsilon_0\varepsilon_2 y_n - 8\varepsilon_0^2 - 16\varepsilon_1\varepsilon_2 x_n - 8\varepsilon_1^2 + 32\varepsilon_4\varepsilon_5 + 8y_n^2 + 32\varepsilon_0\varepsilon_4 x_n + 32\varepsilon_3\varepsilon_4 x_n^2 \\
\xi_4 &= 16\varepsilon_5 + 16\varepsilon_1 y_n + 16\varepsilon_2 x_n y_n + 16\varepsilon_4 y_n^2 + 16\varepsilon_0 x_n + 16\varepsilon_3 x_n^2
\end{aligned}$$

It is essential to note that if the entire polynomial's roots are not real, consequently, in this circumstance, the actual LS values (ϕ) represent the smallest distances s.

The discontinuous Galerkin finite element method (DG-FEM) presented in this study is advantageous and has some great attributes. It provides higher-order accurateness, the ability to cope with complex geometries and unstructured meshes, and the ability to locally conserve physical quantities like energy, momentum, and mass. The DG-FEM can parallel assess the mesh element independently and is more robust than other methods such as Essentially Non-Oscillatory scheme (ENO), Weighted Essentially Non-Oscillatory scheme (WENO), Finite Difference method (FDM), or Finite Volume method (FVM). The preliminary objective of using the DG-FEM in this research is to locally re-initialize the LS method. Also, this method catches the discontinuities correctly and enriches the convergence rate and the method's stability. This method employs the boundary conditions efficiently and can control non-linear problems. The DG-FEM enriched the error approximation and adaptivity to select the basis functions and minimize numerical diffusion.

Research structure chart

We plot the subsequent structure chart for the performed research and provide its explanation below. Fig. 3

Step I: Initially, we take the LS function ϕ for the LS re-initialization. In the LS method, the function ϕ represents interface or boundaries.

Step II: We start re-initialization by evaluating the shape functions using the Vandermonde matrix.

