

# LEVEL SET METHOD FOR SIMULATING THE INTERFACE KINEMATICS: APPLICATION OF A DISCONTINUOUS GALERKIN METHOD

Roozbeh Mousavi<sup>1</sup>, Florian Kummer<sup>2</sup>, Martin Oberlack<sup>2</sup>, Peter F. Pelz<sup>1</sup>

<sup>1</sup>Technische Universität Darmstadt, Institut für Fluidsystemtechnik  
L1|01 465, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany  
roozbeh.mousavi@fst.tu-darmstadt.de

<sup>2</sup>Technische Universität Darmstadt, Fachgebiet für Strömungsdynamik  
L1|01 331, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany

**Keywords:** Discontinuous Galerkin Method, Level Set Method, Re-Initialization Equation

**Abstract.** *A Discontinuous Galerkin (DG) Method was applied for simulating the kinematics of deforming interfaces. The level set method was used as an interface capturing method. The numerical implementations were performed in the context of the in-house code BoSSS developed at the Chair of Fluid Dynamics in Darmstadt University of Technology, see [1, 2]. As the higher-order spectral representation of the variables in the DG method, results in a precise solution to the level set advection equation, the interface kinematics could be accurately simulated without having to solve the level set re-initialization equation. But the solution exhibits an appropriate hp-convergence only if the gradient of the level set function does not have any singularity over the domain of computation. For instance, the signed-distance level set function of a circle has a singular gradient at the center of the circle. As the smoothed Heaviside and Delta functions used in the multiphase flow calculations, are commonly expressed in terms of the level set function, the level set function needs to remain signed distance in order to keep a uniform smoothing width. The signed distance property of the level set function can be recovered by solving the level set re-initialization equation. In order to obtain a solution with a monotonicity preserving behavior, a Godunovs scheme was applied for approximating the Hamiltonian of the re-initialization equation. Moreover, a notable stability improvement was achieved by adding an artificial diffusion along the characteristic lines. The solution exhibits an appropriate hp-convergence and almost no spurious movement of the interface was detected.*

## 1 Introduction

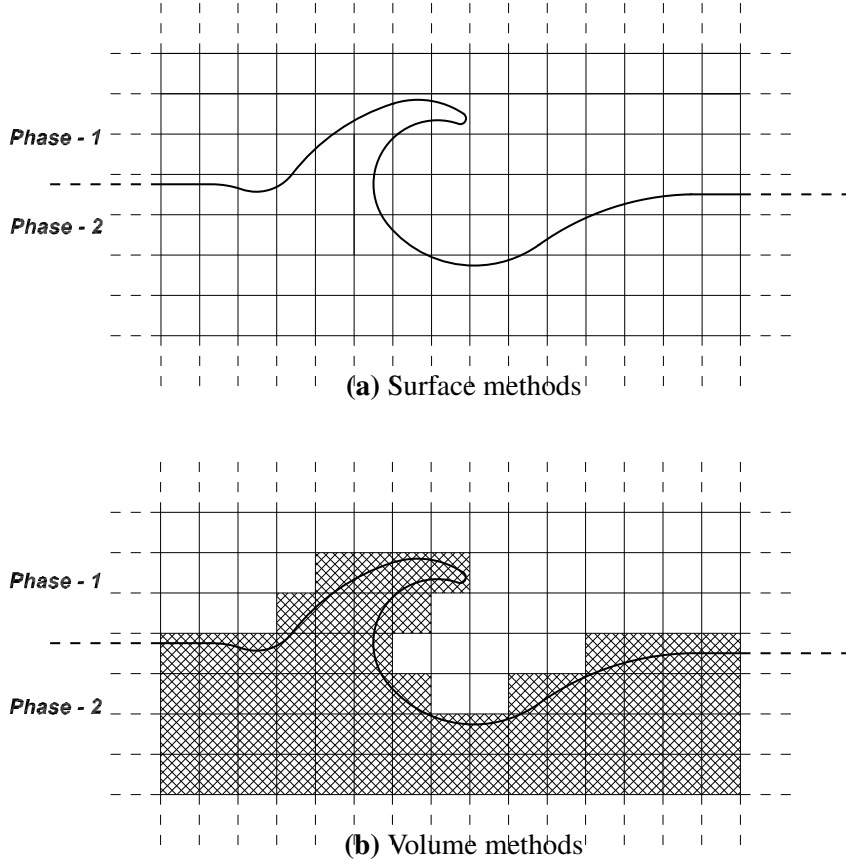
The flows consisting of fluids with different properties, are termed as the multiphase flows. If the fluids are immiscible, they are separated by thin layers known as the interface. Interface is the region across which, the fluid properties as well as some of the flow variables are subjected to steep variations. In the context of the continuum mechanics, an interface is represented as a geometrical surface with zero thickness which may result in mathematical singularities, see [3]. A common approach to overcome the singularity problem is to make a diffuse interface assumption according to which, the fluid properties are supposed to vary smoothly across the interface, see [4].

A major issue in the numerical simulation of the multiphase flows is the manner of representing the interface and simulating its kinematics. Several methods with different levels of the accuracy and robustness have been proposed, among which, the level set method, see [5], is a robust one. The interface in the level set method is represented as the zero iso-value of a function which is called the level set function. The implicit representation of the interface provides an appropriate way for simulating the topological changes. Moreover, as the interface does not need to be reconstructed, this method is quite suitable when a precise calculation of the curvature is required. The accuracy of this method in simulating the interface kinematics is highly dependent on the preciseness of the numerical method applied for solving the corresponding advection equation, namely the level set advection equation. The discontinuous Galerkin (DG) method, see [6], is a modern technique providing a framework where a higher-order approximation can be implemented in a robust way. In this method, the variables are expressed in each cell in terms of an orthonormal basis polynomial space (OBPS). As in this method the in-cell variations are considered, an acceptable level of accuracy can be attained using a rather low spatial resolution. In this way, the total number of the numerical degrees of freedom is reduced although it is increased in each cell. The research presented in [7] is a well-known pioneer study on applying the DG method for solving the level set advection equation. As a consequence of the excellent accuracy they achieved, they claimed that the DG method is the best technique for solving the hyperbolic equations, such as the level set advection equation. But still a lack of Detailed  $hp$ -convergence analysis is notable in the literature. Although in the DG-based level set method the level set function does not need to be signed distance, this property is required for making a uniform diffusion thickness when a diffuse interface assumption is made, see [8, 9]. The signed distance property of a level set function can be maintained by solving an Eikonal equation termed as the level set re-initialization equation, see [10]. Application of the DG method for solving the re-initialization equation has been only considered in [9] yet. Although they have employed a set of the stabilization techniques successfully, a lack of performing a procedural error analysis is obvious in their publications. The present research is mainly motivated by the need of performing the mentioned lacks of the procedural error analyses concerning the solutions to the level set advection and re-initialization equations.

## 2 An Overview on the Numerical Techniques for Simulating the Interface Kinematics

The methods proposed for the numerical representation of the interface, can be classified into two categories, namely the surface methods and the volume methods, see [11]. The surface methods consider the interface itself as an object, either explicitly or implicitly. Whereas, the volume methods consider the phases at the either sides of the interface. Therefore, in the volume methods in order to obtain the normal vector to the interface, the interface needs to be reconstructed. As the interface curvature which is used to predict the surface tension effects, is

the divergence of the normal vector, any inaccuracy in calculating the normal vector intensively appears in the curvature. A common consequence of a non-precise prediction of the surface tension effects is the formation of a set of spurious vortical flows, see [12]. Hence, a surface method is often recommended when the surface tension effects is involved in the problem. The difference between these two types of methods is schematically demonstrated in figures 1a and 1b. The front tracking method introduced in [13], as well as the level set method introduced



**Figure 1** A schematic demonstration of the difference between the surface and volume methods for the numerical representation of an interface

in [14], are the common methods which can be classified under the category of the surface methods. The front tracking method explicitly represents the interface by a set of the connected massless particles. The particles are advected through the domain in a Lagrangian way together with a set of the conditions enforced on the interface. This method is very accurate if the interface is reconstructed by passing a spline of higher-degree over the particles. But the necessity of tracking a rather large number of particles makes this method expensive. Moreover, as the particles need to keep an optimum distance to each other, several particles are necessary to be added or removed during the simulation. In addition, a new reconnection procedure needs to be performed after any change in the configuration of the particles. Furthermore, for the simulation of the interface breakup or coalescence an ad-hoc procedure needs to be performed, see [15]. The level set method which is the subject of the present paper, was shortly introduced in section 1 and will be explained in detail subsequently. Although this method is highly robust, but there is no guaranty for the area/volume conservation if the numerical method applied to solve the

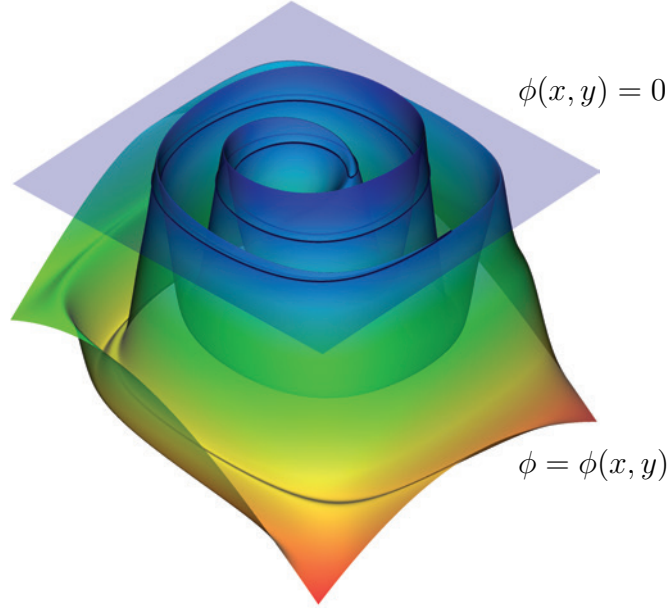
level set advection equation is not precise enough, see [5]. The particle level set method which was introduced in [16], implements a different way for maintaining the area/volume conservation. Each side of the interface in this method is assigned to a distinguished set of the massless particles which can be advected through the domain in a Lagrangian way. As the particles preserve the material characteristics in time, they can be used to reconstruct the interface in the regions where an area/volume loss (or gain) occurs.

The marker and cell method introduced in [17], as well as the volume of fluid method introduced in [18], are the common methods which can be classified under the category of the volume methods. The marker and cell method represents the phases on the either sides of the interface by a set of the massless particles which are advected through the domain in a Lagrangian way. The interface is then reconstructed in the multi-phase cells using the distribution density of the particles. This method is computationally very expensive due to requiring a large number of the particles. Moreover, needing to add additional particles for making an accurate simulation of the interface stretch, is an issue which reduces the robustness of the method. In the volume of fluid method, an indicator function is assigned to the phases at the either sides of the interface. The indicator function is commonly the volume fraction or the mass fraction of one of the phases. Therefore, it has a Heaviside distribution over the domain. The interface kinematics is simulated in this method by solving an advection equation for the indicator function. In the case of using a lower-order spatial discretization method, the Heaviside distribution of the indicator function is numerically smeared out. The region over which, this incorrect interface diffusion takes place, can be even developed by the velocity gradient in the direction normal to the interface. On the other hand, a higher-order numerical representation of the Heaviside distribution can lead to a numerical instability. Therefore, a lower-order method is used together with applying an interface reconstruction technique in order to prevent the development of the diffusion region. The common interface reconstruction techniques include reconstructing an interface of degree zero introduced in [18], and reconstructing an interface of degree 1 introduced in [19]. Another common technique introduced in [20], is to compress the interface by adding a compression term to the advection equation. Although the implementation of the volume of fluid method is rather straightforward, the interface reconstruction techniques always reduce the accuracy of the curvature calculation. But the main advantage of this method is that the area/volume conservation is fulfilled. In [21], they used this property of the volume of fluid method, together with the ability of the level set method in a precise calculation of the curvature, for developing the idea of the coupled level set and volume of fluid method. The interface kinematics in this method is simulated by solving the volume of fluid advection equation. The normal vector which is used for a piecewise linear interface reconstruction, is obtained using the level set function. The updated interface is then used to re-initialize the level set function in order to maintain its signed-distance property.

Comparing the mentioned methods in terms of the accuracy, robustness and ease of the implementation, one can conclude that the classical level set method is an appropriate choice when a solver for a higher-order approximation of the solution to the level set advection equation is available, for instance, the code BoSSS where the DG method is applied.

### 3 Level Set Method for Modeling the Moving Interfaces

The interface in the level set method is represented implicitly as the zero-iso value of a function which is known as the level set function (Figure 2). The implicit representation provides the ability of handling any topological changes of the interface. As it is shown in the picture, although the level set function is defined over whole the domain, only its zero iso-value, sepa-



**Figure 2** Representing an interface as the zero iso-value of a level set function

rating the negative and positive regions of the function, is used for the interface representation. Therefore, any function that its zero iso-value represents the interface, can be used as the level set function. But the level set function is commonly designed to be a function that returns the signed-distance to the interface. The signed-distance level set function  $\phi(\mathbf{x}, t)$  of an interface  $\mathcal{I}$  is defined as,

$$\phi(\mathbf{x}, t) = \begin{cases} -d(\mathbf{x}, t), & \phi < 0, \\ 0, & \phi = 0, \\ d(\mathbf{x}, t), & \phi > 0, \end{cases} \quad (1)$$

where  $d(\mathbf{x}, t)$  is the distance to the interface which is defined as,

$$d(\mathbf{x}, t) = \min(|\mathbf{x} - \mathbf{x}_{\mathcal{I}}(t)|), \quad (2)$$

where  $\mathbf{x}_{\mathcal{I}}(t)$  denotes the interface position. Parameterizing the interface with the surface coordinates  $(\xi_{\mathcal{I}}, \eta_{\mathcal{I}})$ , the interface position can be determined by,

$$\mathbf{x}_{\mathcal{I}}(t) = \mathbf{x}(\xi_{\mathcal{I}}, \eta_{\mathcal{I}}, t). \quad (3)$$

The value of the gradient of a signed-distance level set function is uniformly equal to 1. The advantage of using a signed-distance level set function is that this level set function can be used for constructing the distributions for which the distance to the interface is a parameter. For instance, a smoothed Heaviside function defined as,

$$\mathcal{H}_{\epsilon}(\phi) = \begin{cases} 0, & \phi < -\epsilon, \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right), & -\epsilon \leq \phi \leq \epsilon \\ 1, & \phi > \epsilon, \end{cases} \quad (4)$$

### 3.1 Level Set Advection Equation

Following [10], the evolution of an interface  $\mathcal{I}$  can be determined by,

$$\frac{d\mathbf{x}_{\mathcal{I}}(t)}{dt} = \frac{\partial \mathbf{x}(\xi_{\mathcal{I}}, \eta_{\mathcal{I}}, t)}{\partial t} = \mathbf{u}_{\mathcal{I}}(\mathbf{x}(\xi_{\mathcal{I}}, \eta_{\mathcal{I}}, t), t), \quad (5)$$

where  $\mathbf{u}_\mathcal{I}$  denotes the interface velocity. As  $\phi(\mathbf{x}(\xi_\mathcal{I}, \eta_\mathcal{I}, t), t)$  is defined to be zero for all the time, one can write,

$$\begin{aligned} \frac{d\phi(\mathbf{x}(\xi_\mathcal{I}, \eta_\mathcal{I}, t), t)}{dt} &= \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial x_1} \frac{\partial x_1(\xi_\mathcal{I}, \eta_\mathcal{I}, t)}{\partial t} + \frac{\partial\phi}{\partial x_2} \frac{\partial x_2(\xi_\mathcal{I}, \eta_\mathcal{I}, t)}{\partial t} + \frac{\partial\phi}{\partial x_3} \frac{\partial x_3(\xi_\mathcal{I}, \eta_\mathcal{I}, t)}{\partial t} \\ &= \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial x_{1\mathcal{I}}} u_{1\mathcal{I}} + \frac{\partial\phi}{\partial x_{2\mathcal{I}}} u_{2\mathcal{I}} + \frac{\partial\phi}{\partial x_{3\mathcal{I}}} u_{3\mathcal{I}} = 0, \end{aligned} \quad (6)$$

where  $x_i$  represents the components of  $\mathbf{x}$ , and  $u_{i\mathcal{I}}$  represents the components of  $\mathbf{u}_\mathcal{I}$ . Assuming the interface  $\mathcal{I}$  to be a material surface, for which the velocity of the interface is equal to the velocity of the fluid particles located on the interface, one can write the equation describing the advection of the level set function  $\phi$  as,

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

where  $\mathbf{u}$  is the velocity field.

