Further development of Level Set method : modified level set equation and its numerical assessment

Andrey Ovsyannikov

N° d'ordre : 2013-13 **ANNÉE 2013**

THÈSE

présentée devant

L'ÉCOLE CENTRALE DE LYON

pour obtenir

le titre de DOCTEUR

SPÉCIALITÉ MÉCANIQUE DES FLUIDES

par

Andrey OVSYANNIKOV

Nouveau développement de la méthode Level Set sur la base d'une équation modifiée de suivi d'interface

Further development of Level Set method: modified level set equation and its numerical assessment

Soutenue le 10 Juin 2013

JURY

Marcus Herrmann	Université d'Arizona, USA (rapporteur)
Stéphane Vincent	Université Bordeaux 1 (rapporteur)
Stéphane Zaleski	UPMC, l'Institut d'Alembert
Alain Berlemont	CNRS UMR 6614 CORIA Rouen
Mikhael Gorokhovski	LMFA, Centrale de Lyon (directeur de thèse)
Vladimir Sabel'nikov	DEFA-ONERA (co-directeur de thèse)
Alexey Kudryavtsev	ITAM, Russie (membre invité)
Giovanni Russo	Université de Catane, Italie (membre invité)

Laboratoire de Mécanique des Fluides et d'Acoustique, UMR CNRS 5509
Ecole Centrale de Lyon

Abstract

The level set method was introduced by Osher & Sethian (1988) as a general technique to capture moving interfaces. It has been used to study crystal growth, to simulate water and fire for computer graphics applications, to study two-phase flows and in many other fields. The well-known problem of the level set method is the following: if the flow velocity is not constant, the level set scalar may become strongly distorted. Thus, the numerical integration may suffer from loss of accuracy. In level set methods, this problem is remedied by the reinitialization procedure, *i.e.* by reconstruction of the level set function in a way to satisfy the eikonal equation. We propose an alternative approach. We modify directly the level set equation by embedding a source term. The exact expression of this term is such that the eikonal equation is automatically satisfied. Furthermore on the interface, this term is equal to zero. In the meantime, the advantage of our approach is this: the exact expression of the source term allows for the possibility of derivation of its local approximate forms, of first-and-higher order accuracy. Compared to the extension velocity method, this may open the simplifications in realization of level set methods. Compared to the standard approach with the reinitialization procedure, this may give the economies in the number of level set re-initializations, and also, due to reduced number of re-initializations, one may expect an improvement in resolution of zero-set level. Hence, the objective of the present dissertation is to describe and to assess this approach in different test cases.

Keywords: two-phase flow, level set method, re-initialization, signed-distance function, WENO scheme.

Acknowledgements

This thesis was prepared and accomplished in the Laboratory of Fluid Mechanics and Acoustics in Ecole Centrale de Lyon.

I would like to thank my advisor **Prof. Mikhael Gorokhovski**, who provided me the opportunity of this thesis and awakened my interest in simulation of two-phase flows. Also I would like to thank my co-advisor **Dr. Vladimir Sabel'nikov** for his constant support and guidance in my research.

I would like to thank the members of my PhD defense committee, **Prof. Stéphane Zaleski**, **Prof. Alain Berlemont**, **Prof. Giovanni Russo**, **Prof. Marcus Herrmann**, **Dr. Stéphane Vincent** and **Dr. Alexey Kudryavtsev** for their time and their useful comments and support of this work.

I acknowledge support from the Center for Turbulence Research (Stanford University), and in particular, the hospitality of **Prof. Parviz Moin** during the Summer Program 2012. I also acknowledge support from MAGIE project (Modélisation et Approche Générique de l'Injection Essence) with Continental Automotive, France.

Also, I am very thankful to my colleague and my friend, **Andrey Pushkarev**, who gave me much encouragement and help during these 3 years in Lyon.

Lastly, but most importantly, I am forever in debt to **my great family**, without whom none of this would have been possible.

Lyon, June 2013

Andrey Ovsyannikov

Contents

Nomenclature

Latin letters

A	source term coefficient
C	volume fractions
d	signed-distance function
d_k	WENO optimal weights
D	deformation rate tensor
E	total energy
$E_{\|\nabla\varphi\|}$	gradient error
E_{shape}	shape error
F_{ext}	extension velocity
Fr	Froude number
G	level set function
H	Heaviside function
I	identity tensor
IS_k	smoothness indicators
k	wavenumber
L	length scale
L_2	Lebesgue space
$\vec{n}$	interface normal vector
n_i	components of the normal vector
n	distance to the interface
NB	narrow band domain
N_{iter}	number of iterations

Greek letters

α_k	WENO non-normalized weights
β	narrow band parameter
γ	narrow band parameter
Γ	interface
δ	Dirac delta function
δ_{ij}	Kronecker symbol
χ	characteristic function
ε	smoothing parameter
η	viscosity ratio
κ	interface curvature
λ	density ratio
μ	dynamic viscosity
ν	kinematic viscosity
φ	new level set function
φ_0	initial distribution of φ
ψ	diffuse interface scalar
ρ	density
σ	coefficient of surface tension
τ	stress tensor
ω	vorticity
ω_k	WENO non-linear weights
Ω	physical domain

p	pressure
R^n	Euclidean space
R_+	interval $[0,+\infty)$ of real numbers
Re	Reynolds number
S_{ij}	components of velocity gradient tensor
t	time
t'	pseudo-time in reinitialization
$\vec{u}$	velocity vector
u_i	components of velocity
$\vec{u}^{ext}$	extension velocity
We	Weber number
$\vec{x}$	Cartesian coordinate
x_i	spatial variables
$\vec{x}_f$	location of the front
$\Delta x, \Delta y$	grid resolution
Δt	time step
$\Delta t'$	pseudo-time step
Ω_1	region occupied by fluid 1
Ω_2	region occupied by fluid 2

Subscripts and superscripts

1	fluid one
2	fluid two
$*$	reference value
LA,k	k^{th} – order local approximation
n	time step n
$n+1$	time step $n+1$
T	transpose

Other

∂	partial derivative
∇	Nabla operator
∇_i	partial derivative along x_i
$O(\ )$	order of magnitude
$\big\vert_{\vec{x}}$	quantity taken at point $\vec{x}$
$\big\vert_{\varphi=0}$	quantity taken at the front
$\Vert\ \Vert_2$	norm in L_2

Chapter 1

Introduction

1.1 Flows with interface

In many phenomena two bulk materials with different physical properties (between two different phases, for example) can coexist without molecular mixing. The surface which separates the immiscible phases is often referred to as *phase boundary*, or *interface* or *front*. For example, when the liquid is in contact with the gas, it forms an interface layer. The thickness of this layer is of order of typical radius of the intermolecular cohesion ($\sim 10^{-7}$ cm for simple molecules). In this layer, molecules interact not only with molecules in the liquid but also with neighboring molecules in the gas, which leads to difference in averaged free energy of molecules depending on proximity to the interface layer. For the molecules inside the liquid bulk, forces from all directions may cancel each other out, and the molecules may remain at near equilibrium. The molecules that are at the surface are pulled into the liquid bulk. Therefore the surface layer is under intensive agitation; the molecules at the surface are continuously replaced through their motion into the bulk. The statistical result of such molecular agitation is the surface tension, providing specific physical properties of the interface layer, in difference with the interior of the liquid bulk. Starting from the pioneer studies of Gibbs, the interface layer between liquid and gas is usually considered as a zero mass elastic infinitely thin membrane. Its surface area is introduced as independent parameter in thermodynamics of heterogeneous systems. The thermodynamic force related to the interface is referred to as the surface tension. Over two centuries, since classical studies of Young, Laplace, and Gauss, such an interface was under extensive investigation. Its behavior is complex, depends on its configuration imposed by the flow: it may resist to stretching of the liquid flow, or may contract, thereby changing this flow. As illustration Fig. 1.1 shows photographs of the liquid jet breakup for three different frequencies of excitation.

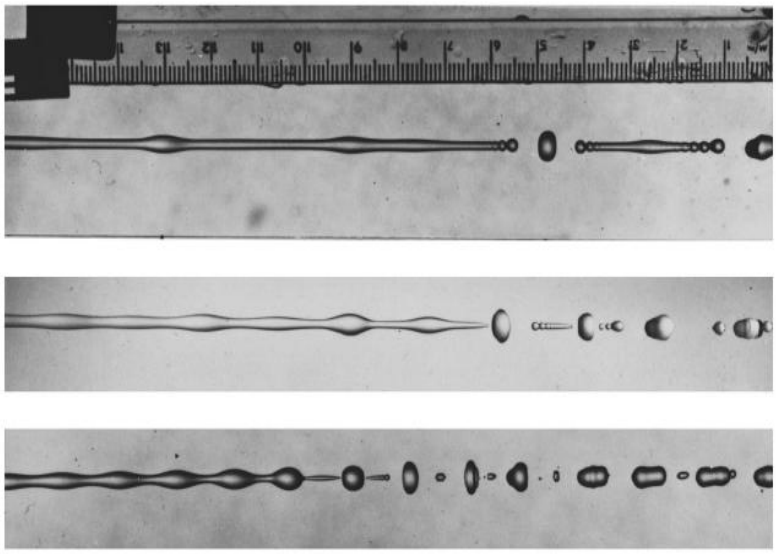


Figure 1.1: Photographs of a decaying jet for three different frequencies of excitation (from Rutland & Jameson, 1971).