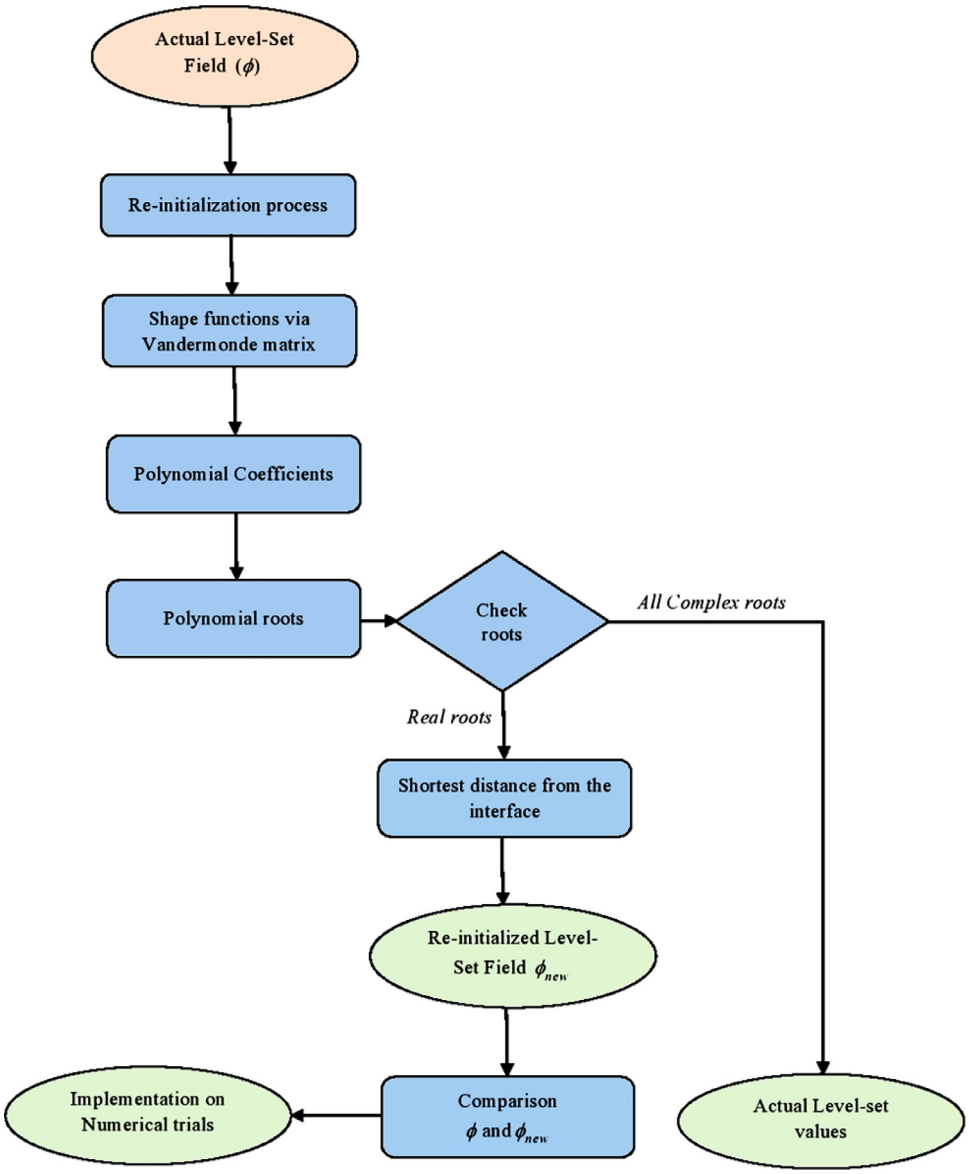

Fig. 3. The Structure's chart for the performed research.

Step III: Using the shape functions, we will compute the polynomial coefficients,
Step IV: we will obtain the roots of the fitted polynomial.
Step V: Check the roots of the polynomial; if real roots appear, we will calculate the shortest distance from the interface, and if all roots are imaginary in this situation, the actual LS values will represent the shortest distance.
Step VI: Formulate the new updated re-initialized LS field ϕ_{new} using the shortest distances.
Step VII: we compared the Actual LS field ϕ and the new updated re-initialized LS field ϕ_{new}.
Step VIII: Validate the formulated LS re-initialization scheme through the benchmark test cases

The actual LS field's visualization before and after re-initialization
Fig. 4 and Fig. 5

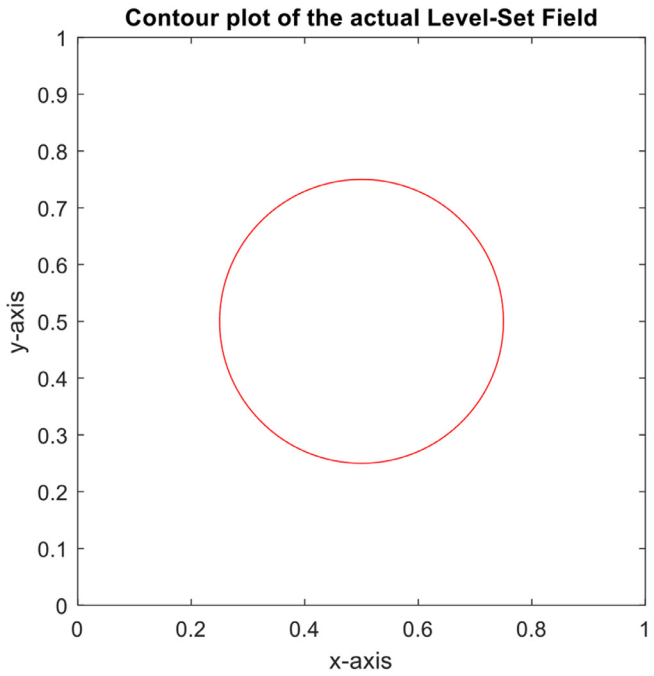

Fig. 4. Contour plot of the actual Level-Set (LS) Field before re-initialization.