### 3.2 Level Set Re-Initialization Equation

A signed-distance level set function keeps its signed-distance property if and only if the advection field meets a condition as,

$$\nabla u_n \cdot \nabla\phi = 0, \quad (8)$$

where  $u_n$  is the component of the velocity field in the direction normal to the level set function, obtained as,

$$u_n = \mathbf{u} \cdot \nabla\phi. \quad (9)$$

According to this condition, a signed-distance level set function remains signed-distance if  $u_n$  does not have any spatial variation in the direction normal to the level set function, see [22].

If the advection field does not meet the condition (8), the signed distance property can be recovered by performing a re-initialization procedure that of course must not move the interface. It means that the re-initialization procedure is supposed to affect the level set function except its zero iso-value. The level set function can be re-initialized either by performing a geometrical technique such as the fast marching method introduced in [23], or operating the re-initialization over the field of level set function. The latter approach which was introduced in [24], is followed in the present research.

In order to derive the re-initialization equation instructively, one can start with considering the hyperbolic equation (10) which describes the motion of the iso-values of a level set function in their normal directions,

$$\frac{\partial\phi}{\partial t} + (u_n \mathbf{N}) \cdot \nabla\phi = 0, \quad (10)$$

where  $\mathbf{N}$  represents the normal vector to each of the iso-values, see [5]. Since,

$$\mathbf{N} \cdot \nabla\phi = \frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla\phi = \frac{|\nabla\phi|^2}{|\nabla\phi|} = |\nabla\phi|, \quad (11)$$

equation (10) can be rewritten as,

$$\frac{\partial\phi}{\partial t} + u_n |\nabla\phi| = 0. \quad (12)$$

By solving the equation (12) in a time interval  $\Delta t$ , the local value of  $\phi$  increases by  $(\Delta t)u_n$  times the value of its local gradient. In order to solve an Eikonal equation of the form  $|\nabla\phi| = 1$ , one can follow a pseudo-time stepping approach and solve an equation of the form of the equation (12) with  $u_n = 1$  and an additional source term 1 as,

$$\frac{\partial\phi}{\partial\tau} + |\nabla\phi| = 1, \quad (13)$$

where  $\tau$  is a pseudo-time, see [25]. By solving the equation (13) in a pseudo-time interval  $\Delta\tau$ , the local value of  $\phi$  increases by the difference of the value of its local gradient and 1. It should be noted that the level set function has the negative sign on the opposite side of the interface. Therefore, equation (13) takes the following form in the region  $\phi < 0$ ,

$$\frac{\partial\phi}{\partial\tau} - |\nabla\phi| = -1, \quad (14)$$

The equations (13) and (14) together with the condition that the interface should not be affected by the re-initialization, can be then combined into the following compact form proposed in [10],

$$\frac{\partial\phi}{\partial\tau} + \text{Sign}(\phi^0)(|\nabla\phi| - 1) = 0, \quad (15)$$

$$\phi(\mathbf{x}, 0) = \phi^0, \quad (16)$$

where  $\text{Sign}(\phi^0)$  is a Signum function which is defined as,

$$\text{Sign}(\phi^0) = \begin{cases} -1 & \phi^0 < 0, \\ 0 & \phi^0 = 0, \\ 1 & \phi^0 > 0 \end{cases} \quad (17)$$

In [26], the re-initialization equation is rewritten in a more illuminating form as,

$$\frac{\partial\phi}{\partial\tau} + \mathbf{w} \cdot \nabla\phi = \text{Sign}(\phi^0), \quad (18)$$

where  $\mathbf{w}$  is the characteristic velocity of the hyperbolic equation (18) and defined as,

$$\mathbf{w} = \text{Sign}(\phi^0) \frac{\nabla\phi}{|\nabla\phi|}. \quad (19)$$

Inclusion of the Signum function in the definition of the characteristic velocity implies that the vector  $\mathbf{w}$  points always outward the interface either within the region of  $\phi^0 < 0$  or within the region of  $\phi^0 > 0$ . It means that the re-initialization of the level set function is started from the interface.

As it was mentioned before, the numerical representation of a jump may produce spurious spatial oscillations leading to a numerical instability. Accordingly, for solving the re-initialization equation (15), one needs to use a smoothed Signum function as an approximation to the exact Signum function (17). Although in the literature, see [24], commonly an infinitely smoothed Signum function is used as,

$$\text{Sign}_{\epsilon \rightarrow \infty}(\phi^0) = \frac{\phi^0}{\sqrt{(\phi^0)^2 + \epsilon^2}} \quad (20)$$

in the present research we used a finitely smoothed Signum function as,

$$Sign_{\epsilon}(\phi^0) = \begin{cases} -1 & \phi^0 < -\epsilon, \\ \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi^0}{\epsilon}\right) & -\epsilon \leq \phi^0 \leq \epsilon, \\ 1 & \epsilon < \phi^0, \end{cases} \quad (21)$$

in order to directly adjust the smoothing width. If the slope of the level set function is less or much less than 1, the smoothing width of the smoothed Signum function increases and consequently the speed of the characteristic lines is reduced. But if the slope is much higher than 1, the smoothed Signum function becomes too steep that may result in the numerical instability. In order to overcome this problem, it is proposed in [27] that an infinitely smoothed formulation in terms of the updated level set function  $\phi$  instead of the initial level set function  $\phi^0$  is used as,

$$Sign_{\epsilon \rightarrow \infty}(\phi) = \frac{\phi}{\sqrt{(\phi)^2 + (|\nabla\phi|\epsilon)^2}}. \quad (22)$$

Multiplying  $\epsilon$  by  $|\nabla\phi|$  in the formulation (22), modifies the smoothing width in order to prevent the numerical instability. As it was mentioned before, in the present research we prefer to use a smoothed Signum function with a finite width. Accordingly, the following formulation is constructed,

$$Sign_{\epsilon}(\phi) = \begin{cases} -1 & \phi < -\alpha\epsilon, \\ \frac{\phi}{\alpha\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\alpha\epsilon}\right) & -\alpha\epsilon \leq \phi \leq \alpha\epsilon, \\ 1 & \alpha\epsilon < \phi \end{cases} \quad (23)$$

where

$$\alpha = \begin{cases} 1, & |\nabla\phi| \leq 1, \\ |\nabla\phi|, & |\nabla\phi| > 1, \end{cases}$$

#### 4 Higher-Order Numerical Approximation

The numerical procedure of solving a scalar transport equation consists of two consecutive stages including the spatial discretization and temporal integration. The spatial discretization consists of representing the solution over a discrete domain and approximating the spatial differential terms converting the PDE to a system of temporal ODEs. The time evolution of the solution can be then obtained by performing a time integration. The present research is focused on the higher-order approximation to the spatial variations of the solution. Generally speaking, increasing the grid resolution is an essential way for improving the solution accuracy. However, the rate of convergence is in a direct relation to the order of the spatial discretization. In the case of applying a lower-order scheme, increasing the grid resolution leads to an error reduction that is relatively small comparing to the additional computational cost, see [28]. An instructive interpretation to this behavior can be made in the context of the Fourier analysis, see [29]: A higher-order spatial discretization, in principle gives the ability of resolving the modes of the solution which have higher wave numbers. Therefore, increasing the grid resolution results in an error reduction with a higher rate. Another consequence of using the higher-order schemes is reducing the numerical dissipation and dispersion errors, see [30]. These errors are quite determinant in making accurate solutions to the hyperbolic conservation laws. For instance,



considering a one-dimensional wave equation as,

$$\frac{\partial \phi}{\partial t} + a \frac{\partial \phi}{\partial x} = 0, \quad (24)$$

with  $a$  as the wave speed, the dissipation error reduces the amplitude of the wave leading to the dissipation of the wave, and the dispersion error affects the speed of the wave and produces spurious oscillations. Therefore, if the solution to the equation (24) has high dissipation and dispersion errors, it results in simulating a dissipative wave moving with a wrong speed. Hence, a major advantage that a numerical method can have, is providing a context in which, a higher-order spatial discretization can be formulated properly.

#### 4.1 Spatial Discretization

The spatial discretization methods are principally classified into three categories including the Finite Difference Method, the Finite Volume Method and the family of the spectral methods. The starting point in the procedure of a spatial discretization is converting the continuous domain to a discrete domain, over which the numerical solution is represented and the spatial differential terms are approximated.

A discrete domain in the finite difference method consists of a set of the nodes distributed in a structured way. Whereas in the Finite Volume Method as well as the spectral methods, the discrete domain is composed of a set of the sub-domains (cells) with arbitrary geometries which can be distributed in an unstructured way.

The solution in the Finite Difference Method is represented by its nodal values. Each nodal value together with the neighbour nodal values within a certain stencil, are associated with a polynomial distribution of a certain degree. Therefore, in the context of the Finite Difference Method, one can achieve a higher-order approximation by using a polynomial distributions of higher-degree over an expanded grid stencil. But having to expand the grid stencil in order to make a higher-order approximation, can be consider as a disadvantage of the Finite Difference Method.

The solution in the Finite Volume Method is represented in each cell uniformly as the cell-averaged value or with a linear variation limiting the order of approximation up to 2. In this method, a PDE is converted to a system of temporal ODEs by integrating the PDE locally over each cell. Although there is a discontinuous variation of the solution at the border of the cells, each pair of the cells are connected via approximating the flux of the solution across the common border. As the numerical flux function is expressed in terms of the cell values, it can be approximated with a higher-order by expanding the stencil of the cells, such as the 3<sup>rd</sup>-order WENO (Weighted Essentially Non-Oscillatory) scheme proposed in [31] which is specifically designed to handle steep variations in the solution.

The solution in the spectral methods is represented as a certain composition of a spectrum of prescribed analytical functions. Therefore one can consider this type of methods as the mathematical spectralizers. For instance, the solution  $\phi(\mathbf{x}, t)$  can be spectralized as,

$$\phi(\mathbf{x}, t) \approx \tilde{\phi}(\mathbf{x}, t) = \sum_{j=1}^{N_{DoF}} \hat{\phi}_j(t) \vartheta_j(\mathbf{x}), \quad (25)$$

where  $\vartheta_j(\mathbf{x})$  represents the set of the analytical functions termed as the basis functions,  $\hat{\phi}_j(t)$  represents the corresponding coefficients, and  $N_{DoF}$  denotes the number of the numerical degrees of freedom within the domain, over which the solution is spectralized. As the basis

functions are prescribed, the solution is obtained by computing the unknown coefficients  $\phi_j(t)$ . It can be shown that the spectral methods yield exponential convergence rate, see [30]. The different spectral methods are distinguished based on the following characteristics:

- The domain, over which the solution is spectralized. In the spectral element methods such as the DG method, the spectralization is performed over each cell separately. Whereas in the the Finite Element Method as well as the Fourier Spectral Method, the spectralization is commonly made globally or over a wider stencil.
- The type of the basis functions and the domain within which, they are defined. In the Finite Element Method as well as the DG methods, the basis functions are defined within each cell. Whereas in the Fourier Spectral Method, they are defined over the whole domain. As stated in [32], the most suitable basis functions for the periodic problems are the trigonometric functions. On the other hand, the orthogonal polynomials are proven to be appropriate for the non-periodic problems.
- The third issue is the technique applied for determining the unknown coefficients  $\hat{\phi}_n(t)$ . For the non-periodic problems, the unknown coefficients are commonly determined applying the method of weighted residuals which is explained subsequently.

#### 4.1.1 Method of Weighted Residuals

Considering the level set advection equation (7), as an approximate solution  $\tilde{\phi}(\mathbf{x}, t)$  does not necessarily satisfy the equation, its substitution into the equation results in the appearance of a residual term as,

$$\frac{\partial \tilde{\phi}}{\partial t} + \nabla \cdot (\tilde{\phi} \mathbf{u}) = \mathcal{R}(\tilde{\phi}). \quad (26)$$

The method of Weighted Residuals is based on seeking an approximate solution that satisfies a certain restriction imposed on the residual function. The restriction is placed by equating the Legendre inner product of the residual function and a test (weight) function to zero as,

$$\langle \chi_j(\mathbf{x}), \mathcal{R}(\mathbf{x}) \rangle = \int_{\Omega} \chi_j(\mathbf{x}) \mathcal{R}(\mathbf{x}) d\mathcal{V} = 0, \quad j = 1, \dots, N_{DoF}. \quad (27)$$

Consequently, the equation (26) takes a form as,

$$\int_{\Omega} \chi_j(\mathbf{x}) \left[ \frac{\partial \tilde{\phi}}{\partial t} + \nabla \cdot (\tilde{\phi} \mathbf{u}) \right] d\mathbf{x} = \int_{\Omega} \chi_j(\mathbf{x}) \mathcal{R}(\tilde{\phi}) d\mathcal{V}, \quad (28)$$

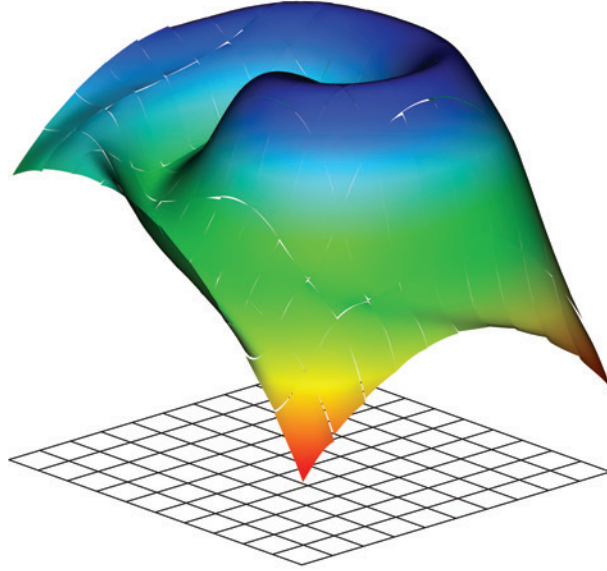
$$j = 1, \dots, N_{DoF},$$

which is in fact a set of  $N_{DoF}$  equations. The different Weighted Residuals methods are distinguished based on the test function they employ, see [30]. Among the different methods, the Galerkin Weighted Residual method uses the same set of the functions as the test functions, which is used as the basis functions. The Finite Element Method as well as the DG method follow the approach of the Galerkin Weighted Residual method.

## 4.2 Discontinuous Galerkin Method

### 4.2.1 Solution Representation

The DG method is classified in the category of the spectral element methods. The cell-wise representation of the solution results in the solution discontinuity across the borders of the cells, as it is shown in the figure 3. The amplitude of the discontinuities are adjusted by the grid resolution as well as the degree of the spectral representation. The procedure of applying the



**Figure 3** A DG-based representation of a solution with observed cell-boundary discontinuities

DG method starts with approximating the physical domain  $\Omega$  bounded by  $\partial\Omega$  by a discrete domain  $\Omega_h$  bounded by  $\partial\Omega_h$  consisting of  $N_C$  non-overlapping boundary conforming cells  $\Omega_k^h$ . The DG field  $\phi_{hp,k}(\mathbf{x}, t)$  which is defined as the DG-based representation of a solution  $\phi(\mathbf{x}, t)$ , can be constructed over each cell  $\Omega_k^h$  employing an orthogonal basis polynomials space  $\vartheta_j(\mathbf{x})$  as,

$$\phi(\mathbf{x}, t)|_{\Omega_k^h} \approx \phi_{hp,k}|_{\Omega_k^h}(\mathbf{x}, t) = \sum_{j=1}^{N_P} \hat{\phi}(t)_{jk} \vartheta_{jk}(\mathbf{x}), \quad k = 1, \dots, N_C, \quad (29)$$

where  $N_P$  denotes the dimension of the orthogonal basis polynomials space. The letter  $h$  in the subscript  $hp$ , indicates the finite spatial resolution, and the letter  $p$  signifies the spectral representation. The dimension of the orthogonal basis polynomials space which is required for constructing a DG field of degree  $p$ , can be calculated as,

$$N_P = \frac{1}{D!} \prod_{1 \leq l \leq D} (p + l), \quad (30)$$

where  $D$  denotes the spatial dimension. It is more convenient to normalize the orthogonal basis polynomials space by the  $\mathbb{L}^2$  norm of each of the polynomials, resulting an orthonormal basis polynomial space. The orthonormal basis polynomials space used in the present research is constructed over a set of  $N_P$  monomials applying the Gram-Schmidt algorithm. For instance

following [37], the monomials space  $\mathcal{P} := \{1, x, y, x^2, y^2, xy\}$  can be converted to an orthonormal polynomial space  $\mathbb{P}$  as,

$$\Theta := \left\{ \frac{1}{2}, \frac{\sqrt{3}}{2}x, \frac{\sqrt{3}}{2}y, \sqrt{5} \left( \frac{3}{4}x^2 - \frac{1}{4} \right), \frac{3}{2}xy, \sqrt{5} \left( \frac{3}{4}y^2 - \frac{1}{4} \right) \right\}$$

In order to obtain the coefficients  $\hat{\phi}(t)$  corresponding the DG-based representation of a solution  $\phi(\mathbf{x}, t)$ , the solution needs to be projected over the orthogonal basis polynomials space  $\vartheta_{jk}(\mathbf{x})$  by performing an inner product as,

$$\begin{aligned} \hat{\phi}(t)_{jk} &= \langle \phi(\mathbf{x}, t), \vartheta_{jk}(\mathbf{x}) \rangle = \int_{\Omega_k^h} \phi(\mathbf{x}, t) \vartheta_{jk}(\mathbf{x}) d\mathcal{V}, \\ j &= 1, \dots, N_P, \\ k &= 1, \dots, N_C, \end{aligned} \quad (31)$$

The procedure of solving an equation in the modal DG method, which is used in the present study, is finalized by obtaining the coefficients  $\hat{\phi}(t)$ . Then the solution values can be calculated at any arbitrary point within the domain in the postprocessing stage.