Another example, where the discontinuity in physical properties is mimicked by evolution of a fluid-interface, is the premixed flame. According to the Arrhenius law, the chemical reaction is strongly sensible to the temperature (by changing the temperature from 1000°K to 2000°K, the reaction rate is 10^5 times increased). Then by heating the fresh combustible fuel-oxidizer mixture in the vicinity of the flame front, the zone of chemical reactions tends to propagate into and to consume the unburned mixture. If the chemical reactions occur at large activation energy (in hydrocarbons it is about 40 Kcal/mol compared to averaged molecular kinetic energy 3 Kcal/mol), the zone of chemical reaction is infinitely thin, and the premixed flame front may be considered as an interface separating the burned and unburned gases. The flow properties in the both mixtures impose a complex configuration of such interface: an example of the premixed flame patterns in Bunsen burner is shown in Fig. 1.2

Another typical example is the motion of a phase boundary between ice and water. This example is illustrated in Fig. 1.3 as ice formation in the polar seas, and the phenomenon is known as the Stefan problem. It is seen that the interface can shrink as the ice melts, or grow as the ice freezes; the speed of the interface propagation is depending on the temperature jump between ice and water.

In nature, in science and in technology, we find another numerous examples of flows with interface. This motivates to predict interfacial flows. One of the well-known approaches to such flows is level set method. Its further development is the main object in our work.

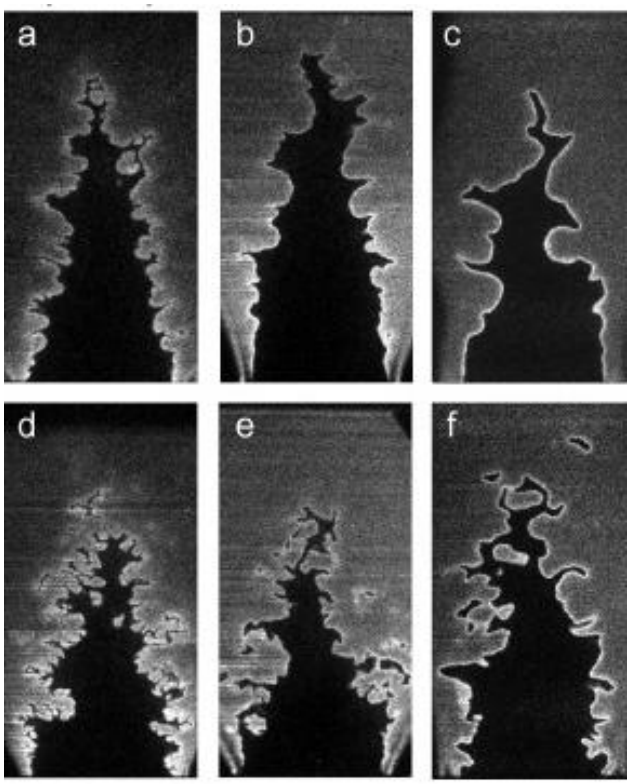


Figure 1.2: OH-PLIF images of turbulent premixed flames (Bunsen burner) with syngas/air mixture at different compositions (from Ichikawa *et al.*, 2011).

Figure 1.3: Ponds on the Arctic Ocean (NASA)

1.2 Navier - Stokes equations for two fluids

Consider the flow of two immiscible incompressible Newtonian fluids in domain Ω which is split by fluid interface Γ into two distinct parts Ω_1 and Ω_2 ; subscripts 1 is ascribed to the first fluid, and subscripts 2 is ascribed to the second one. The interface Γ is assumed to have no mass, it is elastic and infinitely thin, and the continuity condition is imposed across this interface (the phase change is not considered here). In each of the subdomains, the density and viscosity are assumed to be piecewise taken constant here for each fluid, respectively (ρ_1, μ_1) and $(\rho_2, \mu_2),$ but discontinuous at the interface. The conservation equations for mass and momentum valid in each subdomain $\Omega_{1,2} = \Omega_{1,2}(t)$ are

$$\begin{aligned} &\nabla \cdot \vec{u}_1 = 0, \;\; \rho_1 \frac{\mathrm{D}\vec{u}_1}{\mathrm{D}t} = -\nabla p_1 + 2\mu_1 \nabla \cdot D_1 + \rho_1 \vec{g}, \quad \vec{x} \in \Omega_1, \\ &\nabla \cdot \vec{u}_2 = 0, \;\; \rho_2 \frac{\mathrm{D}\vec{u}_2}{\mathrm{D}t} = -\nabla p_2 + 2\mu_2 \nabla \cdot D_2 + \rho_2 \vec{g}, \quad \vec{x} \in \Omega_2, \end{aligned} \tag{1.1}$$

where $\vec{u}$ is the velocity, p is the pressure and $\vec{g}$ is the acceleration due to gravity, $\mathrm{D}/\mathrm{D}t = \partial_t + \vec{u} \cdot \nabla$ is the material derivative, D is the rate of deformation tensor, characterized by components $D_{i,j} = (\partial_i u_j + \partial_j u_i)/2.$ The classical surface stress boundary condition between these two moving fluids is given in book of Landau & Lifshitz (1978) by use of the Laplace equation:

$$(2\mu_2 D_2 - 2\mu_1 D_1) \cdot \vec{n} - (p_2 - p_1)\vec{n} = \sigma \kappa \vec{n} \quad \text{and} \quad \vec{u}_2 = \vec{u}_1, \quad \vec{x} \in \Gamma, \tag{1.2}$$

where $\vec{n}$ is the outwards unit normal to the interface, σ is the coefficient of surface tension (here assumed to be constant), and $\kappa = \nabla \cdot \vec{n}$ is the local curvature of the interface.

Following definition of the stress tensor

$$\tau = -pI + 2\mu D, \tag{1.3}$$

where I is the identity tensor, another form of motion equations (1.1) is

$$\rho_1 \frac{\mathrm{D}\vec{u}_1}{\mathrm{D}t} = \nabla \cdot \tau_1 + \rho_1 \vec{g}, \quad \vec{x} \in \Omega_1 \tag{1.4}$$

$$\rho_2 \frac{\mathrm{D}\vec{u}_2}{\mathrm{D}t} = \nabla \cdot \tau_2 + \rho_2 \vec{g}, \quad \vec{x} \in \Omega_2 \tag{1.5}$$

with boundary conditions (1.2), as

$$(\tau_2 - \tau_1)\cdot\vec{n} = \sigma\kappa\vec{n}, \quad \vec{x}\in\Gamma. \tag{1.6}$$

Integrating (1.4) over Ω_1 and (1.5) over Ω_2 and using the divergence theorem, we obtain

$$\rho_1\int_{\Omega_1}\frac{\mathrm{D}\vec{u}_1}{\mathrm{D}t}d\vec{x} = \int_{\partial\Omega_1}\tau_1\vec{n}\,dS + \rho_1\int_{\Omega_1}\vec{g}d\vec{x}, \tag{1.7}$$

and

$$\rho_2\int_{\Omega_2}\frac{\mathrm{D}\vec{u}_2}{\mathrm{D}t}d\vec{x} = \int_{\partial\Omega_2}\tau_2\vec{n}\,dS + \rho_2\int_{\Omega_2}\vec{g}d\vec{x}. \tag{1.8}$$

We make the following definition

$$\tau = \begin{cases}\tau_1 & \text{if } x\in\Omega_1,\\ \tau_2 & \text{if } x\in\Omega_2.\end{cases} \tag{1.9}$$

Adding (1.7) and (1.8), and using definition of τ, we find

$$\rho_1\int_{\Omega_1}\frac{\mathrm{D}\vec{u}_1}{\mathrm{D}t}d\vec{x} + \rho_2\int_{\Omega_2}\frac{\mathrm{D}\vec{u}_2}{\mathrm{D}t}d\vec{x} = \int_{\partial\Omega}\tau\vec{n}\,dS - \int_{\Gamma}(\tau_2-\tau_1)\vec{n}\,dS + \rho_1\int_{\Omega_1}\vec{g}d\vec{x} + \rho_2\int_{\Omega_2}\vec{g}d\vec{x}. \tag{1.10}$$

Next we again use the divergence theorem to write

$$\int_{\partial\Omega}\tau\vec{n}\,dS = \int_{\Omega}\nabla\cdot\tau\,d\vec{x}. \tag{1.11}$$

Combining (1.9) - (1.11), we obtain

$$\rho_1\int_{\Omega_1}\frac{\mathrm{D}\vec{u}_1}{\mathrm{D}t}d\vec{x} + \rho_2\int_{\Omega_2}\frac{\mathrm{D}\vec{u}_2}{\mathrm{D}t}d\vec{x} = \int_{\Omega}\nabla\cdot\tau\,d\vec{x} - \int_{\Gamma}\sigma\kappa\vec{n}\,dS + \rho_1\int_{\Omega_1}\vec{g}d\vec{x} + \rho_2\int_{\Omega_2}\vec{g}d\vec{x}. \tag{1.12}$$

Now we make two more definitions

$$\vec{u} = \begin{cases}\vec{u}_1 & \text{if } x\in\Omega_1,\\ \vec{u}_2 & \text{if } x\in\Omega_2.\end{cases} \tag{1.13}$$

and

$$\rho = \begin{cases}\rho_1 & \text{if } x\in\Omega_1,\\ \rho_2 & \text{if } x\in\Omega_2.\end{cases} \tag{1.14}$$

Using that $\vec{u}_1 = \vec{u}_2$ at the interface then $\vec{u}$ is a continuous function. Using this fact and definitions

of $\vec{u}$ and ρ, we can write

$$\rho_1 \int_{\Omega_1} \frac{\mathrm{D}\vec{u}_1}{\mathrm{D}t} d\vec{x} + \rho_2 \int_{\Omega_2} \frac{\mathrm{D}\vec{u}_2}{\mathrm{D}t} d\vec{x} = \rho \int_{\Omega} \frac{\mathrm{D}\vec{u}}{\mathrm{D}t} d\vec{x}. \tag{1.15}$$

Combining (1.12) and (1.15) we obtain

$$\rho \int_{\Omega} \frac{\mathrm{D}\vec{u}}{\mathrm{D}t} d\vec{x} = \int_{\Omega} \nabla \cdot \tau \, d\vec{x} - \int_{\Gamma} \sigma \kappa \vec{n} \, dS + \rho \int_{\Omega} \vec{g} d\vec{x}. \tag{1.16}$$

This equation with appropriate boundary conditions for velocity and pressure constitute the physical model describing the motion of flows with interface. The problem is in the new extra-variable which is the interface. Thus this model requires its supplementary extension to find this new variable.