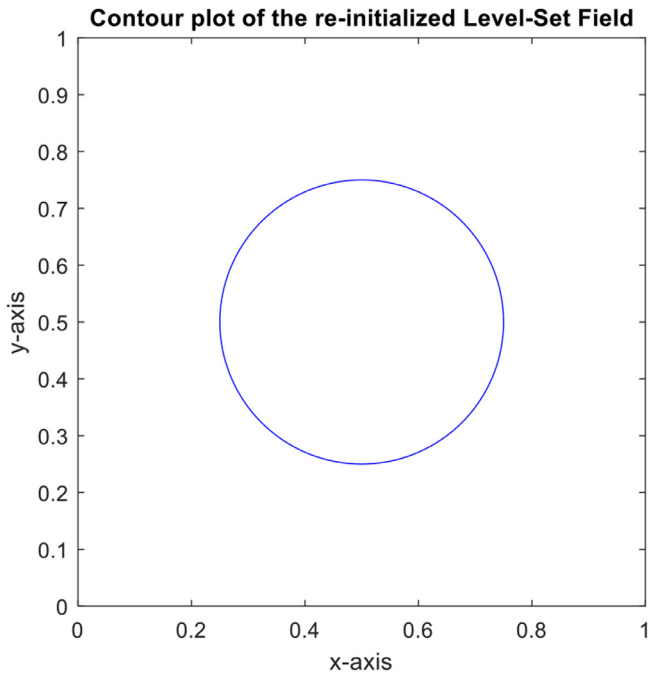

Fig. 5. Contour plot of the actual Level-Set (LS) Field after re-initialization.

Table 1
The re-initialized LS Field Convergence with Euclidean-norm and infinity-norm errors.

Mesh length (*h*)	*Euclidean- norm error (units)*	*Convergence order*	*Infinity-norm error (units)*	*Convergence order*
0.01	0.31787	—	0.15114	—
0.005	0.29211	1.08820	0.11503	2.09107
0.0025	0.11203	2.60743	0.05521	2.08345

Re-initialized LS field's with norm and errors

At mesh length $h = 0.01$, the convergence order cannot be calculated because the previous data is not given for the comparison. We compute the convergence order by comparing the data of two different mesh lengths. In this situation, it is inconceivable to estimate the convergence order.

Table 1 indicates the re-initialized LS field at different mesh lengths. The table provides the Infinity norm error and the Euclidean norm error together with a convergence order at all mesh length given in the Table 1. Infinity norm error gives the second-order convergence, whereas the Euclidean norm error provides the linear convergence. The results indicate that proposed re-initialization scheme for the LS field provides the good convergence behaviour both in the Euclidean norm error and in the Infinity norm error.

Method's application

The proposed LS re-initialization scheme can be applied to plasma because plasma is the ionized gas that comprises ions, neutral particles and electrons. We can simulate the behaviour of the plasma in complex geometries with the LS field re-initialization, which occurs in the plasma etching simulators or the thermonuclear reactions. Also, it has several applications in computational fluid dynamics; it incorporates (i) accurately capturing the moving interface, (ii) efficiently maintaining the signed distance function in the LS simulations, (iii) solving the hyperbolic conservative law with discontinuities, and (iv) It improves the image segmentation and enhances mesh quality and interface resolution. The re-initialization method is also further applicable in moving boundary problems, fluid-structure interaction, topology optimization, inverse problems, phase-field models and shape optimization. The proposed re-initialization method efficiently deals with the challenging issues of science and engineering.

Method validation

No experimental work has been carried out in this research; the presented research work is validated through benchmark test cases employed in the multiphase flow simulations, namely

(i) The numerical simulation of the cross-sectional contour of a lens-shaped interface during advection
(ii) Zalesak's rotational flow computations

Numerical simulation of cross-sectional contour of a lens-shaped interface during advection

At the outset, the circular-shaped bubble is the most renowned and widely used test case for assessing the incompressible two-phase flow model. This test case depends on the material's liquid characteristics, the density proportion between the two phases, and the changes in the bubble's contours from the initial to the final stage.