As a result of the solution discontinuity across the borders of the cells, every point located on a cell border corresponds to a pair of the asymptotic values, namely the inner-cell value and the outer-cell value, denoted by  $\phi_{hp}^-(\mathbf{x}^k)$  and  $\phi_{hp}^+(\mathbf{x}^k)$ , respectively. A conceptual representation of these values is given in figure 4 for a 1D problem. As it is indicated in this figure, the solution discontinuity is quantified by the jump operator  $[[\phi_{hp}]]_{x^k}$  which is defined as the difference between the inner- and the outer-cell values. The inner- and the outer-cell values are defined on the border  $\partial\Omega_{h,k}$  of a cell  $\Omega_{h,k}$  as,

$$\phi_{hp}^-(\mathbf{x}^k) := \lim_{\boldsymbol{\xi}^k \rightarrow \mathbf{x}^k} \phi_{hp}(\boldsymbol{\xi}^k), \quad \mathbf{x}^k \in (\Omega_k^h \setminus \partial\Omega_k^h), \quad (32)$$

$$\phi_{hp}^+(\mathbf{x}^k) := \lim_{\boldsymbol{\xi}^k \rightarrow \mathbf{x}^k} \phi_{hp}(\boldsymbol{\xi}^k), \quad \mathbf{x}^k \notin \bar{\Omega}_k^h, \quad (33)$$

where

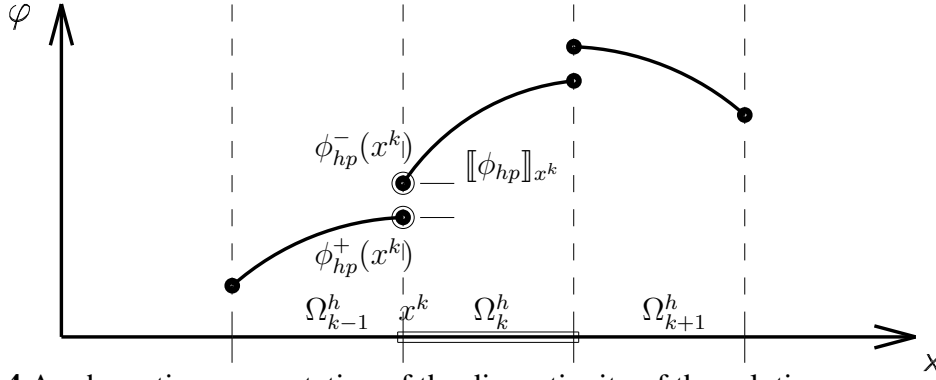
$$\mathbf{x}^k \in (\Omega_k^h \setminus \partial\Omega_k^h) := \{\mathbf{x}^k \in \Omega_k^h : \mathbf{x}^k \notin \partial\Omega_k^h\},$$

and

$$\mathbf{x}^k \notin \bar{\Omega}_k^h \equiv \mathbf{x}^k \notin \left( \Omega_k^h \cup \partial\Omega_k^h \right) := \{\mathbf{x}^k : \mathbf{x}^k \in \Omega_k^h \vee \mathbf{x}^k \in \partial\Omega_k^h\}.$$

#### 4.2.2 Approximation of the Partial Differential Terms

The DG method employs the Galerkin Weighted Residual method for the numerical approximation of the partial differential terms. Considering the level set advection equation (7), this method can be implemented by multiplying the  $N_P$  members of the orthonormal basis polynomial space  $\vartheta_j$  as,



**Figure 4** A schematic representation of the discontinuity of the solution across each of the cell borders

$$\int_{\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \left[ \frac{\partial \phi_{hp}}{\partial t} + \nabla \cdot (\phi_{hp} \mathbf{u}_{hp}) \right] d\mathcal{V} = 0, \quad (34)$$

$$j = 1, \dots, N_P,$$

$$k = 1, \dots, N_C,$$

leading to a matrix equation as,

$$\underbrace{\int_{\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \left( \frac{\partial \phi_{hp}}{\partial t} \right) d\mathcal{V}}_{\mathbf{TM}} + \overbrace{\int_{\Omega_k^h} \vartheta_{jk}(\mathbf{x}) (\nabla \cdot \underbrace{(\phi_{hp} \mathbf{u}_{hp})}_{\text{Non-Linear Flux} := \mathbf{f}}) d\mathcal{V}}^{\mathbf{AM}} = 0, \quad (35)$$

where the matrices are of the size  $1 \times N_P$ . It is more convenient to consider the matrix corresponding to each of the terms separately as **TM** and **AM**, denoting the time and the advection matrices, respectively.

**Time Matrix (TM)** Substituting the DG-based representation of  $\phi$  into this tensor yields,

$$\begin{aligned} \mathbf{TM} &:= \int_{\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \left( \frac{\partial}{\partial t} \left( \sum_{i=1}^{N_P} \hat{\phi}_{ik}(t) \vartheta_{ik}(\mathbf{x}) \right) \right) d\mathcal{V} \\ &= \sum_{i=1}^{N_P} \frac{\partial \hat{\phi}_{ik}(t)}{\partial t} \int_{\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \vartheta_{ik}(\mathbf{x}) d\mathcal{V} \\ &= \sum_{i=1}^{N_P} \frac{d\hat{\phi}_{ik}(t)}{dt} \delta_{ji}. \end{aligned} \quad (36)$$

where  $\delta_{ji}$  is the Kronecker delta function defined as,

$$\delta_{ji} = \begin{cases} 1 & , \text{if } j = i \\ 0 & , j \neq i. \end{cases} \quad (37)$$

**Advection Matrix (AM)** It should be noted that as the variable  $\phi$  is the unknown variable and the velocity field  $\mathbf{u}$  is prescribed, this advection term is not a non-linear term. Taking an integration by parts yields,

$$\mathbf{AM} := \oint_{\partial\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \mathbf{n}_S \cdot \mathbf{f} dS - \int_{\Omega_k^h} \mathbf{f} \cdot \nabla \vartheta_{jk}(\mathbf{x}) d\mathcal{V} \quad (38)$$

where  $\mathbf{n}_S$  is the normal vector to the cell border, directing outward of the cell. In the DG method, although the solution can be discontinuous across the cell borders, each common border of two adjacent cells must be assigned to a unique flux function  $\mathbf{f}^*$  which is called the numerical flux. This explicit specification of the fluxes provides the conservative property of the DG method. The numerical flux is required to be substituted only into the first term of the formulation (38) as this term lives on the cell border where the adjacent cells are in connection. In order to construct a numerical flux at a cell border, the solution needs to be reconstructed there. The solution reconstruction in the DG method is commonly made by applying either a 1<sup>st</sup>-order upwind scheme or a 2<sup>nd</sup>-order central scheme using the inner- and the outer-cell values. An upwind numerical flux can be defined as,

$$\mathbf{f}_U^* = \begin{cases} \phi_{hp}^- \mathbf{u}_{hp}^-, & \frac{\mathbf{u}_{hp}^- \cdot \mathbf{n}_S + \mathbf{u}_{hp}^+ \cdot \mathbf{n}_S}{2} \geq 0, \\ \phi_{hp}^+ \mathbf{u}_{hp}^+, & \frac{\mathbf{u}_{hp}^- \cdot \mathbf{n}_S + \mathbf{u}_{hp}^+ \cdot \mathbf{n}_S}{2} < 0. \end{cases} \quad (39)$$

The upwind scheme is appropriate if the spatial differential term represents a directional phenomenon such as the advection. This is because this scheme is characterized by high dissipation and low dispersion errors. The central scheme is a proper choice for the terms representing the directionless phenomena such as the diffusion. This is because this scheme is characterized by low dissipation and high dispersion errors. Therefore, if it is applied for a directional problems, it does not dissipate the non-resolved modes of the solution and advects them through the domain with wrong speeds in wrong directions, leading to the formation of spurious spatial oscillations, see [38, 30]. The central numerical flux can be defined as,

$$\mathbf{f}_C^* = \frac{\mathbf{u}^- \phi_{hp}^- + \mathbf{u}^+ \phi_{hp}^+}{2} \quad (40)$$

Substituting the numerical flux  $\mathbf{f}^*$  into the first term of the formulation (38), and substituting the DG-based representation of  $\mathbf{f}$  into the second term, yields,

$$\begin{aligned} \mathbf{AM} &:= \oint_{\partial\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \mathbf{n}_S \cdot \mathbf{f}^* dS - \int_{\Omega_k^h} \left( \sum_{i=1}^{N_P} \hat{\mathbf{f}}_{ik}(t) \vartheta_{ik}(\mathbf{x}) \right) \cdot \nabla \vartheta_{jk}(\mathbf{x}) d\mathcal{V} \\ &= \oint_{\partial\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \mathbf{n}_S \cdot \mathbf{f}^* dS - \left( \sum_{i=1}^{N_P} \hat{\mathbf{f}}_{ik}(t) \right) \cdot \int_{\Omega_k^h} \vartheta_{ik}(\mathbf{x}) \nabla \vartheta_{jk}(\mathbf{x}) d\mathcal{V}. \end{aligned} \quad (41)$$

**Source Matrix (SM)** In the case of having an extra source term  $s_\phi$  in the equation, implementing the Galerkin Weighted Residual method results in a source matrix  $\mathbf{SM}$  as,

$$\mathbf{SM} := \int_{\Omega_k^h} \vartheta_{jk}(s_\phi) d\mathcal{V}. \quad (42)$$

Substituting the DG-based representation of  $s_\phi$  as  $s_\phi = \sum_{j=1}^{N_P} \hat{s}(t)_{jk} \vartheta_{jk}(\mathbf{x})$ , yields,

$$\begin{aligned} \mathbf{SM} &:= \int_{\Omega_k^h} \vartheta_{jk}(\mathbf{x}) \left( \sum_{i=1}^{N_P} \hat{s}_{\phi,ik}(t) \vartheta_{ik}(\mathbf{x}) \right) d\mathcal{V} \\ &= \sum_{i=1}^{N_P} \hat{s}_{\phi,ik}(t) \int_{\Omega_k^h} \vartheta_{ik}(\mathbf{x}) \vartheta_{jk}(\mathbf{x}) d\mathcal{V} \\ &= \sum_{i=1}^{N_P} \hat{s}_{\phi,ik}(t) \delta_{ji}. \end{aligned} \quad (43)$$

### 4.2.3 Gradient Calculation

The gradient of a variable  $\phi$  in the DG method can be calculated in two different ways including,

- **Broken gradient:** In this method, considering the DG-based representation of the variable as  $\phi = \sum_{j=1}^{N_P} \hat{\phi}(t)_{jk} \vartheta_{jk}(\mathbf{x})$ , the gradient can be obtained by differentiating the prescribed orthonormal basis polynomial space  $\vartheta_{jk}(\mathbf{x})$ . Therefore, denoting the gradient by  $\mathbf{G} = \nabla \phi$ , its DG-based representation can be obtained as,

$$\sum_{j=1}^{N_P} \hat{\mathbf{G}}_{jk}(t) \vartheta_{jk}(\mathbf{x}) = \sum_{j=1}^{N_P} \hat{\phi}_{jk}(t) \nabla \vartheta_{jk}(\mathbf{x}) \quad (44)$$

- **By-flux gradient:** In this method, each component of the gradient vector is written in the form of a divergence as,

$$G_i = (\nabla_i) \phi \equiv \nabla \cdot (\mathbf{e}_i \phi), \quad i = 1, 2, 3 \quad (45)$$

where  $\mathbf{e}_i$  represents the standard bases. In this way, each component of the gradient term takes the form of an advection term. Therefore, the same procedure described for the numerical approximation of an advection term can be followed, together with using the central scheme for determining the numerical flux. As it is shown in [39], this method leads to a more accurate calculation of the gradient than the former method.

### 4.3 Boundary Condition

As it was explained before, the numerical flux can be determined at the cell borders using a combination of the inner- and the outer-cell values  $\phi_{hp}^+$  and  $\phi_{hp}^-$ . If the cell border is a part of the domain boundary  $\partial\Omega$ , the outer-cell value is an unknown. If a value  $\phi_B$  is imposed on the boundary as a part of the problem definition (Dirichlet boundary condition), the unknown outer-cell value can be determined in two ways, including the direct method as,

$$\phi_{hp}^+ = \phi_B, \quad (46)$$

and the mirror method as,

$$\frac{\phi_{hp}^- + \phi_{hp}^+}{2} = \phi_B. \quad (47)$$

If the value  $\phi_B$  is not specified, the unknown outer-cell value may be allowed to be equal to the inner-cell value, which means a zero gradient of  $\phi$  normal to the domain boundary (Homogeneous Neumann boundary condition).

#### 4.3.1 Numerical integration

In the procedures of implementing the spatial discretization methods as well as in the post-processing stages, one may require the numerical evaluation of definite integrals. The numerical integration methods are commonly referred in the literature as the quadrature rules. Among the different proposed methods, the Gaussian quadrature rules are highly efficient because of the minimal number of evaluations. In principle, a Gaussian quadrature rule consists of evaluating the integrand at a certain set of the integral points (quadrature points) and making a weighted summation of the calculated values, see [40]. For instance, a 1D Gaussian quadrature rule over a domain  $[-1, 1]$  can be written as,

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n \omega_i f(x_i), \quad (48)$$

where  $\omega_i$  represents the prescribed weighting functions and  $x_i$  represents the quadrature points which are distributed in a certain pattern within the domain of integration. The 1D version of a Gaussian quadrature rule can be simply extended to the corresponding multi-dimensional version by making the same distribution of the quadrature points over a multi-dimensional domain, and employing a multi-dimensional weighting function.

If the integrand has a Heaviside distribution, the accuracy of the integration can not be improved by increasing the order of quadrature rule. In this case, the accuracy can be improved by performing a multistage division of each cell and applying a lower-order quadrature rule over each sub-cell. This technique is commonly termed as the Brute Force integration.