1.3 Main approaches in numerical identification of the interface

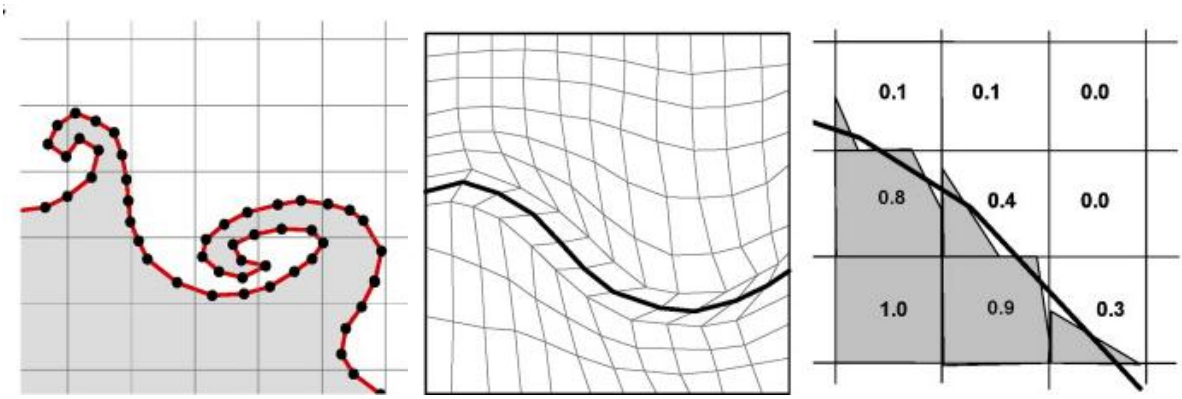
This Section describes approaches to simulation of interface dynamics. Since the interface is moving with the fluid, the time evolution of the interface is governed by the kinematic equation:

$$\frac{d\vec{x}_f}{dt} = \vec{u}(\vec{x}_f, t). \tag{1.17}$$

Once the interface is identified, the density and viscosity stepwise fields, the interaction between phases can be determined. Another way is to identify the interface with the fixed magnitude of a scalar, say $G = 0$. Then using definition of the area element associated with the iso-surface of this scalar $G = 0$, $dS = |\nabla G| \delta(G) d\vec{x}$, equation (1.16) takes the following form:

$$\rho(G) \frac{\mathrm{D}\vec{u}}{\mathrm{D}t} = \nabla \cdot \tau - \sigma \kappa(G) \nabla G \, \delta(G) + \rho(G) \vec{g}. \tag{1.18}$$

Once the scalar field G is known, and the topology of its zero magnitude is determined, it also allows formulating of interaction between phases. In general in computational methods for interfacial flows, (1.17) or (1.18) define the way of simulation. The approaches can be categorized into three groups: (a) *interface tracking* methods; (b) *interface fitting* methods; and (c) *interface capturing* methods. Schematic representations of these methods are given in Fig.1.4.



(a) (b) (c)

Figure 1.4. Representation of the interface by: (a) interface tracking method, (b) interface fitting method, and (c) interface capturing method.

The interface tracking method was pioneered by Harlow & Welch (1965) in their celebrated marker-and-cell (MAC) method for flows with free surface. In this approach, on the fixed mesh, a collection of marker particles is placed on the interface, and the motion of those particles is determined by Eulerian velocity field in (1.17). Originally, the MAC method was designed for one-phase flows with free boundary, but later on, it was generalized for two-phase flows (Daly, 1967) with particles seeded in the whole computational domain. In difference, Daly & Pracht (1968) used particles on the interface only. This procedure was strongly advanced by Unverdi & Tryggvason (1992); Glimm *et al*. (1998); Tryggvason *et al*. (2001) for the case of 3D flows with interface. A main difficulty of interface tracking methods is that the interface may merge or disintegrate. This requires to identify topology changes and to re-parameterize the interface. This is a very difficult task.

In the second group of approaches, referred often to as interface fitting methods, the interface is also controlled by (1.17), but dynamically at each time, the mesh boundary is setup to be attached to the interface and to follow the interface (Hirt *et al*., 1974). Thereby the mesh is deformed, by coarsening and refining, in order to be adapted to the interface in both extrados and intrados domains, but in such a way that its resolution quality remains computationally efficient for capturing the interface. Here also the main problem remains: coalescence of interface or its pinch-off requires application of complex algorithms (Quan *et al*., 2009).

In the third group of interface capturing methods, the interface is implicitly embedded in a scalar field function defined on a fixed Eulerian mesh, such as a Cartesian grid. Among the interface capturing methods, the main approaches are represented by the *volume of fluid* (VOF) method (Hirt & Nichols, 1981; Scardovelli & Zaleski, 1999), by the *level set method* (Osher & Sethian, 1988; Sussman *et al*., 1994) and by the *phase field method* (Anderson *et al*., 1998; Jacqmin, 1999; Lowengrub & Truskinovsky, 1998). In first two methods, the interface is

considered as a sharp front, while in the phase field method, the interface is diffused within narrow mixing layer between two fluids.

The VOF method is a widely used interface-tracking method. In this method, the both fluids are governed by a single momentum equation, and the volume fraction of each fluid is tracked throughout the domain. The mixed cells have a volume fraction between 0 and 1; and cells without interfaces (pure cells) have volume fraction equal to 0 or 1. The transport equation for volume fraction function C is represented by its continuity:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\vec{u}C) = 0. \tag{1.19}$$

The VOF method is particularly attractive, since it can be constructed in the form of inherently mass conserving approach. However, when the volume fraction is abruptly-varying, the significant effort is required for accurate estimation of normal vector to interface and of interface curvature. Specific techniques based on least-squares-fit, height functions or reconstructed distance functions were developed to reconstruct these parameters from volume fractions (see *e.g.* Pilliod & Puckett, 2004; Cummins *et al.*, 2005; Francois *et al.*, 2006).

Another interface capturing method is the phase field method. The basic idea here is to introduce a conserved order parameter or phase-field, φ, to characterize two different phases. This order parameter changes rapidly but smoothly in the thin interfacial region, and it is mostly uniform in the bulk phases, with distinct values φ_+ and φ_-. The interfacial location is defined by the contour level $(\varphi_+ + \varphi_-)/2$. In this method, the interface dynamics is modeled by the advective Cahn-Hilliard equation for φ. This equation is

$$\frac{\partial \varphi}{\partial t} + \vec{u} \cdot \nabla \varphi = \nabla \cdot (M \nabla \mu_\varphi), \tag{1.20}$$

where $M(\varphi) > 0$ is a diffusion parameter, called the mobility. The chemical potential, μ, is the rate of change of free energy (which consists in contribution bulk and interface) with respect to φ and is given by

$$\mu_\varphi = \frac{d\psi}{d\varphi} + \varepsilon_\varphi^2 \nabla^2 \varphi, \tag{1.21}$$

$\psi(\varphi) = \varphi^2(1-\varphi)^2/4$ is the bulk energy, ε_φ is the capillary width, indicative for the thickness of diffuse interface. The Cahn number $Cn = \varepsilon_\varphi / L$ relates ε_φ to a characteristic macroscopic length L.

Since the level set method is our main framework in this study, its description thereafter is provided separately from approaches above, in the next Section. It is worthwhile to that the common advantage of interface capturing methods is their ability to simulate interface with complex topological changes, with merging and breakup. Each interface capturing method has its own advantages and drawbacks. In practice, these methods are often used in the hybrid form.

1.4 Level set method

1.4.1 Background

The level set method was introduced by Osher & Sethian (1988) as general techniques to capture the moving interface in numerous multi-physics problems (picturesque applications may be found on the web home page of Ron Fedkiw). For example, it has been used for Stefan problems to study crystal growth (Tan & Zabaras, 2007), to simulate water and fire interaction, to simulate the premixed flames by so-called G-equation approach, to perform computer graphics, etc. Fig. 1.5-1.7 illustrate examples of recent level set simulations of crystal growth, image recognition and two-phase flow, respectively. The profound introduction into level set methods and in its various applications is given in books by Sethian (1996); Osher & Fedkiw (2003); Giga (2006) and review articles by Osher & Fedkiw (2001) and Smereka & Sethian (2003).