Consider the computational domain $\Omega = [0, 1] \times [0, 1]$, we acquire the LS field solution, the initial condition specified as

$$\phi_l(\mathbf{Y}, 0) = \max\left\{\phi_1(\mathbf{Y}, 0), -\phi_2(\mathbf{Y}, 0)\right\} \tag{14a}$$

Where,

$$\phi_1(\mathbf{Y}, 0) = \left|\mathbf{Y} - \mathbf{Y}_1^d(0)\right| - S \tag{15}$$

$$\phi_2(\mathbf{Y}, 0) = \left|\mathbf{Y} - \mathbf{Y}_2^d(0)\right| - S \tag{16}$$

Here, the computational values are supposed to be $\mathbf{Y}_1^d(0) = (0.45, 0.15)^T, \mathbf{Y}_2^d(0) = (0.45, 0.4)^T$ and $S = 0.25$. As a result, Fig. 6 and Fig. 7 illustrate the bubble advection re-initialization using the lens-shaped interface before and after re-initialization, respectively.

Re-initialized bubble with a lens shape interface with norms and errors

Table 2 exhibits the bubble's re-initialized LS field convergence results with a lens-shaped interface at different mesh lengths during advection. The table provides the Infinity norm error and the Euclidean norm error together with a convergence order at all mesh lengths given in Table 2. The infinity norm error and the Euclidean norm error both give linear convergence in this test case. The test results reveal that the proposed re-initialization method for the LS field is first-order accurate for this test case.

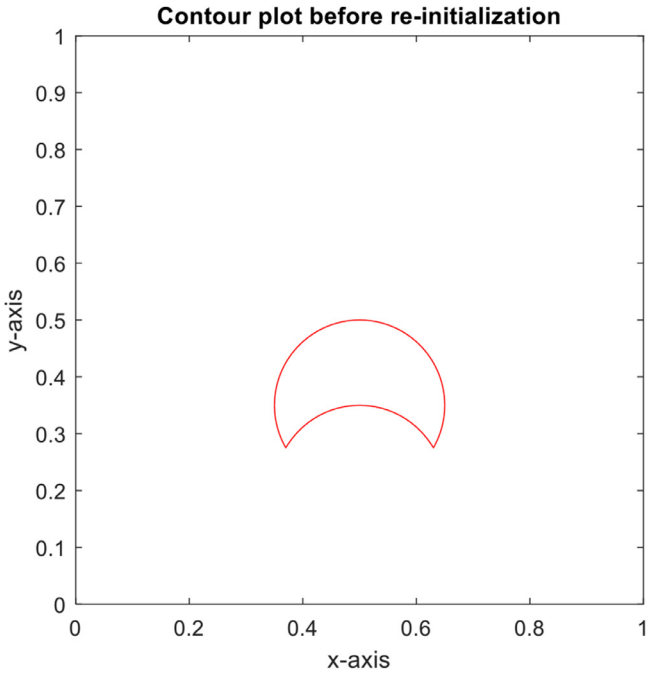

Fig. 6. Bubble with a lens-shaped interface before re-initialization.

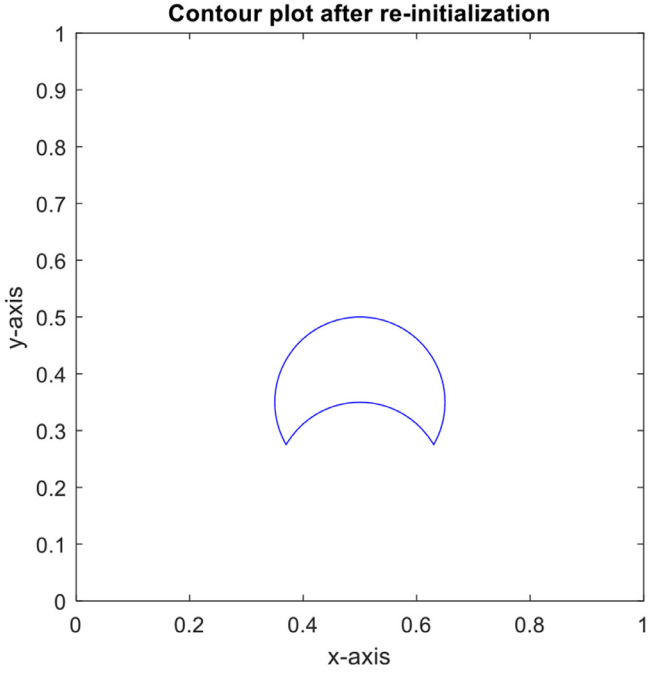

Fig. 7. Bubble with a lens-shaped interface after re-initialization.