#### 4.3.2 Reference Cell

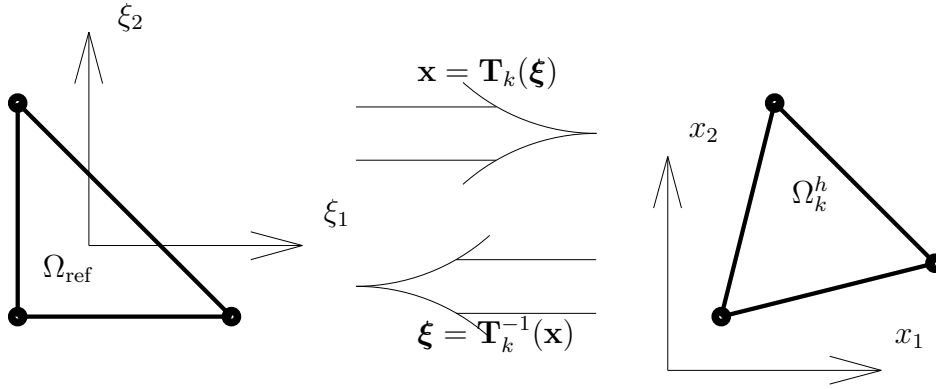
As a numerical grid consists of a set of the cells with various geometries and locations, it is impractical to construct an orthonormal basis polynomial space within each of them. Moreover, implementing a Gaussian quadrature rule for each of the cells is not a clever task. In order to overcome this problem, one can introduce a reference element in another domain characterized by the coordinates  $(\xi_1, \xi_2)$  which are related to the coordinates  $(x_1, x_2)$  using a transformation as,

$$\mathbf{x} = \mathbf{T}_k(\boldsymbol{\xi}) = \mathbf{M}_k \boldsymbol{\xi} + \mathbf{A}_k, \quad k = 1, \dots, N_C \quad (49)$$

where  $\mathbf{M}_k$  is a matrix which accounts for different types of linear deformation such as stretch, rotation and shear, see [1]. The vector  $\mathbf{A}_k$  performs the translation. As the centroid of the reference cell is commonly located on  $(\xi_1 = 0, \xi_2 = 0)$ , the vector  $\mathbf{A}_k$  gives the coordinates of the centroid of the cell  $\Omega_k^h$ . Figure 5 schematically shows a linear transformation between for instance, a triangular reference cell and a triangular cell  $\Omega_k^h$ . According to this transformation, a function  $f(\mathbf{x})$  can be transferred to its corresponding function  $g(\boldsymbol{\xi})$  as,

$$f(\mathbf{T}_k(\boldsymbol{\xi})) = \frac{1}{\sqrt{\det(\mathbf{M}_k)}} g(\boldsymbol{\xi}), \quad k = 1, \dots, N_C \quad (50)$$





**Figure 5** A schematic representation of a linear transformation between the reference cell  $\Omega_{\text{ref}}$  and a cell  $\Omega_k^h$

This transformation can be applied to transfer, for instance, a solution  $\phi_{hp}(\mathbf{x}, t)$  to  $\phi_{ref}(\boldsymbol{\xi}, t)$ , and the orthonormal basis polynomial space  $\vartheta_{jk}(\mathbf{x})$  to  $\vartheta_j(\boldsymbol{\xi})$ , where  $k = 1, \dots, N_C$  and  $j = 1, \dots, N_P$ . It should be noted that the values of the coefficients  $\hat{\phi}_{jk}(t)$  stay the same in the transformation as they are only time dependent functions. Furthermore, an integral over a cell  $\Omega_k^h$  can be transferred to an integral over the reference cell  $\Omega_{ref}$  as,

$$\int_{\Omega_k^h} d\mathcal{V} = \det(\mathbf{M}_k) \int_{\Omega_{ref}} d\mathcal{V}_{ref}, \quad k = 1, \dots, N_C \quad (51)$$

#### 4.4 Time Integration

Considering a temporal ODE as,

$$\phi_t \equiv \frac{d\phi}{dt} = f(\phi, t), \quad (52)$$

the aim of a time integration method is to obtain a sequence of values  $\tilde{\phi}^0, \tilde{\phi}^1, \dots, \tilde{\phi}^n$  such that,

$$\begin{aligned} \tilde{\phi}^n &\approx \phi(t_n), \\ f^n &= f(\tilde{\phi}^n, t). \end{aligned} \quad (53)$$

There are two major families of the numerical time integration methods including the linear multi-step methods and the Runge-Kutta methods, see [33]. The former obtain obtains the  $\tilde{\phi}^{n+1}$  either explicitly using the values of the later time steps, or implicitly using the values of the both later and previous time steps, in multiple steps, each of which includes a single stage. Whereas, the latter obtains it explicitly in a single step from  $\tilde{\phi}^n$  to  $\tilde{\phi}^{n+1}$ , including multiple stages. The multi-stage evaluation makes the Runge-Kutta methods more stable than the linear multi-step methods. However, a possible consequence of using the higher-order schemes is the numerical instability. The Runge-Kutta methods with the total variation diminishing (TVD) property can prevent such a problem, see [34]. Considering the one-dimensional wave equation (24), the total variation of the variable  $\phi$  can be calculated by,

$$TV = \int \left| \frac{\partial \phi}{\partial x} \right| dx. \quad (54)$$

The time integration of equation (24) is TVD if,

$$TV(\phi^{n+1}) \leq TV(\phi^n). \quad (55)$$

It can be proven that a TVD method is monotonicity preserving and also a monotone numerical scheme is TVD, see [35]. A numerical scheme is monotonicity preserving if the following properties are maintained in time, see [36]:

- No creation of any new extremum in the spatial distribution of the solution
- No decreasing values of the local minimums and no increasing values of the local maximums

It can be shown that the 1<sup>st</sup>-order forward (Euler) time integration method is a basic TVD Runge-Kutta method, see [34]. The Euler method can be written as,

$$\tilde{\phi}_t = \frac{\tilde{\phi}^{n+1} - \tilde{\phi}^n}{\Delta t}. \quad (56)$$

Therefore, each stage of a multi-stage higher-order TVD Runge-Kutta method can be constructed by applying an Euler method and combining the results with the initial data using a convex combination, see [5]. The convex combination is a linear combination with the positive coefficients, where the summation of the coefficients is equal to one. For instance following [34], the 3<sup>rd</sup>-order TVD Runge-Kutta method which is applied in the present study, can be constructed as,

$$\begin{aligned} \tilde{\phi}^1 &= \tilde{\phi}^n + \Delta t f^1 \\ \tilde{\phi}^2 &= \frac{3}{4}\tilde{\phi}^n + \frac{1}{4}\tilde{\phi}^1 + \frac{1}{4}\Delta t f^1 \\ \tilde{\phi}^{n+1} &= \frac{1}{3}\tilde{\phi}^n + \frac{2}{3}\tilde{\phi}^2 + \frac{2}{3}\Delta t f^2 \end{aligned} \quad (57)$$

#### 4.5 Solving the Level Set Advection Equation

The spatial discretization of the advection term included in the level set advection equation (7) can be done using the DG method as explained in the section 4.2. The numerical flux is determined using a 1<sup>st</sup>-order upwind scheme. Moreover, a 3<sup>rd</sup>-order TVD Runge-Kutta scheme is used for the time integration as explained in the section 4.4.

#### 4.6 Solving the Level Set Re-Initialization Equation

A stable solution to the re-initialization equation (15) can not be obtained by calculating the components of  $\nabla\phi$  employing a central flux function and considering the term  $Sign(\phi^0)(|\nabla\phi| - 1)$  as a source term. This approach does not necessarily result in a stable solution as it is not monotonicity preserving.

The behavior of the re-initialization equation with respect to a numerical scheme, can be better analyzed when it is considered as a Hamilton-Jacobi equation, see [5]. A general form of the Hamilton-Jacobi equation can be written for the level set function  $\phi(\mathbf{x}, t)$  as,

$$\frac{\partial\phi}{\partial t} + H(\phi_x, \phi_y) = 0, \quad (58)$$

where  $H$  is called the Hamiltonian, and  $\phi_{x_i}$  represents  $\partial\phi/\partial x_i$ . The difficulties in solving the Hamilton-Jacobi equation arise from the fact that this equation develops solutions with singular derivatives even if the initial condition is smooth, see [41]. In order to overcome this problem, a generalized concept of the solution is proposed in [41] which is called the viscosity solution. This name refers to the method of vanishing viscosity which is used to prove the existence of

this type of solutions. The viscosity solution is a weak solution which does not need to be differentiable everywhere, while its existence, uniqueness and stability hold under certain assumptions. Therefore, the solution to every Hamilton-Jacobi equation needs to converge to a viscosity solution. It is proved in [42] that the class of monotone schemes results in the solutions which can meet this condition. Application of a variety of the monotone schemes, such as the Lax-Friedrichs scheme, the Roe-Fix scheme and the Godunov's scheme for solving the Hamilton-Jacobi equations are investigated in [43]. Concerning the re-initialization equation as an Hamilton-Jacobi equation, it is highly recommended in [5] to apply the Godunov's scheme. This approach is followed in the present research accordingly.

#### 4.6.1 Godunov's Scheme

The Godunov's scheme was applied in [44] for solving the Hamilton-Jacobi equation. In order to describe this scheme, we consider the variables  $G_i = \phi_{x_i}$  ( $i = 1, 2$ ) representing the components of  $\nabla\phi$  in 2D. Having the DG-based representation of  $\phi$ , the DG-based representation of each of the variables  $G_i$  can be obtained using the "By-Flux" method as explained before. In order to use the "By-Flux" method, one needs to calculate the numerical fluxes  $f_i^* = (\mathbf{e}_i\phi)^*$  ( $i = 1, 2$ ), either by using an upwind method or a downwind method as,

$$f_i^{*U} = \begin{cases} \phi^- \mathbf{e}_i, & \mathbf{e}_i \cdot \mathbf{n}_s \geq 0 \\ \phi^+ \mathbf{e}_i, & \mathbf{e}_i \cdot \mathbf{n}_s < 0 \end{cases} \quad (59)$$

$$f_i^{*D} = \begin{cases} \phi^- \mathbf{e}_i, & \mathbf{e}_i \cdot \mathbf{n}_s < 0 \\ \phi^+ \mathbf{e}_i, & \mathbf{e}_i \cdot \mathbf{n}_s \geq 0 \end{cases} \quad (60)$$

where ( $i = 1, 2$ ), and  $\phi^-$  and  $\phi^+$  denote the inner- and outer-cell values of  $\phi$  at the border of a cell. In addition,  $\mathbf{e}_i$  represents the standard bases and  $\mathbf{n}_s$  denotes the normal vector to the border of a cell. Following [45], the variables  $G_i^U$  and  $G_i^D$  ( $i = 1, 2$ ), are introduced using the numerical flux  $f_i^{*U}$  and  $f_i^{*D}$  respectively, for calculating  $G_i$ . In the Godunov's scheme, the Hamiltonian is approximated by a numerical Hamiltonian  $\tilde{H}$  as a function of  $G_i^U$  and  $G_i^D$ , expressed as,

$$\begin{aligned} H(G_1, G_2) &\approx \tilde{H}(G_1^U, G_1^D, G_2^U, G_2^D) \\ &= \mathbf{ext}_x \mathbf{ext}_y \mathbf{H}(G_1, G_2), \end{aligned} \quad (61)$$

where

$$\begin{aligned} \mathbf{ext}_x H(G_1, G_2) &= \begin{cases} \min(H(G_1^U, G_2), H(G_1^D, G_2)), & G_1^U \leq G_1^D \\ \max(H(G_1^U, G_2), H(G_1^D, G_2)), & G_1^U > G_1^D \end{cases} \\ \mathbf{ext}_y H(G_1, G_2) &= \begin{cases} \min(H(G_1, G_2^U), H(G_1, G_2^D)), & G_2^U \leq G_2^D \\ \max(H(G_1, G_2^U), H(G_1, G_2^D)), & G_2^U > G_2^D \end{cases} \end{aligned}$$

The numerical Hamiltonian (61) is then considered as a source term in the equation and the equation is solved by performing a time integration.

#### 4.6.2 Godunov's Scheme for Solving the Level Set Re-initialization Equation

The term  $Sign(\phi^0)(|\nabla\phi| - 1)$  in the re-initialization equation corresponds to the Hamiltonian in the Hamilton-Jacobi equation. A compact formulation representing the application of the Godunov's scheme for solving the re-initialization equation is proposed in [27] which can be expressed in 2D as,  $Sign(\phi^0)(|\nabla\phi| - 1)$

$$\approx \begin{cases} Sign(\phi^0)(\sqrt{\max[(G_1^{U+})^2, (G_1^{D-})^2] + \max[(G_2^{U+})^2, (G_2^{D-})^2]} - 1), & Sign(\phi^0) \geq 0 \\ Sign(\phi^0)(\sqrt{\max[(G_1^{U-})^2, (G_1^{D+})^2] + \max[(G_2^{U-})^2, (G_2^{D+})^2]} - 1), & Sign(\phi^0) < 0 \end{cases} \quad (62)$$

where

$$\begin{aligned} G_i^{U+} &= \max(G_i^U, 0), & G_i^{U-} &= \min(G_i^U, 0), \\ G_i^{D+} &= \max(G_i^D, 0), & G_i^{D-} &= \min(G_i^D, 0), \end{aligned} \quad (63)$$

and  $(i = 1, 2)$ . A  $3^{rd}$ -order TVD Runge-Kutta time integration scheme is used for this equation as well.

### 5 Numerical Simulations and Results

This section is assigned to present a set of the test cases considered to investigate the application of the numerical schemes discussed in the previous section for solving the level set advection and re-initialization equations.

#### 5.1 Error Calculation

The numerical results of the test cases considered in the present research are analyzed based on measuring three kinds of the errors including "volume/area loss", "interface  $\mathbb{L}^1$ -error" and " $\mathbb{L}^2$ -error", which are briefly describes as follows:

**Volume/Area Loss** This error gives the difference between the volume/area occupied by the computed interface and the correct volume/area. Supposing that the region occupied by the interface corresponds to the negative part of the level set function, the volume/area of this region can be calculated as,

$$\Omega_{(\phi_{hp} < 0)}^h = \int_{\Omega_{(\phi_{hp} < 0)}^h} d\mathcal{V} \quad (64)$$

Therefore the volume/area loss can be calculated in percentage as,

$$\text{Area Loss} = 100 \times \frac{\Omega_{(\phi^{ref} < 0)} - \Omega_{(\phi_{hp} < 0)}^h}{\Omega_{(\phi^{ref} < 0)}} \quad (65)$$

where  $\phi^{ref}$  denotes a reference level set function which is compared to the calculated field of the level set function.

**Interface  $\mathbb{L}^1$ -Error** This error gives an  $\mathbb{L}^1$  measure on the spurious movement of the interface. This error can be calculated as,

$$\mathbb{L}_{\mathcal{I}}^1 = \frac{1}{\mathcal{L}} \int_{\Omega^h} |\mathcal{H}(\phi_{hp}) - \mathcal{H}(\phi^{ref})| d\mathcal{V} \quad (66)$$

where  $\mathcal{L}$  is the interface circumference,  $\phi^{ref}$  is the reference level set function and  $\mathcal{H}(\phi)$  is the Heaviside function. The interface  $\mathbb{L}^1$ -Error as well as the area loss are calculated in the present research by taking a brute-force integration performing a five-stage cell division.

**$\mathbb{L}^2$  Error** This error gives an  $\mathbb{L}^2$  measure on the accuracy of computing the field of a variable such as the level set function  $\phi$ . Having the reference field of the variable available as an analytic function, the  $\mathbb{L}^2$  error is calculated as,

$$\mathbb{L}_{\phi_{hp}}^2 = \|\phi_{hp} - \phi^{ref}\|_{\Omega^h} = \left( \int_{\Omega^h} \left[ \left( \sum_{k=1}^{N_C} \sum_{j=1}^{N_P} \hat{\phi}_{jk} \vartheta_{jk} \right) - \phi^{ref} \right]^2 d\mathcal{V} \right)^{\frac{1}{2}} \quad (67)$$

where  $\phi^{ref}$  is evaluated at the quadrature points in the procedure of the numerical integration.

## 5.2 Periodic Deformation of a Circle in a Swirl Flow

This section is assigned to verify the numerical solution to the level set advection equation (7) by simulating the periodic deformation of an eccentric circle in a prescribed velocity field corresponding to a time-dependent swirl flow. This test case which was originally considered in [46], is an appropriate benchmark for investigating the dissipative and dispersive errors of a numerical method applied for solving the level set advection equation.