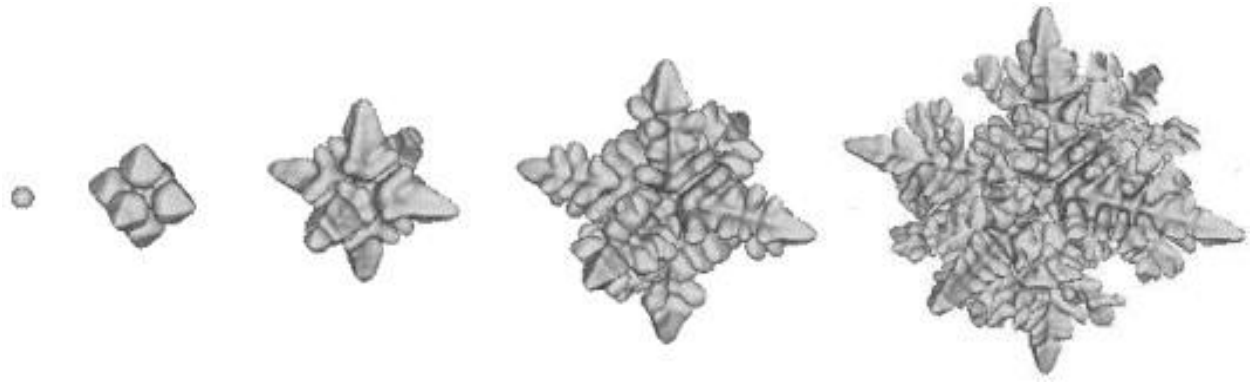
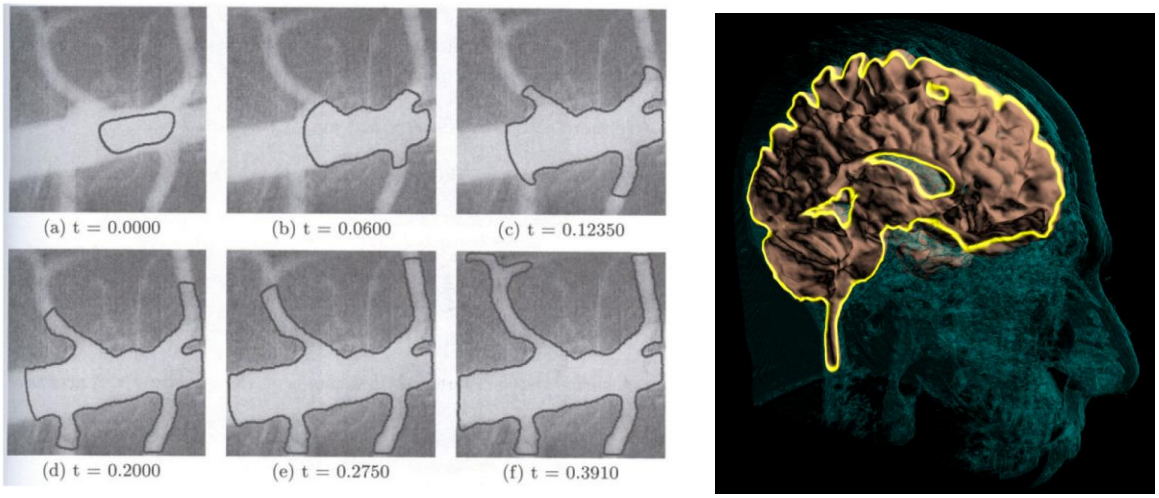


Figure 1.5: Level set simulation of crystal growth from Tan & Zabaras (2007).



(a) Malladi & Sethian (1995) (b) http://www.sci.utah.edu

Figure 1.6: Level set simulation for medical image segmentation.

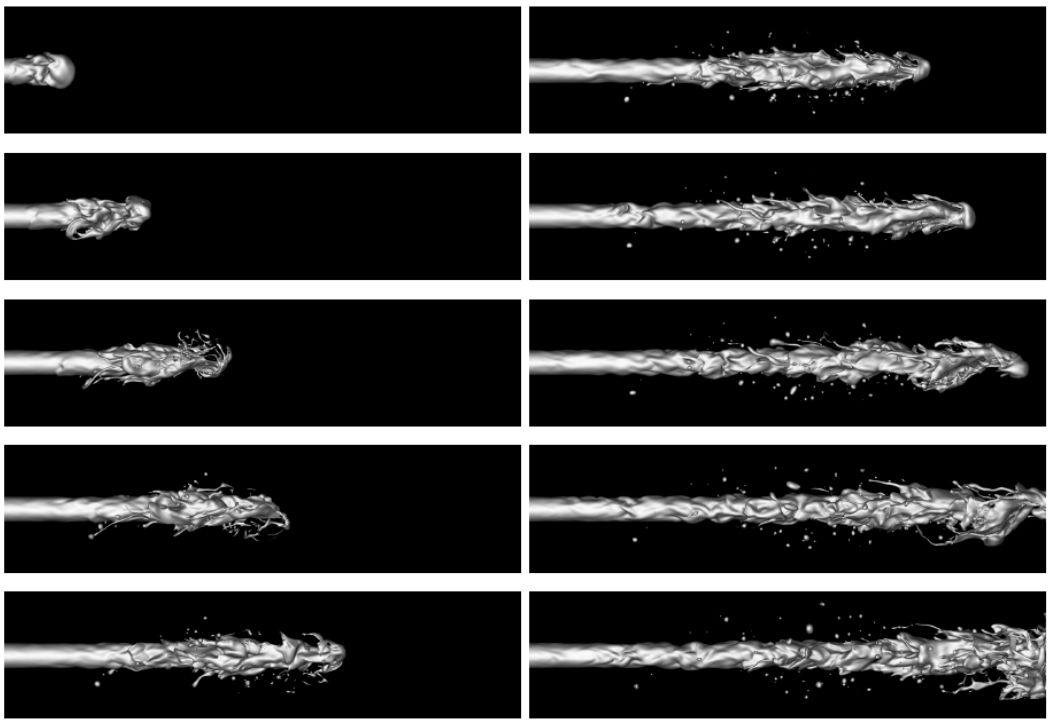


Figure 1.7: Level set simulation of primary atomization of diesel-type jet from Desjardins (2008).

The underlying idea of the level set method is to embed an interface Γ in R^n as the zero level set of a smooth scalar function $G: R^n \times R_+ \to R$, *i.e.*

$$\Gamma = \{\vec{x} \in \Omega:\ G(\vec{x},t) = 0\}. \tag{1.22}$$

The function G is defined everywhere in the domain Ω: it is positive in the region occupied by the fluid 1 and negative in the region occupied by the fluid 2. The evolution of G is given by the following transport equation:

$$\frac{\partial G}{\partial t} + \vec{u} \cdot \nabla G = 0. \tag{1.23}$$

This equation, with a given initial distribution of interface $G(\vec{x}, t=0) = G_0(\vec{x})$, is often referred to as the *level set equation*.

Away from the interface the level set scalar is assumed to be a signed distance function to the interface; *i.e.*, $G(\vec{x},t) = -d$ in Ω_1 and $G(\vec{x},t) = +d$ in Ω_2 where d is the shortest distance from the point $\vec{x}$ to the interface at given time t. Fig. 1.8 illustrates the level set function in two dimensions.

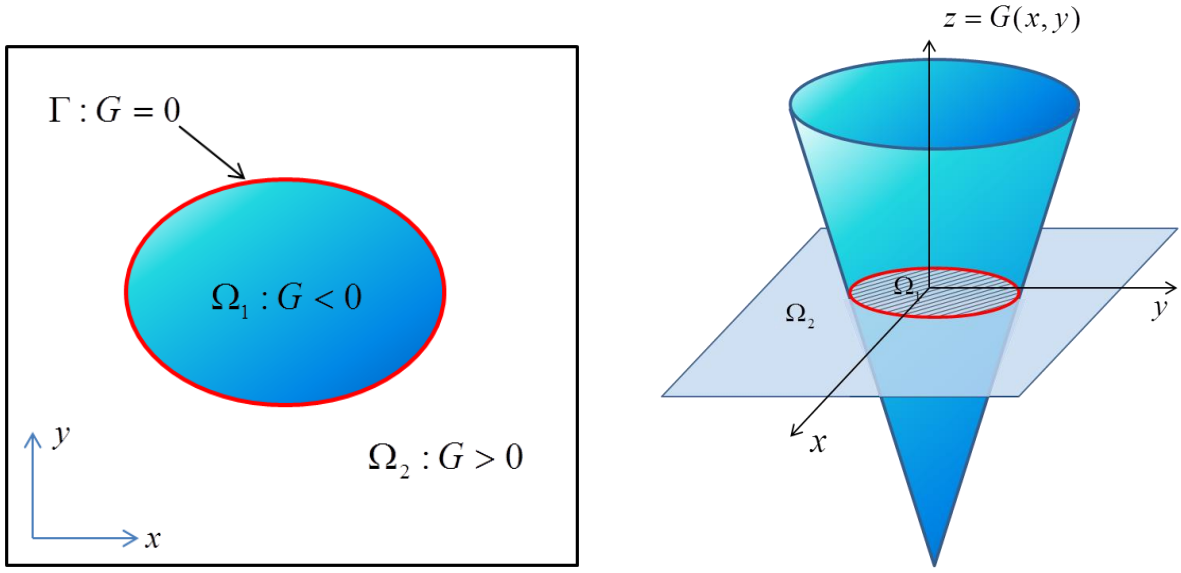


Figure 1.8: Representation of the interface Γ in 2D by signed distance level set function.

The unit normal on the interface can easily be expressed in terms of $G(x,t)$:

$$\vec{n} = -\frac{\nabla G}{|\nabla G|}. \tag{1.24}$$

The mean curvature of the interface is defined as the divergence of the normal

$$\kappa = \nabla \cdot \vec{n} = -\nabla \cdot \frac{\nabla G}{|\nabla G|}. \tag{1.25}$$

The volume (area in 2D case) enclosed by the zero level set isosurface can be integrated as follows:

$$V = \int_{\Omega} H(G) d\vec{x}, \tag{1.26}$$

where H is the Heaviside function.

Typically, approximation of jump conditions in physical properties by step change across the interface leads to poor computational results. Therefore, the properties near the interface are defined using the smoothed Heaviside kernel function, H_{ε}, given by:

$$H_{\varepsilon}(G) = \begin{cases} 0 & \text{if} \quad G < -\varepsilon, \\ \frac{1}{2}\left[1 + \frac{G}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi G}{\varepsilon}\right)\right] & \text{if} \quad |G| \le \varepsilon, \\ 1 & \text{if} \quad G > \varepsilon. \end{cases} \tag{1.27}$$

In the same way, the Dirac delta function in (1.18) is smeared-out according to:

$$\delta_{\varepsilon}(G) = \begin{cases} 0 & \text{if} \quad |G| > \varepsilon, \\ \frac{1}{2\varepsilon}\left[1 + \cos\left(\frac{\pi G}{\varepsilon}\right)\right] & \text{if} \quad |G| \le \varepsilon. \end{cases} \tag{1.28}$$

Here $\varepsilon = 1.5 \cdot \Delta x$ is the smoothing parameter, Δx is the grid size.