Zalesak's rotational flow computations

Zalesak's rotational flow computation trial is a well-known and famous interface representation trial, as found in many research articles [12–15]. We have not proposed this trial; we just used this trial to validate the proposed re-initialization scheme for the LS field.

Table 2
The re-initialized bubble with a lens-shaped interface convergence with Euclidean-norm and infinity-norm errors.

Mesh length (*h*)	*Euclidean-Norm (units)*	*Convergence Order*	*Infinity-norm Error (units)*	*Convergence Order*
0.01	0.23272	—	0.16164	—
0.005	0.22764	1.02232	0.15223	1.06181
0.0025	0.20298	1.12150	0.14125	1.07773

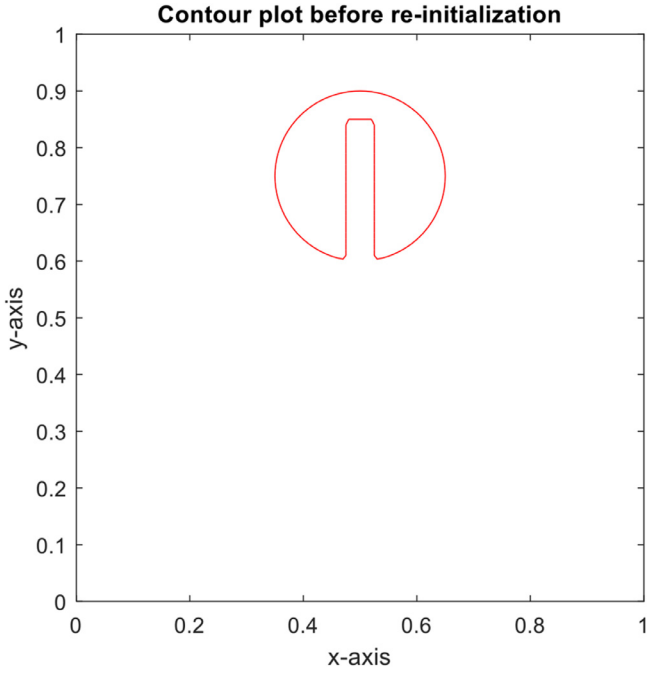

Fig. 8. Zalesak's rotational flow before re-initialization.

In Zalesak's rotational flow computations, we consider the exact computational domain defined in the prior test problem. We express the initial condition subsequently for the LS field as,

$$\phi_I(\mathbf{Y},0) = \max\left\{\phi_1(\mathbf{Y},0),\ -\phi_2(\mathbf{Y},0)\right\} \tag{17}$$

Where,

$$\phi_1(\mathbf{Y},0) = \left|\mathbf{Y} - \mathbf{Y}^c(0)\right| - S \tag{18}$$

$$\phi_2(\mathbf{Y},0) = \max\left(\left|\mathbf{Y}_1 - \mathbf{Y}_1^d(0)\right| - m, \left|\mathbf{Y}_2 - \mathbf{Y}_2^c(0) + 2m\right| - b\right) \tag{19}$$

Here, the computational values are supposed to be $\mathbf{Y}^d(0) = (0.45, 0.65)^T$, $S = 0.20$ and $\phi_2(\mathbf{Y},0)$ constitute the rectangular region with breath $m = S/6$ and length $b = S$, respectively. Consequently, we depict the re-initialization of the Zalesak's rotating disc before and after re-initialization in Fig. 8 and Fig. 9, respectively.

Re-initialized Zalesak's rotational flow computations with norms and errors

Table 3 reveals the re-initialized LS field convergence results for Zalesak's rotational flow computations at different mesh lengths. The table provides the Infinity norm error and the Euclidean norm error together with a convergence order at all mesh lengths, as shown in Table 3. In this trial, the Infinity norm and Euclidean norm errors give the convergence order of 1.7, close to 2 or amid first and second-order convergence rates. The test results demonstrate that the presented re-initialization method for the LS field gives good convergence behaviour for this issue.