**Problem Description** The domain of computation is a square with the lower-left corner located at  $(0, 0)$  and the upper-right corner located at  $(1, 1)$ . The initial geometry of the interface is a circle with the radius  $R = 0.15$ , centered at  $(x_c = 0.5, y_c = 0.75)$ . The signed-distance level set function of the interface is defined as,

$$\phi^0(\mathbf{x}) = \sqrt{(x - x_c)^2 + (y - y_c)^2} - R, \quad (68)$$

which corresponds to a singular gradient at the center of the circle. In order to investigate the effects of such a singularity, a non-signed-distance level set function is considered additionally, which does not correspond to a singular gradient. This function is expressed as,

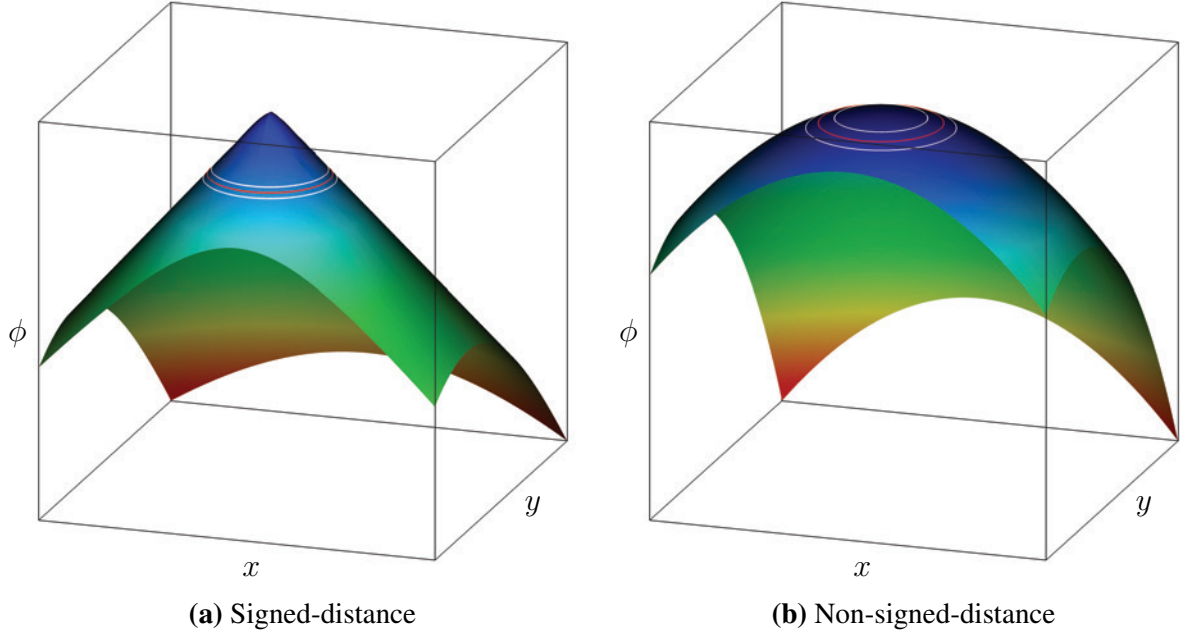
$$\phi^0(\mathbf{x}) = (x - x_c)^2 + (y - y_c)^2 - R^2. \quad (69)$$

Figures 6 illustrate the difference between the initial signed-distance and non-signed-distance level set functions. As it is shown, the non-signed-distance level set function does not have a sharp apex. Therefore, this function can be properly projected to an orthonormal basis polynomial space of a certain degree. Figure 7 illustrates the difference between the broken gradients of the signed-distance and non-signed-distance level set functions after one time step. As it is shown in Figure 7a, the effect of the singularity is not limited to the center of the circle. In order to reduce the size of the singular region, one can either make a grid refinement, or use an orthonormal basis polynomial space of lower-degree.

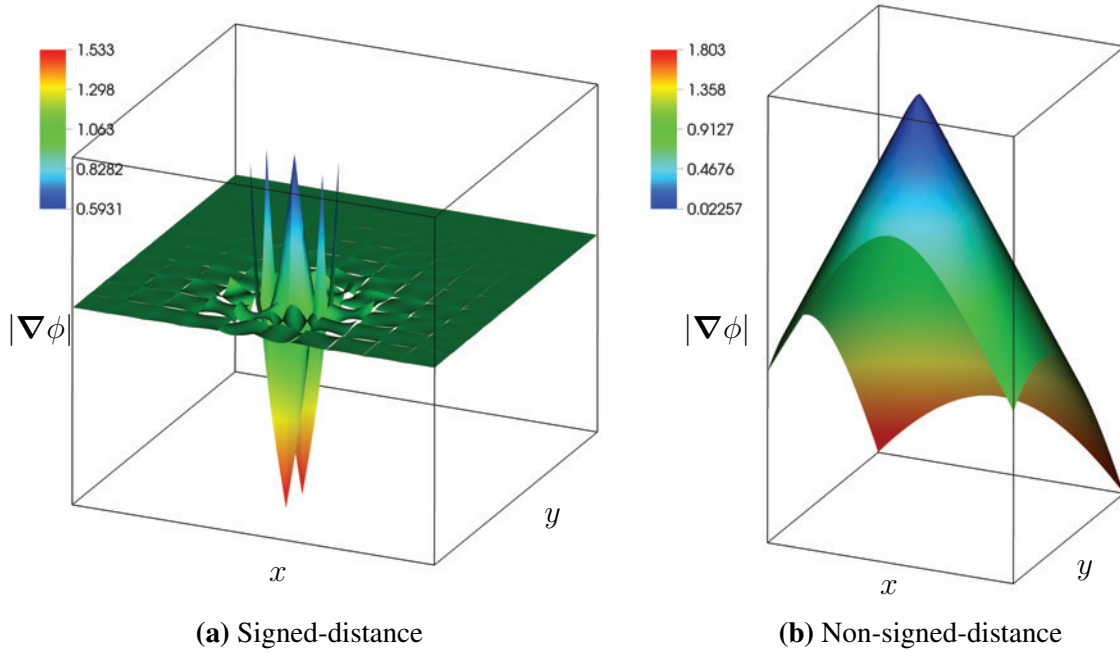
The prescribed velocity field corresponding to the transient swirl flow, is defined as

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= u_x(x, y)\mathbf{e}_x + u_y(x, y)\mathbf{e}_y \\ &= \sin(2\pi y) \sin^2(\pi x) \cos\left(\frac{\pi t}{T}\right) \mathbf{e}_x \\ &\quad - \sin(2\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \mathbf{e}_y \end{aligned} \quad (70)$$

where the term  $\cos(\pi t/T)$  is multiplied for making a time periodicity in the interval  $T$ . The value of  $T$  for this test case is 8.

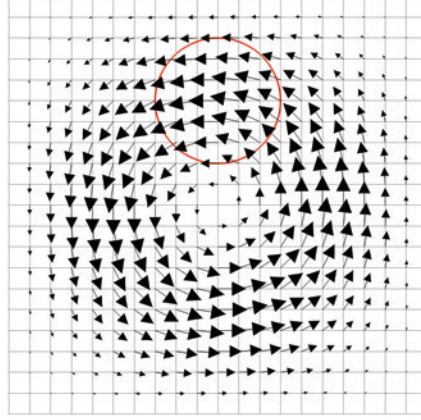


**Figure 6** Initial signed-distance and non-signed-distance level set functions of a circle. The picture corresponds to  $p = 3$  and  $N_C = 10 \times 10$ . The red curve represents  $\phi = 0$  and the white curves represent  $\phi = -0.01$  and  $\phi = 0.01$ .



**Figure 7** Gradients of the signed-distance and non-signed-distance level set functions of a circle after one time step. The picture corresponds to  $p = 3$  and  $N_C = 10 \times 10$ .

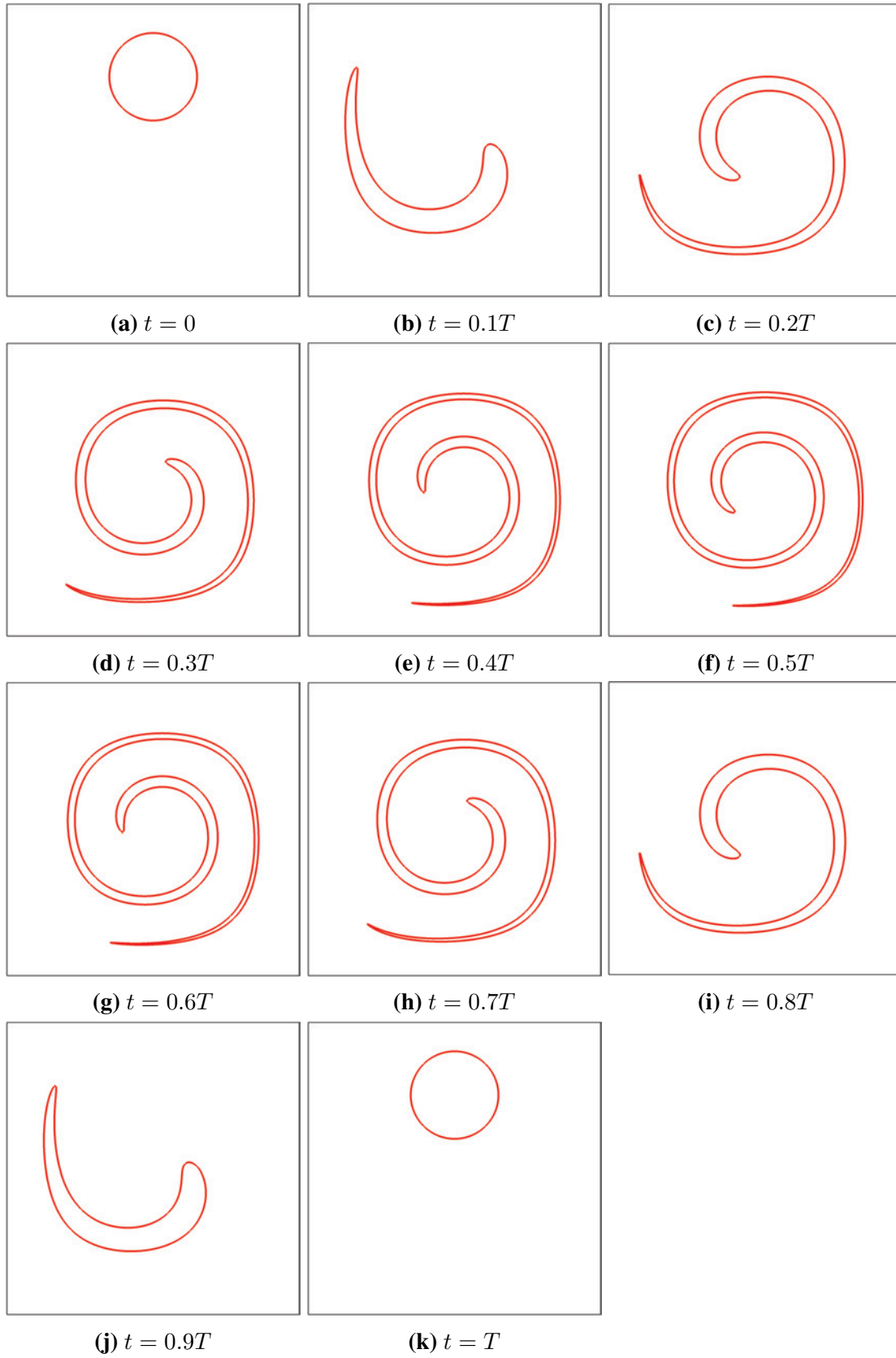
In order to improve the accuracy as well as the numerical stability, the velocity field is updated in each sub-stage of the  $3^{rd}$ -order TVD Runge-Kutta integration which is implemented in the present study. A vector plot of the velocity field is shown in figure 8.



**Figure 8** The prescribed velocity vector field corresponding to a swirl flow

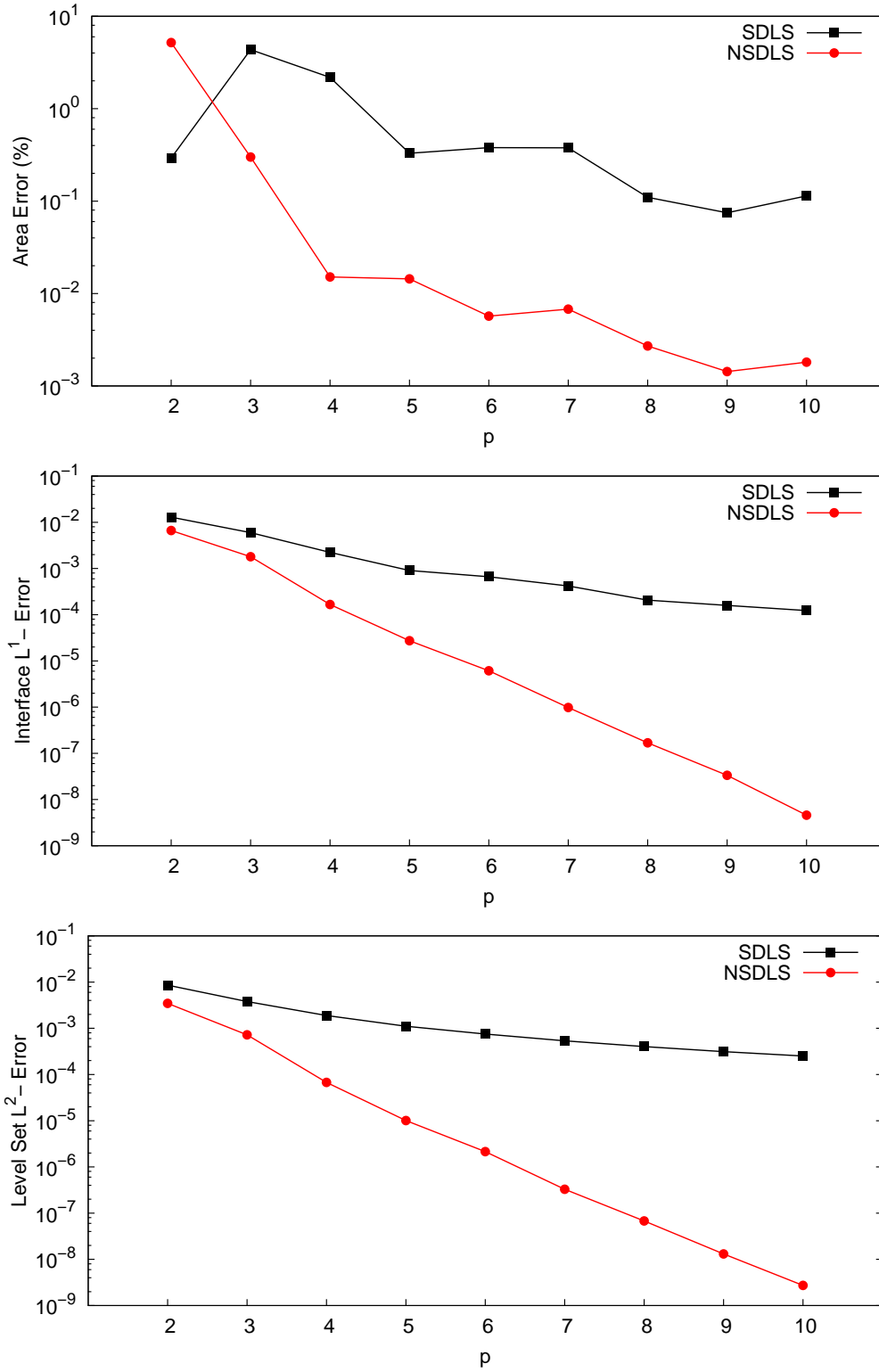
**Numerical Settings** This test case is assigned to perform two classes of the convergence studies including the  $p$ -convergence study and the  $h$ -convergence study, employing both the signed-distance and non-signed-distance level set functions. For the  $p$ -convergence study, the orthonormal basis polynomial space of degrees  $p = 2, 3, 4, 5, 6, 7, 8, 9, 10$  are used and the domain is discretized to a set of the quadrilateral cells with the resolution  $N_C = 40 \times 40$ . For the  $h$ -convergence study, the orthonormal basis polynomial space of degrees  $p = 3, 4, 5, 6, 7$  are used and the domain is discretized to the sets of the quadrilateral cells with the resolutions  $N_C = 10 \times 10, 20 \times 20, 40 \times 40, 80 \times 80$ . The time step for this study is set to 0.000125 corresponding to the CFL number of  $1/p^2$ . In addition to the convergence studies, a case with  $p = 4$  and  $N_C = 32 \times 32$  is considered in order to make a comparison with the available results reported in the literature. A homogeneous Neumann boundary condition is imposed on the entire boundary of the domain.

**Results** Figure 9 shows snapshots of the interface shape captured in one period of the deformation using  $p = 7$ ,  $N_C = 160 \times 160$  and an initially non-signed-distance level set function. The diagrams shown in the Figure 10 illustrate the  $p$ -Convergence studies on the area error, interface  $\mathbb{L}^1$ -error and level set  $\mathbb{L}^2$ -error at  $t = T$ , employing both the initially signed distance (denoted as SDLS) and initially non-signed distance (denoted as NSDLS) level set functions. A major fact illustrated in all of these diagrams is that the singularity dramatically reduces the  $p$ -convergence rate. Furthermore, as the interface  $\mathbb{L}^1$ -error measures a difference between the Heaviside functions but not a difference between the fields of the level set functions, such a large reduction in the  $p$ -convergence rate supports the argument that the negative effect of the singularity is not limited to the point of the singularity even at  $t = T$  when the circle is reformed.