For what follows, let us write equations in dimensionless form. We define reference variables L, U for the length and velocity, respectively. The relevant dimensionless variables in the Navier-Stokes equations can be defined as

$$u_i^* = \frac{u_i}{U}, \quad x_i^* = \frac{x_i}{L}, \quad t^* = \frac{tU}{L},$$
$$p^* = \frac{p}{\rho U^2}, \quad \rho^* = \frac{\rho}{\rho_1}, \quad \mu^* = \frac{\mu}{\mu_1}, \tag{1.29}$$

where the superscript * denote dimensionless variables. In the non-dimensional variables, the continuity equation remains invariant, *i.e.*

$$\nabla \cdot \vec{u} = 0. \tag{1.30}$$

Substitution of variables (1.29) into (1.18) leads to:

$$\rho(G)\frac{\mathrm{D}\vec{u}}{\mathrm{D}t} = -\nabla p + \frac{1}{Re}\nabla \cdot (2\mu(G)D) - \frac{1}{We}\kappa(G)\delta(G)\nabla G + \frac{1}{Fr^2}\rho(G)\vec{e}_g, \tag{1.31}$$

where $\vec{e}_g$ is a unit vector aligned with the gravity force. For simplicity, the superscript $\star$ dropped here. The momentum equation (1.31) contains on the right hand side three dimensionless parameters known as:

the *Reynolds number*,

$$Re = \frac{\rho_1 UL}{\mu_1}, \tag{1.32}$$

the *Weber number*,

$$We = \frac{\rho_1 LU^2}{\sigma}, \tag{1.33}$$

and the *Froude number*,

$$Fr = \frac{U}{\sqrt{gL}}. \tag{1.34}$$

The Reynolds number expresses the ratio of inertia to viscous forces, the Weber number is the ratio of inertia forces to surface tension and the Froude number is the ratio of inertia to gravity forces.

The fluid properties are thus defined (in dimensionless form) by

$$\rho(G) = \lambda + (1-\lambda)H_\varepsilon(G) \quad \text{and} \quad \mu(G) = \eta + (1-\eta)H_\varepsilon(G), \tag{1.35}$$

where $\lambda = \rho_2 / \rho_1$ is the density ratio and $\eta = \mu_2 / \mu_1$ is the viscosity ratio.

1.4.2 Problems and recent developments

While exhibiting some advantages over alternative numerical approaches to capture the interface, level set methods have several difficulties.

(i) Mass conservation

The drawback of Level set methods is that they are not volume (or mass) conserving for divergence free velocity fields. Consequently, the volume enclosed by zero iso-surface of the level set scalar is not conserved. In order to improve the mass conservation property of the level set method, several hybrid approaches have been proposed. Enright *et al.* (2002) proposed a particle level set method (PLS), where Lagrangian markers are employed to correct the front location predicted by Eulerian transport. Sussman & Puckett (2000) proposed to couple the level set method with the VOF approach (CLSVOF), hence realizing benefits from both approaches, the mass conservation property of VOF and the smooth interface description of the level set method. While both these hybrids schemes have been successful in applications, another problem is that their cost is typically much greater than the cost of a simple level set method. The reason is as follows: in PLS approach, many particles per cell are required to get accurate solution; in the CLSVOF method, there are restrictions on the time step in the VOF scalar transport. Moreover, due to complexity of hybrid schemes, their realization becomes complicated, and requires many additional efforts compared to original approaches. Another way to cope with the mass conservation problem in level set methods concerns the mesh refinement around the interface (Gomez *et al*., 2005; Herrmann, 2008), with target on decrease of numerical errors. In order to reduce the mass conservation errors, Olsson & Kreiss (2005) proposed a simple modification of the level set method, known as conservative level set (CLS) method. In the meantime, they retained the simplicity of the original method. These Authors replaced the usual signed distance function of the classical level set approach by the hyperbolic tangent profile which is transported and re-initialized using conservative equations. They showed that the mass conservation error is possible to reduce by an order of magnitude in comparison with the results based on the signed distance function. In Olsson *et al.* (2007); Desjardins *et al.* (2008) CLS method was further developed and assessed. However, realization of hybrid schemes becomes complicated, and requires many additional efforts compared to original approaches.

(ii) Re-initialization

The well-known problem addressed to (1.23) is this: if the flow velocity is not constant, the level set function G may become strongly distorted: which means that its gradient may become very large or very small around the interface. For illustration of this, it is easy to show that if G is the level set function governed by (1.23), then the squared norm of the level set gradient changes with time by the following equation:

$$\frac{\mathrm{D}}{\mathrm{D}t}\left|\nabla G\right|^2 = -2S_{ij}\frac{\partial G}{\partial x_i}\frac{\partial G}{\partial x_j}, \tag{1.36}$$

where $S_{ij} = \dfrac{\partial u_i}{\partial x_j}$ are the components of the velocity gradient tensor. Here we used the summation convention on the repeated suffix.

For example, if we take the velocity field in the form of $S_{ij} = S(t)\delta_{ij}$, where δ_{ij} is the Kronecker symbol, then the solution of (1.36) is the following:

$$\left|\nabla G(\vec{x},t)\right| = e^{-\int_0^t S(\xi)d\xi}. \tag{1.37}$$

We see that the gradient grows exponentially with time (for negative $S(t)$) or decays exponentially for positive $S(t)$. In such cases the numerical integration of (1.23) may suffer from loss of accuracy. In level set methods, this problem is remedied by the *re-initialization* procedure (Chopp, 1993), *i.e.* by reconstruction of the level set function in a way to satisfy the eikonal equation:

$$\left|\nabla G(\vec{x},t)\right| = 1. \tag{1.38}$$

The solution of the eikonal equation (Arnold, 1983) is given by a signed distance function with respect to the zero level set. However, strictly from the mathematical viewpoint, equations (1.23) and (1.38) are incompatible (*e.g.* Gomes & Faugeras, 2000): equation (1.23) is evolutional equation whereas equation (1.38) is not. It is clearly seen by comparing (1.36) with equation (1.38). As a consequence, even if the initial G-function is defined as a signed distance function, the solution of (1.23) does not retain this property in the general case. Therefore, in repairing the level set function by equation (1.38), the common practice is to use (1.38) in its evolving form through an iterative process (*e.g.* Sussman *et al.*, 1994):

$$\frac{\partial G^{\nu}}{\partial t'} + S(G)\left(\left|\nabla G^{\nu}\right| - 1\right) = 0, \tag{1.39}$$

with initial condition

$$G^{\nu}(\vec{x}, t' = 0) = G(\vec{x}, t). \tag{1.40}$$

Here G^{ν} is a scalar that represents the corrected distance field, t' is the pseudo-time, $S(G)$ is a smoothed sign function which is given by

$$S(G) = \frac{G}{\sqrt{G^2 + \varepsilon^2}}, \tag{1.41}$$

where ε is a small parameter to avoid singularity[1].

Analytically, it is stated that in the limit $t' \to \infty$, the solution of (1.39) tends to the unique viscosity solution of (1.38) without perturbation of zero level set. However from numerical experience, it has been observed (Russo & Smereka, 2000) that after several iterations in (1.39), the zero level set may move towards nearest grid points which will not lie directly on the interface. This may incur errors into solution of (1.23). Along with additional modifications to (1.39) proposed in Sussman *et al.* (1998); Sussman & Fatemi (1999); Cheng & Tsai (2008); Hartmann *et al.* (2008, 2010), an alternative approach is to circumvent iterative re-initialization (1.39). One way is as follows. The characteristics of the eikonal equation represent straight lines, normal to the interface, and the propagation speed of level sets along these lines is given by $u_k n_k$ (here $k = 1,2,3$; components n_i are the direction cosines of the normal vector to interface, and, as hereafter, the summation convention on the repeated suffix is used). Then introducing temporary signed distance function $G^{temp}(\vec{x},t)$ in respect to the eikonal equation, $\left|\nabla G^{temp}\right| = 1$, *i.e.* with preserved distances between level sets, it is clear that propagation speed $F^{ext} = u_k n_k$ of all such temporary functions does not change along the characteristics:

$$F^{ext} = u_k n_k = \left(u_k n_k\right)_{n=0}. \tag{1.42}$$

Here the propagation speed F^{ext} is known as the "extension velocity", and n is the distance from the interface along the characteristics. Then using definition of the normal derivative $\frac{\partial}{\partial n} = n_j \frac{\partial}{\partial x_j}$, where, in respect to the eikonal equation, $n_j = -\frac{\partial G^{temp}}{\partial x_j}$, one sees that (1.42) gives an equation for reconstruction of G^{temp} and F^{ext}:

$$\frac{\partial F^{ext}}{\partial n} = \frac{\partial G^{temp}}{\partial x_j} \cdot \frac{\partial F^{ext}}{\partial x_j} = 0. \tag{1.43}$$

On the interface at a given time, $G^{temp}\Big|_{n=0}$ matches exactly the zero level set $G\big|_{n=0}$, and F^{ext} equals the speed of interface in the normal direction. Hence, when the constructed from (1.43) extension velocity is used to advance the level set function $G(\vec{x},t)$ by solving

[1] An example in Appendix A serves to demonstrate the re-initialization procedure in the simple case of one-dimensional flow produced by homogeneous strain

$\frac{\partial G}{\partial t} + F^{ext}\left|\nabla G\right| = 0$, the zero-level set is mapped in respect to the eikonal equation; thereby $G(\vec{x},t)$ is never re-initialized. This time-marching process was developed in Adalsteinsson & Sethian (1999) (see also Zhao *et al*., 1996). However it has been mentioned in Chopp (2009) that in complex flows, the cost of determining the extension velocity becomes computationally expensive, and in some cases, the time-marching method of Adalsteinsson & Sethian (1999) can lead to unexpected behavior. The further development of the extension velocity method was introduced and assessed in Chopp (2009), with numerical scheme of simultaneous integration of (1.43) and (1.23).