Due to the interface problem, owning Zalesak's rotational flow computations is problematic since the interface is solely C° continuous during the Zalesak's rotational flow computations test case; in addition, it demonstrates both concave and convex attributes in each area. Employing numerical approaches to conserve the different edges of the interface also affects their efficacy. This research presents a new LS re-initialization scheme depending on the discontinuous Galerkin finite element method to alleviate these complexities. The devised re-initialization scheme showed superior performances, more accurateness, higher accuracy, enriched stability

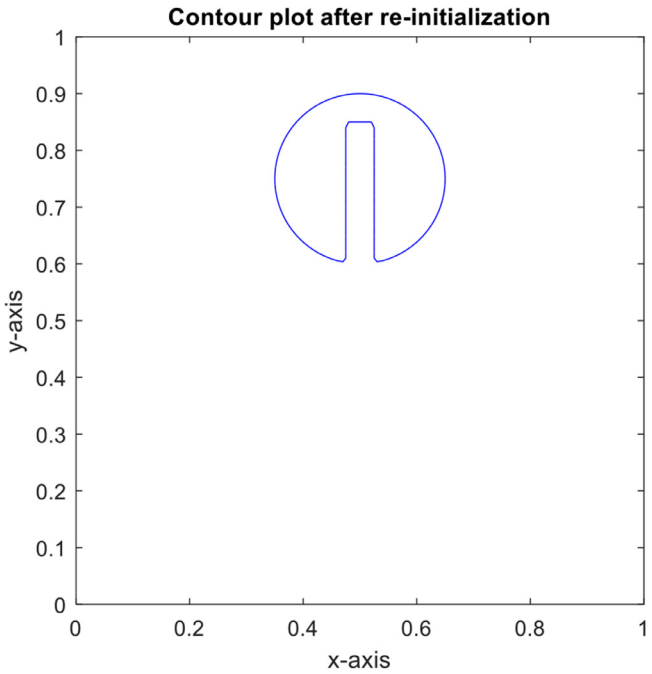

Fig. 9. Zalesak's rotational flow after re-initialization.

Table 3
The re-initialized Zalesak's rotational flow computations achieved convergence with Euclidean-norm and infinity-norm errors.

Mesh length (*h*)	*Euclidean Norm (units)*	*Convergence Order*	Max. *Norm Error (units)*	*Convergence Order*
0.01	0.13975	—	0.10924	—
0.005	0.08495	1.64508	0.06392	1.70901
0.0025	0.04822	1.76172	0.03694	1.73037

and minimized numerical diffusion. The presented scheme is quite influential and delivers good results, even if it requires lesser degrees of freedom that permit efficient parallelization. Moreover, the introduced scheme differs entirely from the other methods due to adaptivity and the discontinuous Galerkin composition. Consequently, the presented re-initialization scheme effectively tackles the complexities of Zalesak's rotational flow computations, providing valid and accurate results.

Conclusion

This paper presents the optimum Level-Set (LS) field re-initialization approach depending on the FEM for unstructured meshes. The re-initialization method is admiringly productive and efficacious for conserving the LS mass. Also, the introduced method depends upon the Eulerian-Lagrange multiplier technique. The introduced approach is based on geometry and conjoined with a finite element method applied to polynomials of higher degrees on unstructured meshes: the benchmark trials exhibited the introduced method's efficiency and mass-conservation of the LS field. The proposed re-initialization scheme provides second-order convergence rate, stability, and good accuracy and is flexibly applied on the unstructured meshes; additionally, it effectively controls significant distortions. In future research directives, the researchers can advance the executed research to the highest polynomial degree, precisely, third degree, fourth degree and so on. Also, the presented method applies to the polygonal and tetrahedral geometries to attain excellent accurateness. The devised re-initialization method is an advantageous tool for the application problems of engineering and computational fluid dynamics. This method also shows strength and adaptability.

Limitations

The Fluid behaviour in this research is examined as ***incompressible flow***, and no experimental work has been conducted.