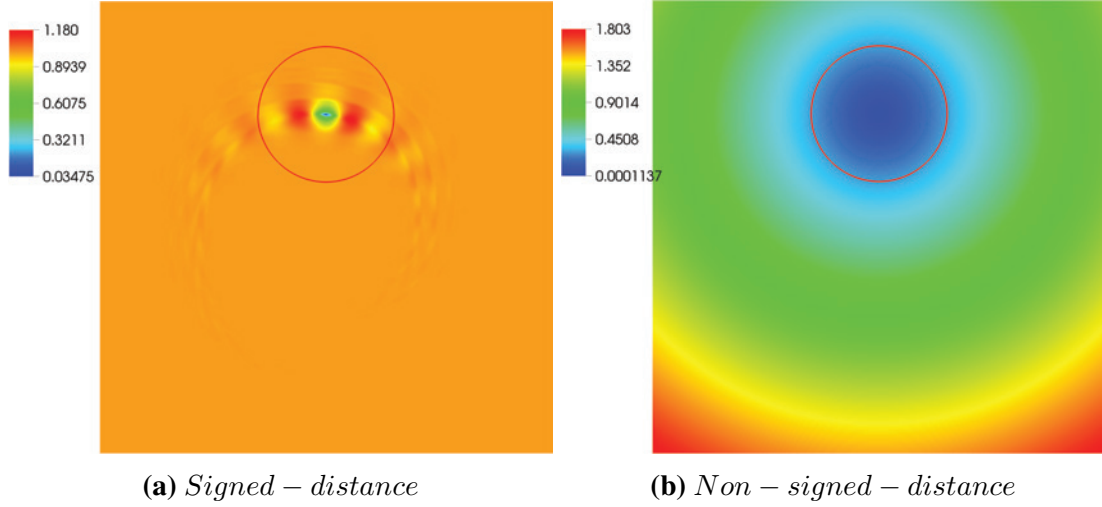
**Figure 9** Periodic deformation of an eccentric circle in a swirl flow. The picture corresponds to  $p = 7$  and  $N_C = 160 \times 160$ . The level set function is initially non-signed-distance.





**Figure 10**  $p$ -Convergence study on the area error, the interface  $L^1$ -error and the level set  $L^2$ -error at  $t = T$  for simulating the periodic deformation of a circle in a swirl flow. The grid resolution is  $N_C = 40 \times 40$ .

Figure 11 shows the difference between the gradients of the initially signed-distance and non-signed distance level set functions after one period of deformation. The left picture indicates the footprint of the singularity over the domain. Table 1 demonstrates an  $h$ -Convergence study



**Figure 11** Gradient of the initially signed-distance and non-signed-distance level set functions of a circle after one period of deformation in a swirl flow. The picture corresponds to  $p = 10$  and  $N_C = 40 \times 40$ .

on the level set  $\mathbb{L}^2$ -error, using the initially signed-distance and non-signed-distance level set functions. This error is theoretically expected to behave as, see e.g. [6],

$$\mathbb{L}_\phi^2 = C(p)(h)^{p+1} \quad (71)$$

where  $h$  denotes the characteristics cell size,  $p$  denotes the degree of the orthonormal basis polynomial space and  $C$  denotes a coefficient which is dependent on  $p$ . According to this expression, increasing the grid resolution is theoretically expected to result in an exponential error reduction of order  $p + 1$ . The error reduction rate or the experimental error order (EEO) can be calculated for each error with respect to the error corresponding to the coarser grid, as

$$\begin{aligned} \mathbb{L}_{\phi,1}^2 &= C(p)(h_1)^O, \\ \mathbb{L}_{\phi,2}^2 &= C(p)(h_2)^O, \\ O &= \frac{\ln(\mathbb{L}_{\phi,2}^2/\mathbb{L}_{\phi,1}^2)}{\ln(h_2/h_1)}, \end{aligned} \quad (72)$$

where  $O$  is the  $h$ -convergence rate or EEO. The third column of the table illustrates that using an initially signed-distance level set function does not results in the expected  $h$ -convergence rate which is  $p + 1$ . Moreover, it is almost the same for all degrees of the orthonormal basis polynomial space. According to the fifth column of the table, employing an initially non-signed-distance level set function results in the expected  $h$ -convergence rate, but not for all of the grid resolutions. It means that using  $N_C = 10 \times 10$  and  $N_C = 20 \times 20$  is not suitable for this certain problem with the corresponding specifications. Diagram 12 is made by plotting the errors listed in table 1 versus the numerical degrees of freedom  $N_{DoF}$ . This diagram illustrates that using an orthonormal basis polynomial space of a higher-degree results in an accuracy which is more than the accuracy achieved by the one with a lower-degree but with the same  $N_{DoF}$ . This

**Table 1**  $h$ -Convergence study on the level set  $\mathbb{L}^2$ -error for simulating a periodic deformation of a circle in a swirl flow

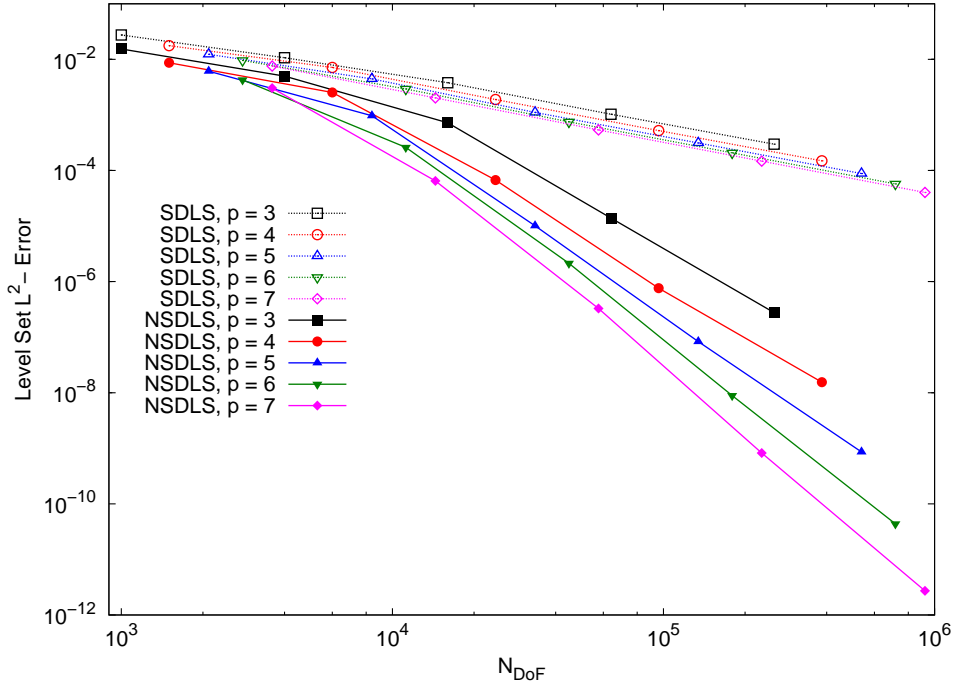
$p$	$N_C$	Signed Distance		Non- Signed Distance	
		Level Set $\mathbb{L}^2$ -Error	EEO	Level Set $\mathbb{L}^2$ -Error	EEO
3	$10 \times 10$	$2.75E - 02$	$\dots$	$1.54E - 02$	$\dots$
	$20 \times 20$	$1.07E - 02$	1.36	$4.99E - 03$	1.63
	$40 \times 40$	$3.80E - 03$	1.50	$7.20E - 04$	2.79
	$80 \times 80$	$1.03E - 03$	1.89	$1.39E - 05$	5.69
	$160 \times 160$	$2.98E - 04$	1.78	$2.76E - 07$	5.66
4	$10 \times 10$	$1.76E - 02$	$\dots$	$8.70E - 03$	$\dots$
	$20 \times 20$	$7.13E - 03$	1.31	$2.54E - 03$	1.78
	$40 \times 40$	$1.89E - 03$	1.92	$6.73E - 05$	5.24
	$80 \times 80$	$5.23E - 04$	1.85	$7.62E - 07$	6.47
	$160 \times 160$	$1.49E - 04$	1.81	$1.55E - 08$	5.62
5	$10 \times 10$	$1.23E - 02$	$\dots$	$6.12E - 03$	$\dots$
	$20 \times 20$	$4.45E - 03$	1.47	$9.64E - 04$	2.67
	$40 \times 40$	$1.11E - 03$	2.00	$1.01E - 05$	6.58
	$80 \times 80$	$3.12E - 04$	1.83	$8.32E - 08$	6.92
	$160 \times 160$	$8.79E - 05$	1.83	$8.60E - 10$	6.60
6	$10 \times 10$	$9.58E - 03$	$\dots$	$4.28E - 03$	$\dots$
	$20 \times 20$	$2.94E - 03$	1.70	$2.61E - 04$	4.04
	$40 \times 40$	$7.52E - 04$	1.97	$2.14E - 06$	6.93
	$80 \times 80$	$2.08E - 04$	1.85	$8.91E - 09$	7.91
	$160 \times 160$	$5.73E - 05$	1.86	$4.36E - 11$	7.67
7	$10 \times 10$	$7.56E - 03$	$\dots$	$3.04E - 03$	$\dots$
	$20 \times 20$	$2.02E - 03$	1.90	$6.50E - 05$	5.55
	$40 \times 40$	$5.37E - 04$	1.91	$3.29E - 07$	7.63
	$80 \times 80$	$1.47E - 04$	1.87	$8.22E - 10$	8.64
	$160 \times 160$	$4.03E - 05$	1.87	$2.72E - 12$	8.24

signifies the computational efficiency achieved by applying the higher-order methods. Table 2 makes a comparison between the results obtained in the present research and a number of the available results reported in the literature. As it is shown, the accuracy of the DG method is much higher than the FV method even if a higher-order WENO flux function is employed.

### 5.3 Periodic Deformation of a Slotted Disk in a Swirl Flow

This section is assigned to verify the solution to the level set advection equation (7) by simulating the periodic deformation of an eccentric slotted disk in a prescribed velocity field corresponding to a time-dependent swirling flow expressed as the equation 70.

**Problem Description** The domain of computation is a square with the lower-left corner located at  $(0, 0)$  and the upper-right corner located at  $(1, 1)$ . The initial geometry of the interface



**Figure 12** Level set  $\mathbb{L}^2$ -error versus  $N_{DoF}$  for simulating a periodic deformation of a circle in a swirl flow

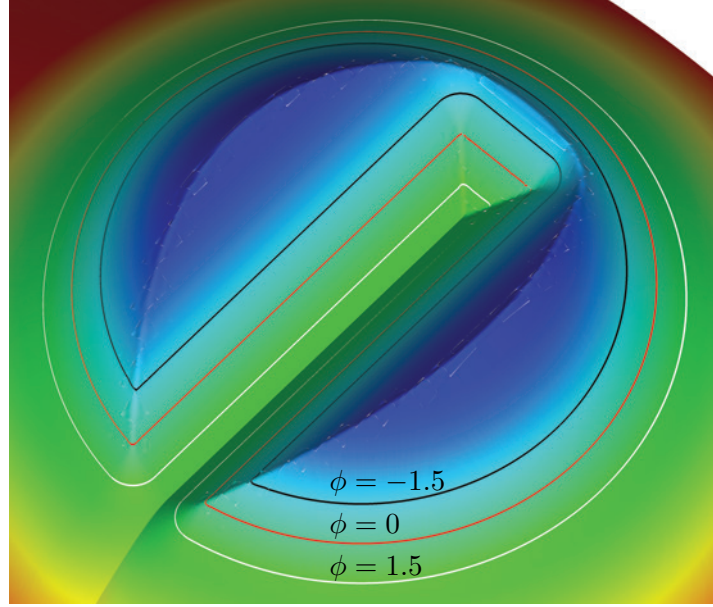
**Table 2** Accuracy of the DG method comparing to a higher-order WENO Finite Volume method for simulating the periodic deformation of a circle in a swirl flow

Method	$p$	$N_C$	$N_{DoF}$	SDLS	Area Loss (%)	$\mathbb{L}^1$ -error
DG <sup>1</sup>	4	$32 \times 32$	15360	Yes	-3.81	4.38E-03
DG <sup>1</sup>	4	$32 \times 32$	15360	No	0.1	7.73E-04
DG <sup>2</sup>	4	$32 \times 32$	15360	No	0.71	5.5E-4
5 <sup>th</sup> -Order WENO FV <sup>3</sup>	0	$128 \times 128$	16384	Yes	-39.8	3.1E-2
5 <sup>th</sup> -Order WENO FV <sup>4</sup>	0	$128 \times 128$	16384	Yes	-0.71	1.4E-3

<sup>1</sup> Present. <sup>2</sup> DG method [7]. <sup>3</sup> Classical level set method [47]. <sup>4</sup> Particle level set method [47]

is a notched disk with the radius  $R = 0.15$ , the slot length  $L_{Slot} = 0.25$  and the slot width  $W_{Slot} = 0.5$ , centered at  $(x_c = .5, y_c = 0.75)$ . The manually constructed signed-distance level set function of the interface is shown in the figure 13. The period of deformation is set to  $T = 8$ .

**Numerical Settings** The domain is discretized to a set of the quadrilateral cells with  $N_C = 100 \times 100$ , according to which, the slot passes 10 cells along its width. The degree of the orthonormal basis polynomial space is set to  $p = 7$ . The time step is set to  $\Delta t = 0.0002$  corresponding to the CFL number of  $1/p^2$ . A homogeneous Neumann boundary condition is imposed on the entire boundary of the domain.



**Figure 13** The initial signed distance level set function of a slotted disk. The picture corresponds to  $p = 7$  and  $N_C = 100 \times 100$ .

**Results** Figure 14 shows snapshots of the interface shapes captured in one period of deformation, using  $p = 7$ . As it is shown in the figure, despite the complexity in the geometry of the interface as well as the long period of deformation, the original shape of the interface is fairly regained at  $t = T$ . The area loss is  $-1.85\%$  and the interface  $L^1$ -error is 0.000974.

#### 5.4 Re-initializing the Level Set Function of a Circle

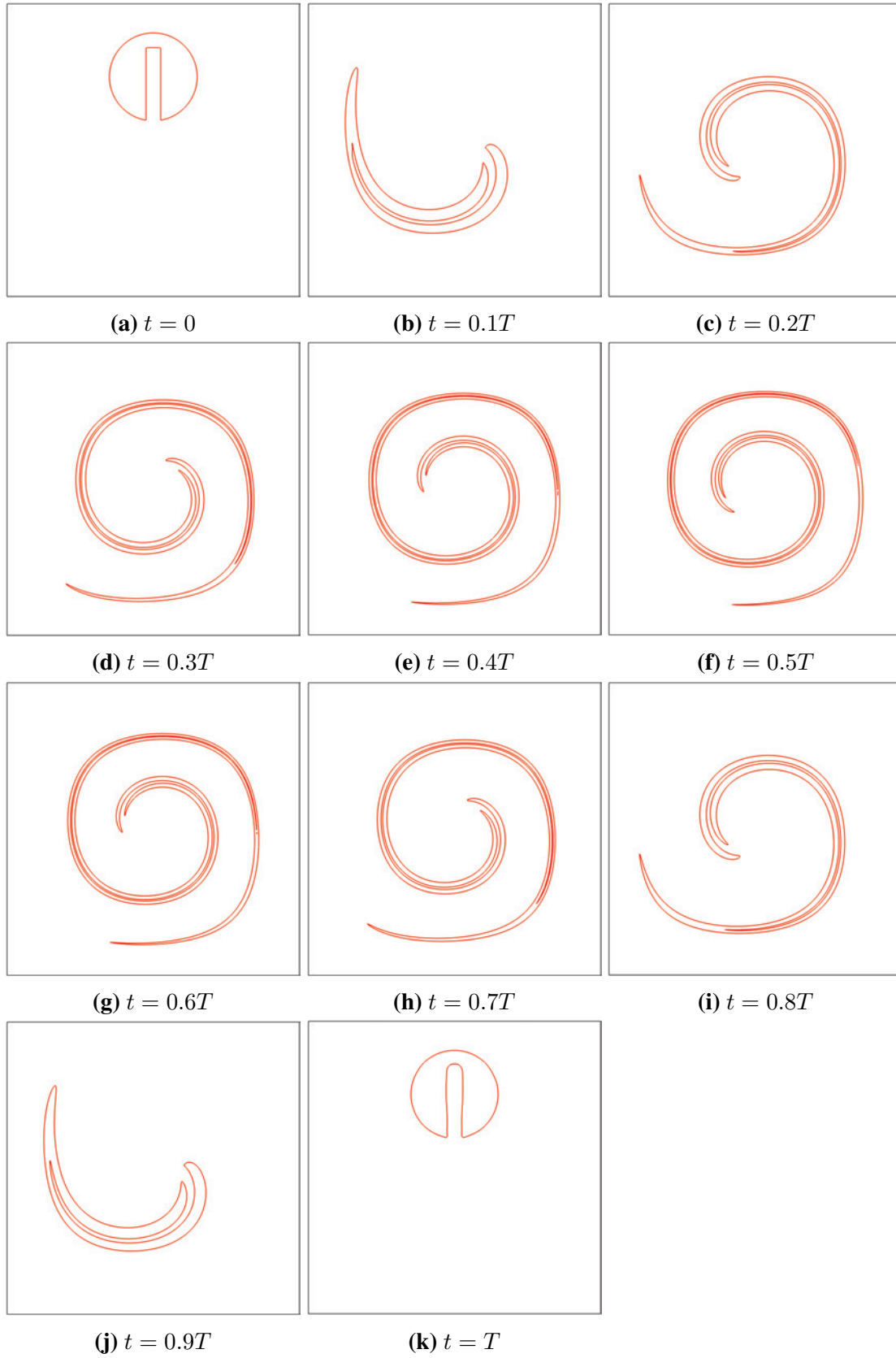
This section is assigned to verify the solution to the level set re-initialization equation (15) by re-initializing the level set function of a circle. As a result of the simple geometry of the interface, this test case clarifies the numerical challenges inherent in the application of the DG method for solving the re-initialization equation.