1.4.3 Our motivation

In our work we propose an alternative approach, in which the intermediate step (1.43) is circumvented. We modify directly the level set equation by embedding a source term. The exact expression of this term is such that the eikonal equation is automatically satisfied. Furthermore on the interface, this term is equal to zero. Thereby integrating this new form of the level set equation, there is no more necessity in re-initialization of the level set function, as in the extension velocity method. In the meantime, the advantage of our approach is this: the exact expression of the source term allows for the possibility of derivation of its local approximate forms, of first-and-higher order accuracy. Compared to the extension velocity method, this may open the simplifications in realization of level set methods. Compared to the standard approach with re-initialization procedure, this may give the economies in the number of level set re-initializations, and also, due to reduced number of re-initializations, one may expect an improvement in resolution of zero-set level. Hence, the objective of the present work is to describe and to assess this approach in different test cases.

1.5 Scope and Presentation

First, in Chapter 2 we present a numerical method for solving incompressible Navier-Stokes equations, which we applied in this work, coded and validated. Two-dimensional test problems were assessed with analysis of accuracy, and with comparison of obtained results with those found in the literature. In Chapter 3 we present the modified level set equation with embedded source term in a way that the eikonal equation is satisfied, and the evolution of the zero level set remains the same as for standard level set equation. Different approximate forms for the source term coefficient are derived. Relation with the extension velocity approach is presented. Chapter 4 is devoted to assessment of the modified level set equation. In this Chapter, we used one- and two-dimensional different test cases. Results of comparison with standard approach with and without re-initializations of level sets, as well as with analytical solutions are methodologically

discussed. Finally, in Chapter 5, we give several examples of computation and analysis of interfacial flows. The manuscript is ended up by conclusion, with perspectives of use of the modified level set equation for modeling free flows with interface. The manuscript is completed by three Appendixes.

Chapter 2

Numerical schemes for incompressible Navier-Stokes equations

In this chapter we present the numerical method for solving two-dimensional incompressible Navier-Stokes equations. Our numerical algorithm is based on the scheme of Zhang & Jackson (2009). It represents a high-order incompressible flow algorithm based on the projection method and Weighted Essentially Non-Oscillatory (WENO) finite differences. The algorithm of Zhang & Jackson (2009) was successfully applied for three-dimensional LES (large eddy simulation) of turbulent channel flow by Shetty *et al*. (2010). Note that very similar algorithms were developed in Ph.D. theses of Tanguy (2004) and Couderc (2007) for simulation of complex two-phase phenomena, such as turbulent air-blast atomization. In order to increase the accuracy, we introduced in our work two modifications in the method of Zhang & Jackson (2009). First, we implement low dissipative WENO-Z scheme of Borges *et al.* (2008) and bandwidth optimized WENO-SYM scheme of Martin *et al.* (2006), instead of classical WENO scheme of Shu & Jiang (1996). Second, we used the high-order interpolation scheme in order to determine velocity on the staggered grid, instead of the standard linear interpolation. Our algorithm is validated on different test problems.

For purposes of this Chapter, let us remind the basic equations for viscous incompressible flows:

$$\begin{cases} \dfrac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\dfrac{1}{\rho} \nabla p + \dfrac{1}{\rho} \nabla \cdot (2\mu D) + \vec{F}, \\ \nabla \cdot \vec{u} = 0; \end{cases} \tag{2.1}$$

where p is the pressure, $\vec{u} = (u, \mathrm{v})^T$ is the velocity field, ρ is the density, μ is the viscosity and D is the deformation rate tensor, which is the following in 2D case:

$$D = \frac{1}{2}(\nabla\vec{u} + \nabla\vec{u}^T) = \frac{1}{2}\begin{pmatrix} 2u_x & u_y + \mathrm{v}_x \\ u_y + \mathrm{v}_x & 2\mathrm{v}_y \end{pmatrix} \tag{2.2}$$

$\vec{F} = (F_x, F_y)$ is the volume force, such as gravity force, $\vec{F} = (0, -g)$, or $\vec{F} = \sigma\kappa\delta(G)\nabla G$ represents the surface tension force in case of two-phase flow. Note that in the latter case the density and the viscosity are variable coefficients, which are given by (1.35). For one-phase flows ρ and μ are constant in the whole domain.

2.1 Projection method

The main difficulty in obtaining a time-accurate solution for incompressible Navier-Stokes equations arises from the fact that continuity equation does not contain the time derivative explicitly. The constraint of mass conservation is achieved by implicit coupling between the continuity equation and the pressure field in the momentum equations. Numerically, this difficulty can be circumvented by using the *artificial compressibility method* of Chorin (1967), which introduces a time derivative of pressure in the continuity equation:

$$\frac{1}{\beta}\frac{\partial p}{\partial t} + \nabla\cdot\vec{u} = 0, \tag{2.3}$$

where β is the artificial compressibility parameter that needs to be properly chosen in order to achieve numerical convergence. The WENO scheme was also incorporated by means of artificial compressibility in Chen *et al.* (1999); Wu (2007). Although it improved the accuracy in the spatial discretization, this scheme still suffers from the stiffness of the system when the artificial compressibility has to be minimized. In addition, the usual question raises: what is the best choice of the tunable artificial compressibility parameter.

Another class of the numerical methods is based on *projection methods*. The projection method was originally introduced by Chorin (1968) and independently by Temam (1968) as an efficient method of solving the incompressible Navier-Stokes equations. The theoretical background of projection type method is the decomposition theorem of Ladyzhenskaya, sometimes referred to as Helmholtz-Hodge decomposition. It states that the vector field $\vec{u}$, defined on a simply connected domain, can be uniquely decomposed into a divergence-free (solenoidal) part and an irrotational part:

$$\vec{u} = \vec{u}_{sol} + \nabla G, \tag{2.4}$$

where G is some scalar function. Let us show how the set of equations (2.1) can be integrated from time t^n to time t^{n+1} using the projection method. Typically, the algorithm consists of two

stages. In the first stage (velocity *predictor*), an intermediate velocity $\vec{u}^* = (u^*, \mathrm{v}^*)^T$ that does not satisfy the incompressibility constraint is computed:

$$\frac{\vec{u}^* - \vec{u}^n}{\Delta t^n} + (\vec{u}^n \cdot \nabla)\vec{u}^n = \frac{\nabla \cdot (2\mu D)^n}{\rho} + \vec{F}^n, \tag{2.5}$$

where $\vec{u}^n$ is the velocity at n^{th} time level. In the second step, the pressure is used to project the intermediate velocity onto a space of divergence-free velocity field. Then the velocity field at the new time step, $\vec{u}^{n+1} = (u^{n+1}, \mathrm{v}^{n+1})^T$, is given by (velocity *corrector*):

$$\frac{\vec{u}^{n+1} - \vec{u}^*}{\Delta t^n} = -\frac{\nabla p^{n+1/2}}{\rho}, \tag{2.6}$$

Computing the right-hand side of the above equation requires knowledge of the pressure $p^{n+1/2}$. Taking the divergence of (2.6), with satisfaction of divergence free condition for $\vec{u}^{n+1}$, one comes to the Poisson equation for pressure:

$$\nabla \cdot \left(\frac{\nabla p^{n+1/2}}{\rho} \right) = \frac{\nabla \cdot \vec{u}^*}{\Delta t^n}. \tag{2.7}$$

It is worthwhile to note that equation (2.6) may be written in the following form

$$\vec{u}^* = \vec{u}^{n+1} + \frac{\Delta t}{\rho} \nabla p^{n+1/2}, \tag{2.8}$$

which is the standard Hodge decomposition for vector field $\vec{u}^*$.

In the Chorin's original version of the projection method, all spatial gradients in (2.5) - (2.7) were approximated by second central difference schemes. In this work we use high-order projection method by Zhang & Jackson (2009). Details of the spatial discretization and time integration scheme will be given in the next sections.

2.2 Variables arrangement

We use a finite difference approximation for governing equations. A computational domain is a rectangle with sizes L_x and L_y in x and y direction respectively. A variant of the staggered Cartesian grid of Harlow & Welch (1965) is used as depicted in Fig. 2.1. The cell corners (x_i, y_j) (grid points) are represented by circles. The centers of the cell faces, $(x_{i+1/2}, y_j)$ and $(x_i, y_{j+1/2})$, are denoted by triangles in this figure. The pressure variable is defined at the corners

of each cell, and the velocity variables on the faces, which are parallel to their directions, u and v, are defined on horizontal and vertical edges of the cell in two dimensions, respectively.