Credit author statement

Mr. Umer Siddiqui (First author) wrote the manuscript and prepared all diagrams and tables under Dr. Fahim Raees's (Second author) supervision. Both authors have reviewed the manuscript.

Ethics statements

Both authors declare that ethical policies were used to conduct this numerical study, which did not include human participants, experimental animals, or sensitive information. The Level-Set method re-initialization scheme discussed in this paper was created and evaluated through numerical simulations using MATLAB, with the results studied and presented to sustain their integrity and confidentiality. Both authors also declare that they do not have any conflict of interest.

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.

Data availability

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

Acknowledgment

This research was funded by NED. University of Engineering and Technology, Karachi, Pakistan. The author and co-author both are very thankful to NED University and acknowledged for their support.

References

[1] F. Raees, D. Roy van der Heul, C. Vuik, A mass-conserving LS method for simulation of multiphase flow in geometrically complicated domains, Int. J. Num. Meth. Fluids (2015), doi:10.1002/fld.4188.
[2] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. R. Astron. Soc. 181 (1977) 375–389, doi:10.1093/mnras/181.3.375.
[3] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comp. Phys. 39 (1981) 201–225, doi:10.1016/0021-9991(81)90145-5.
[4] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comp. Phys. 79 (1988) 12–49, doi:10.1016/0021-9991(88)90002-2.
[5] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, *Annu* Rev. *Fluid. Mech.* 30 (1998) 329–364, doi:10.1146/annurev.fluid.30.1.329.
[6] W. Rider, D. Kothe, A marker particle method for interface tracking, in: Proceedings of the Sixth International Symposium on Computational Fluid Dynamics, 1995, pp. 976–981.
[7] W. Khan, B. Ullah, Z. &Ullah, The localized radial basis functions for parameterized LS based structural optimization, Engr. Anal. Boundary Elements 113 (2020) 296–305, doi:10.1016/j.enganabound.2020.01.008.
[8] Y. Zhang, Fast approach to checkerboard corner detection for calibration, Opt. Eng. 53 (2014) 112203 -112203, doi:10.1117/1.oe.53.11.112203.
[9] W. Li, J. Qian, Joint inversion of gravity and travel time data using a LS based structural approach, in: SEG 2019 *Workshop: Mathematical Geophysics: Traditional* vs *Learning*, Society of Exploration Geophysicists, Beijing, China, 2019, pp. 5–8, doi:10.1190/iwmg2019_02.1.
[10] R.F. Ausas, E.A. Dari, G.C. Buscaglia, A Mass-Preserving Geometry-Based Re-initialization method for the LS Function, Asociación Argentina de Mecánica Computacional, Mecánica Computacional 27 (2008) 13–32 http://venus.santafe-conicet.gov.ar/ojs/index.php/mc/article/viewFile/1395/1360.
[11] D. Hartmann, M. Meinke, W. Schroder, Differential equation based constrained re-initialization for LS methods, J. Comp Phys. 227 (2008) 6821–6845, doi:10.1016/j.jcp.2009.10.042.
[12] F. Raees, A Mass-Conserving hybrid Interface Capturing Methods For the Geometrically Complicated Domain, TU Delft University of Technology. The Netherlands, 2016 PhD thesis, doi:10.4233/uuid:871b1879-d83c-47cb-9480-1cc3df37e3eb.
[13] L. Ville, L. Silva, T. Coupez, Convected level set method for the numerical simulation of fluid buckling, Int. J. Numer. Methods Fluids. 66 (3) (2011) 324–344, doi:10.1002/fld.2259.
[14] A. Zhang, J. Du, Z. Guo, Q. Wang, S. Xiong, Conservative phase-field method with a parallel and adaptive-mesh-refinement technique for interface tracking, *Phys.* Rev. *E* 100 (2) (2019) 023305, doi:10.1103/PhysRevE.100.023305.
[15] F. Zhang, T. Liu, M. Liu, A high-order maximum-principle-satisfying discontinuous Galerkin method for the Level Set problem, J. Sci. Comput. 87 (2) (2021) 45, doi:10.1007/s10915-021-01459-2.