**Problem Description** The domain of computation is a square with the lower-left corner located at  $(-0.5, -0.5)$  and the upper-right corner located at  $(0.5, 0.5)$ . The interface geometry is a circle with the radius  $R = 0.25$ , centered at  $(x_c = 0, y_c = 0)$ . The initial level set function of the interface is analytically expressed as,

$$\phi^0(\mathbf{x}) = 0.5 \cdot \left( \sqrt{(x - x_c)^2 + (y - y_c)^2} - R \right), \quad (73)$$

which corresponds to a gradient value or a slope of 0.5. As it is discussed in the section 5.2, the gradient of this level set function is singular at the center of the circle.

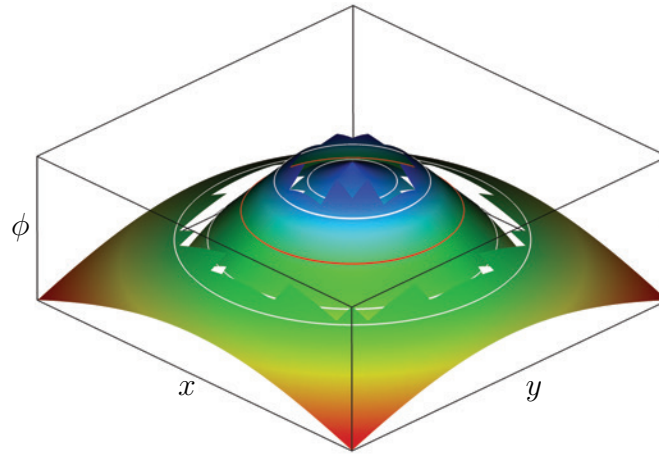
**Numerical Setting** The domain is discretized to a set of the quadrilateral cells with  $N_C = 20 \times 20$ , according to which, the circle passes 10 cells along its diameter. The degree of the orthonormal basis polynomial space is set to  $p = 5$ . The time step is set to  $\Delta t = 0.002$  corresponding to the CFL number of  $1/p^2$ . The regularized Signum function 21 is used with the smoothing parameter  $\epsilon = h$ . As the signed distance property of a level set function needs to be recovered only in the region where the interface diffusion takes place, the re-initialization



**Figure 14** Periodic deformation of an eccentric slotted disk in a swirl flow. The picture corresponds to  $p = 7$  and  $N_C = 100 \times 100$ .

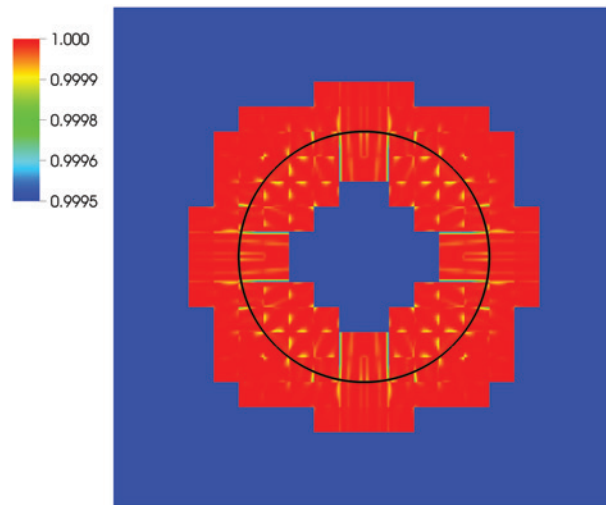
equation is solved only in a narrow band around the interface in order to save the computational cost. In the direction normal to the interface, the narrow band includes a cell which is cut by the interface and one neighbor cell at each side. A homogeneous Neumann boundary condition is imposed on the boundary of the narrow band.

**Results** Figure 15 shows the level set function re-initialized in a narrow band around the interface. The red curve represents  $\phi = 0$  and the white curves represent  $\phi = -0.075$  and  $\phi = 0.075$ . As the re-initialization is not supposed to move the zero iso-value of the level set function (the interface), only one red curve is observed in the picture. But as the other iso-values can be moved, two iso-values corresponding to  $\phi = -0.075$  and two iso-values corresponding to  $\phi = 0.075$  are formed after the re-initialization. Figure 16 shows the gradient value of



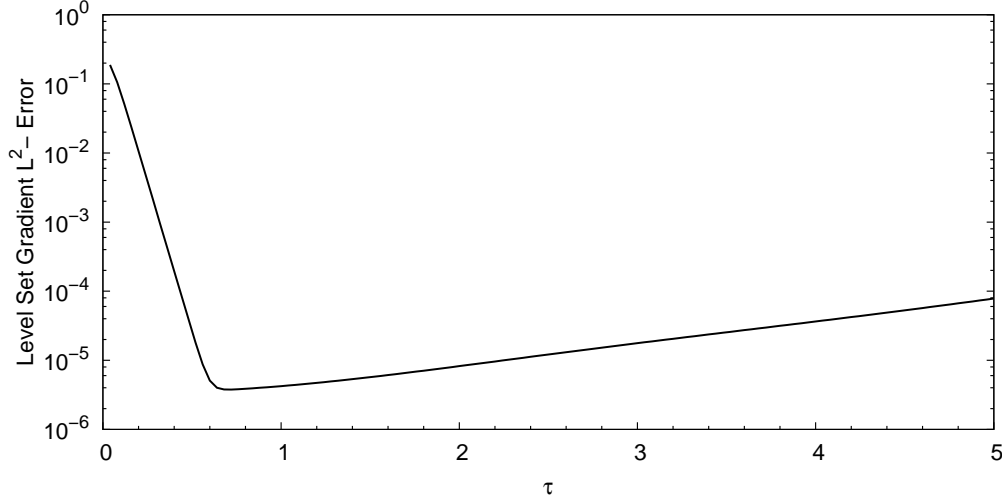
**Figure 15** The level set function of a circle with the initial slope of 0.5 re-initialized in a narrow band around the interface. The picture corresponds to  $p = 5$  and  $N_C = 20 \times 20$ .

the level set function after the re-initialization, implying spurious spatial oscillations in the field of the level set function. Diagram 17 shows the pseudo-time history of the  $\mathbb{L}^2$ -error of



**Figure 16** Gradient value of the re-initialized level set function of a circle. The initial slope of the level set function is 0.5. The picture corresponds to  $p = 5$  and  $N_C = 20 \times 20$ .

$|\nabla\phi|$  during the re-initialization. As it is shown, although the error decreases at the beginning with an appropriate rate, the solution starts to diverge after a certain pseudo-time. This is in connection with the spurious spatial oscillation observed in the figure 16. Following [9], the



**Figure 17**  $\mathbb{L}^2$ -error of  $|\nabla\phi|$  during the re-initialization of the level set function of a circle. The initial slope of the level set function is 0.5. The degree  $p = 5$  is used with  $N_C = 20 \times 20$ .

solution stability can be improved by adding an artificial diffusion along the characteristic lines of the solution as,

$$\frac{\partial\phi}{\partial\tau} + \text{Sign}(\phi^0)(|\nabla\phi| - 1) - \nu_{\mathbf{n}} \frac{\partial^2\phi}{\partial\mathbf{n}^2} = 0 \quad (74)$$

where  $\nu_{\mathbf{n}}$  is the coefficient of the artificial diffusion and  $\mathbf{n}$  is the interface normal vector which is calculated as,

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}.$$

As

$$\frac{\partial\phi}{\partial\mathbf{n}} = \mathbf{n} \cdot \nabla\phi,$$

the diffusion term can be calculated as,

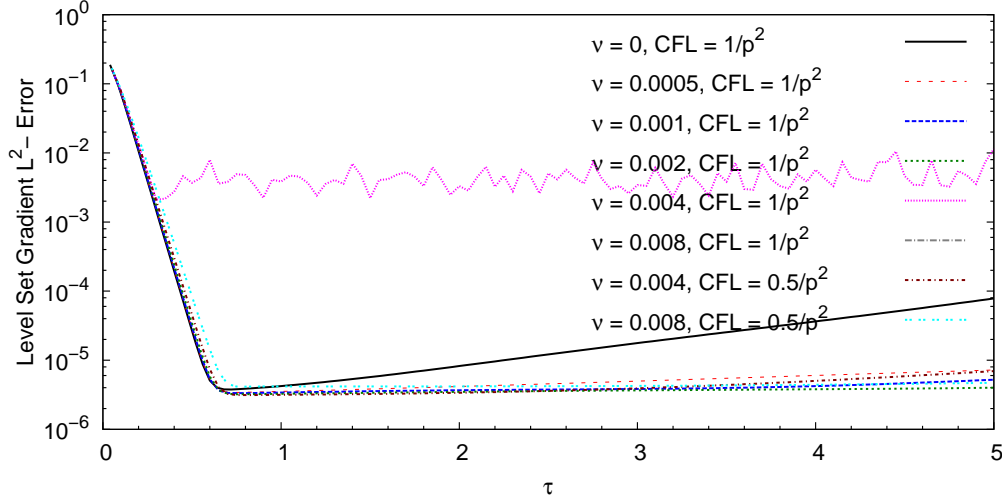
$$\nu_{\mathbf{n}} \left( \frac{\partial^2\phi}{\partial\mathbf{n}^2} \right) = \nu_{\mathbf{n}} (\mathbf{n} \cdot \nabla (\mathbf{n} \cdot \nabla\phi)).$$

As  $|\nabla\phi| = 1$ , therefore,

$$\nu_{\mathbf{n}} (\mathbf{n} \cdot \nabla (\mathbf{n} \cdot \nabla\phi)) = 0,$$

which means that the diffusion term does not affect the solution in the steady state. Diagram 18 shows the effect of adding a diffusion term to the re-initialization equation. As it is shown, using the diffusion coefficients  $\nu_{\mathbf{n}} = 0.0005, 0.001, 0.002$  has a notable stabilization effect. As adding a second-order term to the equation imposes more CFL restriction, a lower time step is required in order to use larger diffusion coefficients. Table 3 reports the effects of adding a diffusion term, on the area error, interface  $\mathbb{L}^1$ -error and the  $\mathbb{L}^2$ -error of the level set gradient after performing a re-initialization. The errors, indicate that the spurious movement of the interface as a result of the re-initialization, is quite small and it is further reduced by adding an artificial



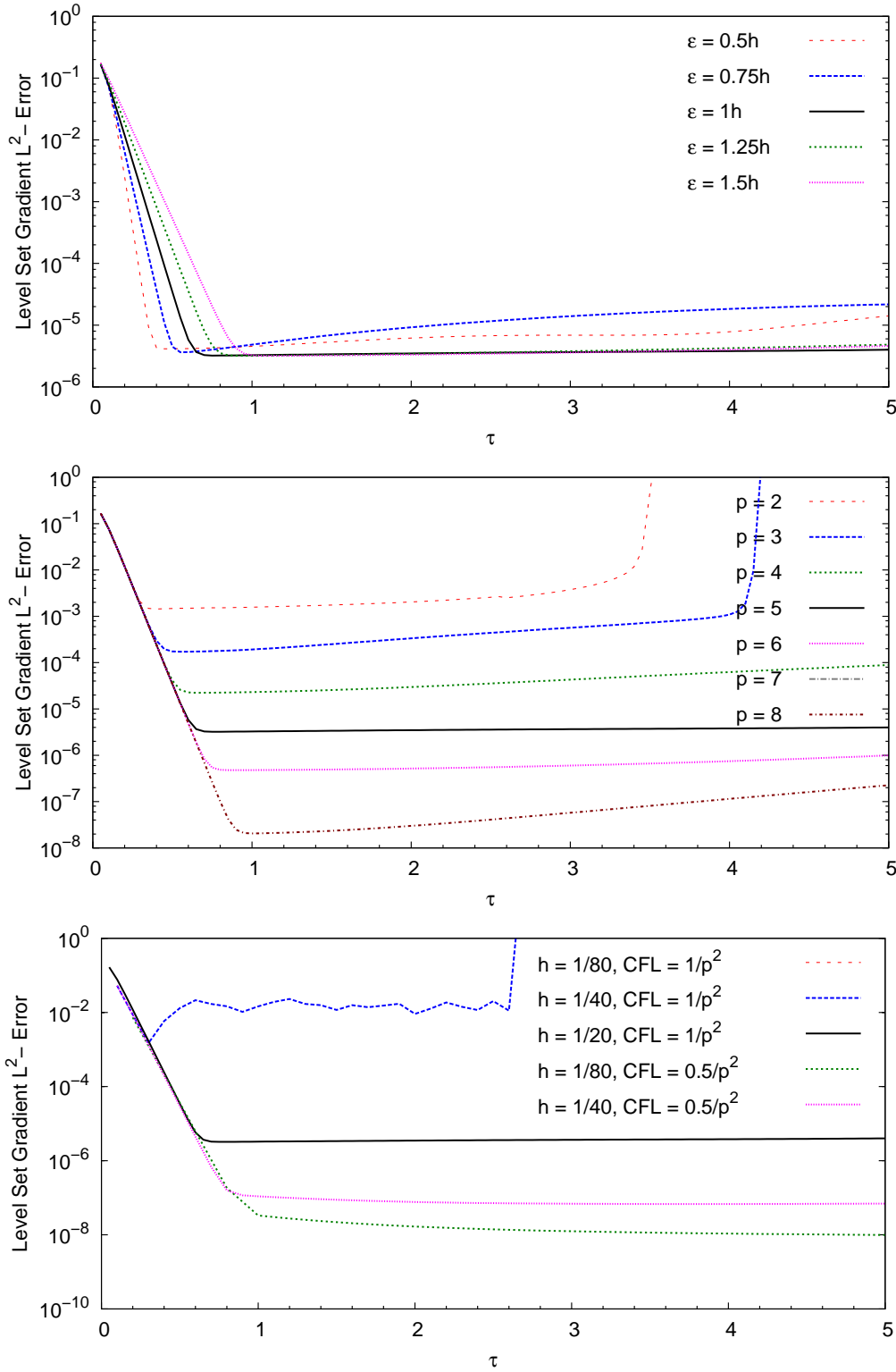


**Figure 18** Effect of the artificial diffusion on the  $\mathbb{L}^2$ -error of  $|\nabla\phi|$  during the re-initialization of the level set function of a circle. The initial slope of the level set function is 0.5. The degree  $p = 5$  is used with  $N_C = 20 \times 20$ .