Let Ω_{ij} be a particular cell of the computational grid. Then the following relations are valid:

$$\Omega_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}], \qquad 1 \le i \le N_x, 1 \le j \le N_y, \tag{2.9}$$

where the cell corners and the spatial increments are defined by

$$\begin{aligned} x_i &= (i-1)\Delta x, \quad \Delta x = L_x / (N_x - 1), \\ y_j &= (j-1)\Delta y, \quad \Delta y = L_y / (N_y - 1), \end{aligned} \tag{2.10}$$

and the cell faces points are defined by

$$x_{i+1/2} = \frac{1}{2}(x_i + x_{i+1}), \quad y_{j+1/2} = \frac{1}{2}(y_j + y_{j+1}). \tag{2.11}$$

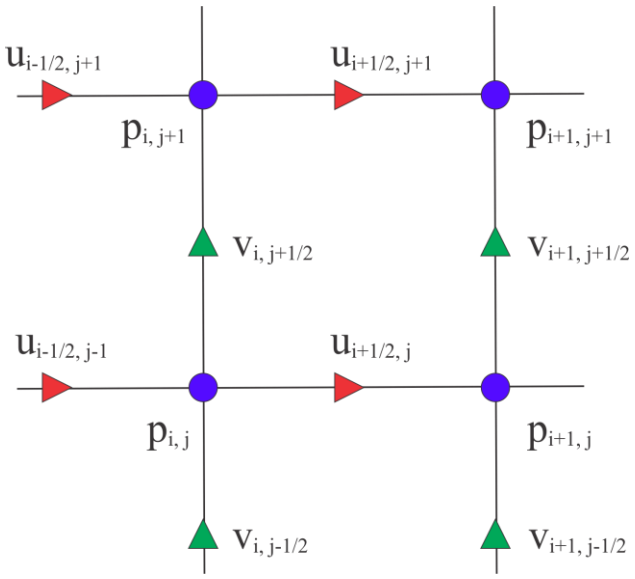


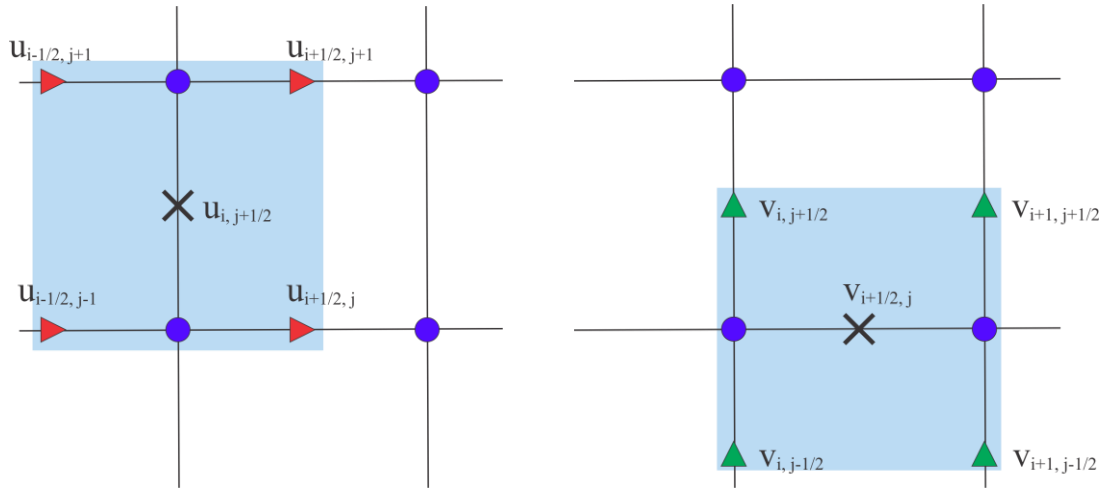
Figure 2.1: Schematic of the staggered grid in two dimensions.

With the staggered grid, the momentum equations are evaluated at velocity nodes, and the continuity equation is enforced in each cell. A main advantage of the staggered arrangement is the strong coupling between the velocities and pressure. This alleviates the convergence problem and oscillations in pressure and velocity fields, occurring in the non-staggered discretization (Ferziger & Peric, 2002). Another advantage of this particular grid arrangement is that the

tangential velocity on the boundary can be imposed directly (Zhang & Jackson, 2009). On the other hand, a disadvantage of the staggered grid is that the additional interpolation is required to determine unknown values of the velocity components on cell faces, $u_{i,j+1/2}$ and $v_{i+1/2,j}$. The standard approach of finding these unknown values is the simple averaging (bilinear interpolation) of velocities from neighbor grid points (see Fig. 2.2):

$$u_{i,j+1/2} = \frac{u_{i+1/2,j} + u_{i-1/2,j} + u_{i+1/2,j+1} + u_{i-1/2,j+1}}{4},$$
$$v_{i+1/2,j} = \frac{v_{i,j+1/2} + v_{i,j-1/2} + v_{i+1,j+1/2} + v_{i+1,j-1/2}}{4}. \tag{2.12}$$

However, interpolation procedure (2.12) has the second-order spatial accuracy only. In this work, we use a higher-order WENO interpolation scheme, which leads to 5^{th}-order spatial accuracy. Details of the interpolation scheme are given in Appendix C.



(a) (b)

Figure 2.2: Schematic of linear interpolation of the velocity on the staggered grid: (a) for u-component, (b) for v-component.

2.3 WENO schemes

In comparison to low-order numerical methods, the high-order methods (third-order and higher) offer an enhanced efficiency in the flow resolution, especially in multidimensional problem and for long time of integration. Recently, Zhang & Jackson (2009) proposed a high-order scheme for integration of the incompressible Navier-Stokes equations. Motivation was to handle the sharp gradients. This method incorporates the projection method and 5th-order WENO scheme (Jiang & Shu, 1996), applied for approximation of convective terms. The classic WENO scheme, developed for flows with shock waves, is numerically dissipative. The higher-order hybrid methods (Pirozzoli, 2002; Deng & Zhang, 2000) or modified WENO formulations, for example optimized WENO scheme (Martin *et al.*, 2006) and WENO-Z scheme (Borges *et al.*, 2008), can decrease such a dissipation.

2.3.1 WENO reconstruction

Without loss of generality, our description is focused on one-dimensional scalar law:

$$\frac{\partial u}{\partial t}+\frac{\partial f}{\partial x}=0, \tag{2.13}$$

where $f=f(u)$ is the flux function. First, following Shu & Osher (1988), the numerical flux function $h(x)$ is determined by

$$f(x)=\frac{1}{\Delta x}\int_{x-\Delta x}^{x+\Delta x} h(\xi)d\xi. \tag{2.14}$$

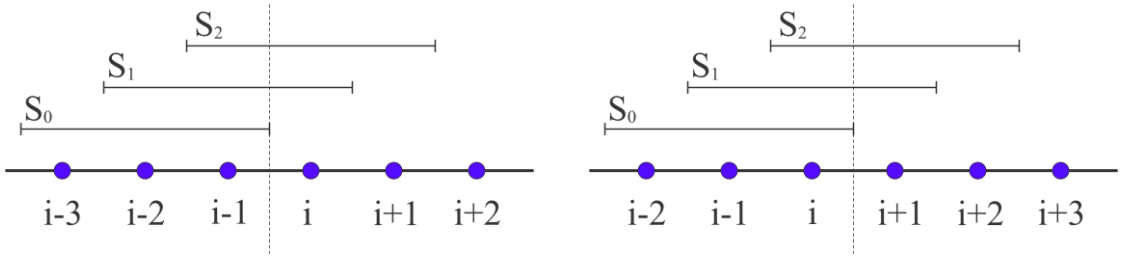
Equation (2.14) defines $h(x)$ implicitly in such way that

$$\frac{\partial u_i}{\partial t}=-\frac{h_{i+1/2}-h_{i-1/2}}{\Delta x}. \tag{2.15}$$

The conservative numerical schemes can be formulated by approximating $h(x)$ in (2.15). These approximations of $h(x)$ are denoted by $\hat{f}(x)$, and they are constructed using a polynomial form with undetermined coefficients. Substitution of this polynomial into (2.14) leads to the system of equations where the flux is a known quantity at the nodes surrounding the interface of interest. This allows the unique set of coefficients to be found. Then the spatial derivative in (2.13) is approximated by $\hat{f}(x)$ obtained:

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_i} \approx \frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{\Delta x}. \tag{2.16}$$

Fifth-order WENO schemes are based on the stencils displayed in Fig. 2.3. The numerical flux, $\hat{f}^k_{i+1/2} = h_{i+1/2} + O(\Delta x^3)$ with $k \in \{0,1,2\}$, is calculated for each of the three point stencils. The $\hat{f}^k_{i+1/2}$ from these stencils are then combined in a weighted average in a way to get maximal order of convergence. In the flow regions containing discontinuities, the weights are assigned such that the solution is essentially non-oscillatory.



(a) (b)

Figure 2.3: Discretization stencils for the standard five-order WENO scheme: (a) left-biased derivative, (b) right-biased derivative.

The numerical flux is now calculated according to

$$h_{i+1/2} \approx \hat{f}_{i+1/2} = \sum_{k=0}^{2} \omega_k \hat{f}^k_{i+1/2}. \tag{2.17}$$

where $\hat{f}^k_{i+1/2}$ are r^{th}-order accurate polynomial interpolants evaluated at $x_{i+1/2}$, and computed by

$$\hat{f}^k_{i+1/2} = \sum_{l=0}^{2} a_{kl} f_{i-2+k+l}. \tag{2.18}$$

Here a_{kl} are the stencil coefficients. For $k = 0,1,2$, we have

$$\begin{aligned}
\hat{f}^0_{i+1/2} &= \frac{1}{3} f_{i-2} + \frac{7}{6} f_{i-1} + \frac{11}{6} f_i, \\
\hat{f}^1_{i+1/2} &= \frac{1}{3} f_{i-2} + \frac{7}{6} f_{i-1} + \frac{11}{6} f_i, \\
\hat{f}^2_{i+1/2} &= \frac{1}{3} f_{i-2} + \frac{7}{6} f_{i-1} + \frac{11}{6} f_i,
\end{aligned} \tag{2.19}$$

where weights ω_k in (2.17) are defined by

$$\begin{aligned}
\alpha_1 &= \frac{d_1}{(\varepsilon + IS_1)^2}, \quad \omega_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \\
\alpha_2 &= \frac{d_2}{(\varepsilon + IS_2)^2}, \quad \omega_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}, \\
\alpha_3 &= \frac{d_3}{(\varepsilon + IS_3)^2}, \quad \omega_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}.
\end{aligned} \tag{2.20}$$