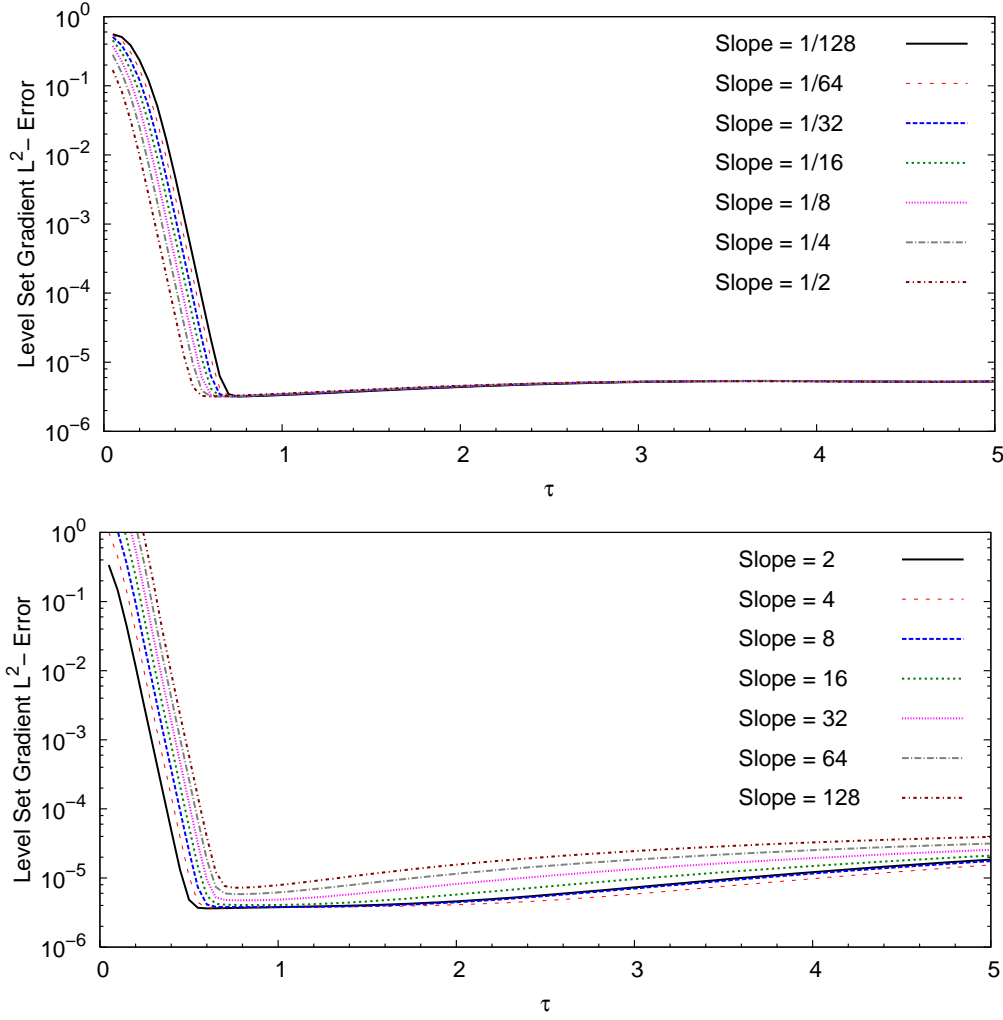
**Table 3** Effects of adding a diffusion term to the re-initialization equation for re-initializing the level set function of a circle. The degree  $p = 5$  is used with  $N_C = 20 \times 20$ .

	$\nu = 0$	$\nu = 0.002$
Area Loss (%)	-0.00136	-0.00154
Interface $\mathbb{L}^1$ -error	$4.0385E - 06$	$3.914E - 07$
LS Gradient $\mathbb{L}^2$ -error	$2.616E - 05$	$4.762E - 06$

diffusion to the re-initialization equation. Figure 19 shows the effects of the smoothing width  $\epsilon$ , the degree  $p$  and the characteristic cell size  $h$  as well as the artificial diffusion, on the pseudo-time history of the  $\mathbb{L}^2$ -error of  $|\nabla\phi|$ . As it is shown, increasing the smoothing width of the Signum function improved the stability although it reduces the convergence rate. Moreover, it is shown that increasing the degree of the orthonormal polynomial space, improves the stability in addition to increasing the accuracy. Furthermore, it is shown that although increasing the grid resolution improves the stability, it imposes more restriction on the CFL number. Figure 20 shows the pseudo-time history of the  $\mathbb{L}^2$ -error of  $|\nabla\phi|$  to verify the conditional Signum function (23) for re-initializing the level set functions with very small or very large initial slopes. The diagrams indicate the proper performance of this Signum function.



**Figure 19** Effects of  $\epsilon$ ,  $p$  and  $h$  as well as the artificial diffusion, on the pseudo-time history of the  $L^2$ -error of  $|\nabla\phi|$  during the re-initialization of the level set function of a circle. The initial slope of the level set function is 0.5.



**Figure 20** pseudo-time history of the  $L^2$ -error of  $|\nabla\phi|$  to verify the Signum function (23) for re-initializing the level set functions with very small or very large initial slopes

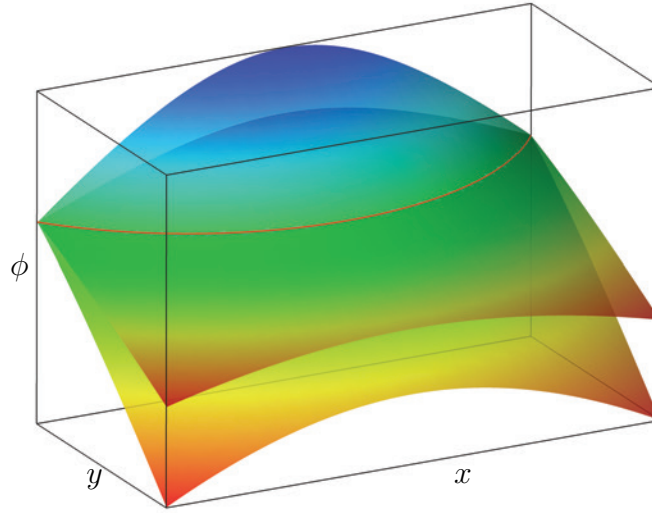
### 5.5 Re-initializing the Level Set Function of an Arc

This section is assigned to verify the solution to the level set re-initialization equation (15) by re-initializing the level set function of an arc. This test case is mainly aimed to investigate the  $h$ -convergence of the solution to the re-initialization equation. Such an investigation can not be done for the test case 5.4 because of the singularity in the gradient of the signed-distance level set function of a circle. On the other hand, if the re-initialization equation is solve over a narrow band around the interface in order to leave the point of singularity outside the domain of computation, the the narrow bands in different grid resolutions do not have the same thickness. Therefore, in the present test case an arc with the center located outside the domain of computation is considered and the re-initialization equation is solved over the entire domain.

**Problem Description** The domain of computation is a rectangle with the lower-left corner located at  $(-0.5, -0.25)$  and the upper-right corner located at  $(0.5, 0.25)$ . The interface geometry is a circular arc with the radius 0.625 centered at  $(x_c = 0, y_c = 0.625)$ . The initial level set function is analytically expressed by the equation 73 that corresponds to an initial slope of 0.5.

**Numerical Setting** The domain is discretized to a set of the quadrilateral cells with  $N_C = 10 \times 5$ ,  $N_C = 20 \times 10$ ,  $N_C = 40 \times 20$ . The degree of the orthonormal basis polynomial space is set to  $p = 5, 6, 7$ . The time step is set to  $\Delta t = 0.00025$ . A regularized Signum function  $Sign(\phi^0)$  is used with the smoothing parameter  $\epsilon$  of 0.1.

**Results** Figure 21 shows the level set function before and after the re-initialization. Table 4 lists the  $\mathbb{L}^2$ -errors of the gradient value of the level set function as well as the experimental error orders which are calculated using the expression (72). This table indicates an acceptable  $h$ -convergence of the solution.



**Figure 21** Re-initialization of the level set function of an arc with the initial slope of 0.5. The picture corresponds to  $p = 5$  and  $N_C = 10 \times 5$ .

**Table 4**  $h$ -Convergence study on the  $\mathbb{L}^2$ -Error of  $|\nabla \phi|$  for re-initializing the level set function of an arc with the initial slop of 0.5.

$p$	$N_C$	$\mathbb{L}^2$ -Error of $ \nabla \phi $	EEO
5	$10 \times 5$	$3.10E - 06$	...
	$20 \times 10$	$3.08E - 08$	6.65
	$40 \times 20$	$8.34E - 10$	5.21
6	$10 \times 5$	$5.90E - 07$	...
	$20 \times 10$	$2.53E - 09$	7.87
	$40 \times 20$	$2.61E - 11$	6.59
7	$10 \times 5$	$5.61E - 08$	...
	$20 \times 10$	$3.58E - 10$	7.29
	$40 \times 20$	$2.67E - 12$	7.07

## 6 CONCLUSIONS

- A higher-order Modal Discontinuous Galerkin Method was successfully applied for solving the level set advection and re-initialization equations.
- As a result of the higher/order approximation, an accurate solution to the level set advection equation obtained without a necessity to solve the re-initialization equation.
- Concerning the solution to the level set advection equation, a singularity in the gradient of the level set function resulted in a large reduction of the the  $hp$ -convergence rate.
- Employing the Godunov's scheme in approximating the Hamiltonian of the level set re-initialization equation as well as adding a diffusion term, resulted in a stable solution to the equation.
- As a result of the higher/order approximation, an accurate solution to the re-initialization equation was found which does not move the interface spuriously.
- As a future work, filtering the solution of the re-initialization equation is expected to improve the long-term stability.

## 7 Acknowledgement

The present research was accomplished under a PhD grant from the Center of Smart Interfaces at Darmstadt University of Technology.

## REFERENCES

- [1] F. Kummer, *The BoSSS Discontinuous Galerkin solver for incompressible fluid dynamics and an extension to singular equations*. PhD Dissertation, Technische Universität Darmstadt, 2012.
- [2] R. Mousavi, *Level Set Method for Simulating the Dynamics of the Fluid-Fluid Interfaces: Application of a Discontinuous Galerkin Method*. PhD Dissertation, Technische Universität Darmstadt, 2014.
- [3] Y. Shikhmurzaev, *Capillary Flows with Forming Interfaces*. Chapman and Hall-CRC, 2007.
- [4] D. M. Anderson, G. B. McFadden, A. A. Wheeler, Diffuse-Interface Methods in Fluid Mechanics. *Annual Review of Fluid Mechanics*, **30**, 139–165, 1998.
- [5] S. Osher, R. Fedkiw, *Level-set methods and dynamic implicit surfaces*. Springer-Verlag New York, Inc., 2003.
- [6] D. A. D. Pietro, A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods*. Springer-Verlag, 2011.
- [7] E. Marchandise, J. F. Remacle, N. Chevaugeon, A quadrature-free discontinuous Galerkin method for the level set equation. *Journal of Computational Physics*, **212**, 338–357, 2006.
- [8] S. Zahedi, K. Gustavsson, G. Kreiss, A conservative level set method for contact line dynamics. *Journal of Computational Physics*, **228**, 6361–6375, 2009.

- [9] J. Grooss, J. S. Hesthaven, A level set discontinuous Galerkin method for free surface flows. *Journal of Computer Methods in Applied Mechanics and Engineering*, **195**, 3406–3429, 2006.
- [10] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows. *Computers and Fluids*, **27**, 663–680, 1998.
- [11] J. H. Ferziger, M. Peric, *Computational Methods for Fluid Dynamics*. Springer-Verlag, 2002.
- [12] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling Merging and Fragmentation in Multiphase Flows with SURFER. *Journal of Computational Physics*, **113**, 134–147, 1994.
- [13] J. Glimm, J. Grove, B. Lindquist, The bifurcation of tracked scalar waves. *SIAM Journal on Scientific and Statistical Computing*, **9**, 61–79, 1988.
- [14] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, **79**, 12–49, 1988.
- [15] S. O. Unverdi, G. Tryggvason, A Front-Tracking Method for Viscous, Incompressible, Multi-fluid Flows. *Journal of computational physics*, **100**, 25–37, 1992.
- [16] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A Hybrid Particle Level Set Method for Improved Interface Capturing. *Journal of Computational Physics*, **183**, 83–116, 2002.
- [17] F. H. Harlow, J. E. and Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, **8**, 2182–2189, 1965.
- [18] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, **39**, 201–225, 1981.
- [19] D. L. Youngs, M. J. Morton, M. J. Baines, *Time-dependent multi-material flow with large fluid distortion*, in: *Numerical Methods for Fluid Dynamics*. Academic Press, New York, 1982.
- [20] O. Ubbink, *Time-dependent multi-material flow with large fluid distortion*, in: *Numerical Methods for Fluid Dynamics*. PhD Dissertation, Imperial College of Science, Technology and Medicine, 1997.
- [21] M. Sussman, E. G. Puckett, A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, **162**, 301–337, 2000.
- [22] M. Raessi, J. Mostaghimi, M. Bussmann, Advecting normal vectors: A new method for calculating interface normals and curvatures when modeling two-phase flows. *Journal of Computational Physics*, **226**, 774–797, 2007.
- [23] J. A. Sethian, A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proceedings of National Academy of Science*, **93**, 1591–1595, 1998.

- [24] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, **114**, 146–159, 1994.
- [25] E. Rouy, A. Tourin, A viscosity solutions approach to shape-form-shading. *SIAM journal on Numerical Analysis*, **29**, 867–884, 1992.
- [26] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An Adaptive Level Set Approach for Incompressible Two-Phase Flows. *Journal of Computational Physics*, **148**, 81–124, 1999.
- [27] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method. *Journal of Computational Physics*, **155**, 410–438, 1999.
- [28] C. W. Shu, High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. *International Journal of Computational Fluid Dynamics*, **17**, 107–118, 2003.
- [29] M. Ainsworth, Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *International Journal of Computational Fluid Dynamics*, **198**, 106–130, 2004.
- [30] G. E. Karniadakis, S. J. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, USA, 2005.
- [31] X. D. Liu, S. Osher, T. Chan, Weighted Essentially Non-Oscillatory Schemes. *Journal of Computational Physics*, **126**, 202–212, 1996.
- [32] B. Fornberg, *A practical guide to pseudospectral methods*. Cambridge University Press, 1998.
- [33] L. N. Trefethen, *Finite difference and spectral methods for ordinary and partial differential equations*. Cornell University, 1996.
- [34] S. Gottlieb, C. W. Shu, Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation*, **67**, 73–85, 1998.
- [35] A. Harten, High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, **49**, 357–393, 1983.
- [36] C. Hirsch, *Numerical Computation of Internal and External Flows: Fundamentals of Computational Fluid Dynamics*. Butterworth-Heinemann, 2007.
- [37] N. Emamy, *BoSSS for solving conservation laws*. Annual report, Technische Universität Darmstadt, Fachgebiet Strömungsdynamik, 2010.
- [38] E. Marchandise, N. Chevaugeon, J. F. Remacle, Spatial and spectral superconvergence of discontinuous Galerkin method for hyperbolic problems. *Journal of Computational and Applied Mathematics*, **215**, 484–494, 2010.
- [39] N. Emamy, *Numerical simulation of a droplet in a stationary electric field using DG*. PhD Dissertation, Technische Universität Darmstadt, 2014.

- [40] R. Kress, *Numerical analysis*. Springer-Verlag New York, 1998.
- [41] M. G. Crandall, P. L. Lions, Viscosity Solutions of Hamilton-Jacobi Equations. *Transactions of the American Mathematical Society*, **277**, 1–42, 1983.
- [42] M. G. Crandall, P. L. Lions, Two Approximations of Solutions of Hamilton-Jacobi Equations. *Mathematics of Computation*, **43**, 1–19, 1984.
- [43] S. Osher, C. W. Shu, High-Order Essentially Nonoscillatory Schemes for Hamilton-Jacobi Equations. *SIAM Journal on Numerical Analysis*, **28**, 907–922, 1991.
- [44] M. Bardi, S. Osher, The Nonconvex Multidimensional Riemann Problem for Hamilton-Jacobi Equations. *SIAM Journal on Mathematical Analysis*, **22**, 344–351, 1991.
- [45] J. Yan, S. Osher, A local discontinuous Galerkin method for directly solving Hamilton-Jacobi equations. *Journal of Computational Physics*, **230**, 232–244, 2011.
- [46] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, **33**, 627–665, 1996.
- [47] D. Enright, F. Losasso, R. Fedkiw, A Fast and Accurate Semi-Lagrangian Particle Level Set Method. *Computers and Structures*, **83**, 479–490, 2004.