In these expressions, coefficients $d_0 = 0.3$, $d_1 = 0.6$, $d_2 = 0.1$ are called ideal weights or optimal weights, since they generate the central upstream 5$^{\text{th}}$-order scheme for 5-points stencil S_5. We refer α_k to as the non-normalized weights. The parameter ε is set to 10^{-6} in order to avoid division by zero in the calculations of the non-linear weights ω_k. The smoothness indicators IS_k measure the regularity of the k^{th} polynomial approximation $\hat{f}^k(x_i)$ at the stencil S_k, and they are given by

$$IS_k = \sum_{l=1}^{2} \Delta x^{2l-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \left(\frac{d^l}{dx^l} \hat{f}^k(x) \right)^2 dx. \tag{2.21}$$

In terms of discretized $f(x)$, *i.e.* f_i, the expression of IS_k are given by

$$\begin{aligned}
IS_0 &= \frac{13}{12}(f_i - 2f_{i+1} + f_{i+2})^2 + \frac{1}{4}(f_i - 4f_{i+1} + 3f_{i+2})^2, \\
IS_1 &= \frac{13}{12}(f_{i-1} - 2f_i + f_{i+1})^2 + \frac{1}{4}(f_{i-1} - f_{i+1})^2, \\
IS_2 &= \frac{13}{12}(f_{i-2} - 2f_{i-1} + f_i)^2 + \frac{1}{4}(3f_{i-2} - 4f_{i-1} + f_i)^2,
\end{aligned} \tag{2.22}$$

In general sense, definition (2.20) of weights is that on smooth parts of the solution, the smoothness indicators IS_k are all small, thereby generated weights ω_k become close to ideal weights d_k. On the other hand, if a given stencil S_k contains a discontinuity, then $IS_k \simeq O(1)$,

and the corresponding non-linear weight ω_k is small relatively to other weight corresponding to another k.

2.3.2 Order of convergence at critical points

In Henrick *et al.* (2005), truncation error analysis of the finite difference approximation (2.17) led to necessary and sufficient conditions on the weights ω_k for the WENO scheme in order to achieve the formal fifth-order $O(\Delta x^5)$ convergence at smooth parts of the solution. The Authors have also found that at first-order critical points x_c, points where the first derivative of the solution vanishes ($f'(x_c)=0$), convergence degraded to only third-order $O(\Delta x^3)$. The demarche of Authors was as follows. Adding and subtracting $\sum_{k=0}^{2} d_k \hat{f}_{i\pm1/2}^{k}$ to (2.17), leads to:

$$\begin{aligned}\hat{f}_{i\pm1/2} &= \sum_{k=0}^{2} d_k \hat{f}_{i\pm1/2}^{k} + \sum_{k=0}^{2} (\omega_k^{\pm} - d_k)\hat{f}_{i\pm1/2}^{k} = \\ &\left[h_{i\pm1/2} + B^{\pm}\Delta x^5 + O(\Delta x^6)\right] + \sum_{k=0}^{2} (\omega_k^{\pm} - d_k)\hat{f}_{i\pm1/2}^{k}.\end{aligned} \tag{2.23}$$

Here, the superscripts $\pm$ correspond to the $\pm$ in $\hat{f}_{i\pm1/2}$. Including the next term in the Taylor expansion $\hat{f}_{i+1/2}^{k} = h_{i+1/2} + O(\Delta x^3)$, Henrick *et al.* (2005) have written:

$$\begin{aligned}\sum_{k=0}^{2} (\omega_k^{\pm} - d_k)\hat{f}_{i\pm1/2}^{k} &= \sum_{k=0}^{2} (\omega_k^{\pm} - d_k)\left(h_{i\pm1/2} + A_k\Delta x^3 + O(\Delta x^4)\right) = \\ &h_{i\pm1/2}\sum_{k=0}^{2} (\omega_k^{\pm} - d_k) + \Delta x^3 \sum_{k=0}^{2} A_k(\omega_k^{\pm} - d_k) + \sum_{k=0}^{2} A_k(\omega_k^{\pm} - d_k)O(\Delta x^4).\end{aligned} \tag{2.24}$$

Substituting this result at a finite difference formula (2.23) for the polynomial approximation $\hat{f}_{i\pm1/2}$, Henrick *et al.* (2005) obtained:

$$\begin{aligned}\frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{\Delta x} &= \frac{h_{i+1/2} - h_{i-1/2}}{\Delta x} + O(\Delta x^5) + \frac{\sum_{k=0}^{2}(\omega_k^{+} - d_k)\hat{f}_{i+1/2}^{k} - \sum_{k=0}^{2}(\omega_k^{-} - d_k)\hat{f}_{i-1/2}^{k}}{\Delta x} = \\ &= f'(x_i) + O(\Delta x^5) + \frac{h_{i+1/2}\sum_{k=0}^{2}(\omega_k^{+} - d_k) - h_{i-1/2}\sum_{k=0}^{2}(\omega_k^{-} - d_k)}{\Delta x} + \\ &\Delta x^2 \sum_{k=0}^{2} A_k(\omega_k^{+} - \omega_k^{-}) + \left[\sum_{k=0}^{2}(\omega_k^{+} - d_k) - \sum_{k=0}^{2}(\omega_k^{-} - d_k)\right]O(\Delta x^3).\end{aligned} \tag{2.25}$$

The $O(\Delta x^5)$ term remains after division by Δx because $B^+ = B^-$ in (2.23). Thus, necessary and sufficient conditions for fifth-order convergence are given by

$$\sum_{k=0}^{2} A_k(\omega_k^{\pm} - d_k) = O(\Delta x^{p_1}), \qquad \omega_k^{\pm} - d_k = O(\Delta x^{p_2}). \tag{2.26}$$

The order of the WENO scheme is given by

$$p = \min(5, p_1 + 2, p_2 + 3) \tag{2.27}$$

For fifth-order accuracy, it is necessary $p_1 = 3,\ p_2 = 2$.

2.3.3 WENO-Z scheme

Borges *et al.* (2008) derived a new set of WENO weights ω_k that satisfies the necessary and sufficient conditions (2.26) for fifth-order convergence. The idea was to use the whole 5-points stencil S_5 (see Fig. 2.3) in order to introduce a new smoothness indicator of a higher order than the classical smoothness indicators. Denoting it by IS_5, and it is given by

$$IS_5 = \left| IS_0 - IS_2 \right| \tag{2.28}$$

Then Borges *et al.* (2008) defined the new smoothness indicators IS_k^z in the following form:

$$IS_k^z = \frac{IS_k + \varepsilon}{IS_k + IS_5 + \varepsilon}, \quad k = 0,1,2 \tag{2.29}$$

with the new WENO weights ω_k^z correspondingly:

$$\omega_k^z = \frac{\alpha_k^z}{\alpha_k^z + \alpha_k^z + \alpha_k^z}, \quad \alpha_k^z = \frac{d_k}{\beta_k^z} = d_k \left(1 + \left(\frac{IS_5}{IS_k + \varepsilon} \right)^q \right). \tag{2.30}$$

When $q = 2$, conditions (2.26) are satisfied and the fifth order is achieved.

2.3.4 Bandwidth-optimized WENO-SYM scheme

Martin *et al.* (2006) discussed the dispersion and dissipation properties of WENO schemes. First let us interpret the numerical dispersion and dissipation in a simple way, as follows.

The bandwidth properties of linear numerical schemes are determined by Fourier analysis. Consider a pure harmonic function

$$f(x) = e^{ikx}. \tag{2.31}$$

Then exact spatial derivative of $f(x)$ is given by

$$f'(x) = ike^{ikx}. \tag{2.32}$$

First derivative of $f(x)$ obtained from the finite difference scheme can be expressed as

$$f'(x) = ik^* f(x), \tag{2.33}$$

where $k^* = k^*(k)$ is the modified wavenumber. The real part of k^* is responsible for phase errors; the imaginary part of k^* describes dissipative amplitude errors. It is easy to show that any central scheme has zero dissipation. In practice the scaled modified wavenumber is used in analysis: $\tilde{k}^* = k^* \Delta x$. For example, consider the standard second order central difference scheme. We have

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_j} = \frac{f_{j+1} - f_{j-1}}{2\Delta x} = \frac{f_j e^{ik\Delta x} - f_j e^{-ik\Delta x}}{2\Delta x} = \frac{i\sin(k\Delta x)}{\Delta x} f_j, \tag{2.34}$$

therefore, the modified wavenumber in this case is defined by

$$\tilde{k}^*(k) = \sin(k\Delta x). \tag{2.35}$$

In similar way, the modified wave number for WENO scheme given by (2.17) can be obtained and it is the following:

$$\tilde{k}^*(k) = -i(1 - e^{-ik}) \sum_{m,l} d_m a_{ml} e^{i(-2+m+l)k}. \tag{2.36}$$

Here, the expression of $\tilde{k}^*$ contains the stencil coefficients and optimal weights of WENO scheme as free parameters. One may design bandwidth-optimized optimal weights d_k that maximize Lele's bandwidth resolving efficiency index. The bandwidth optimization is expressed as the minimization of an integrated error function. Following Martin *et al*. (2006), the error function used is

$$I = \int_0^{\pi} e^{\nu(\pi - k)} \left(\sigma \left[\mathrm{Re}(\tilde{k}^*) - k \right]^2 + (1 - \sigma) \left[\mathrm{Im}(\tilde{k}^*) - \gamma \sin^{\mu}(k/2) \right]^2 \right) dk. \tag{2.37}$$

The optimal weights d_k, contained in this function in k^*, are free parameters. Martin *et al.* (2006) have chosen to define the stencil coefficients a_{kl}, also embedded in the modified wavenumber, completely through the order of accuracy constraints. Taking the following values of free parameters $\sigma = 1/2$, $\mu = 16$, $\nu = 6$, $\gamma = -1$, the integral in (2.37) has been minimized. In so doing, they specified the resulting WENO schemes of the third-order accuracy with optimal weights: $d_0 = 0.094647545896$, $d_1 = 0.428074212384$, $d_2 = 0.408289331408$ and $d_3 = 0.068988910311$. Note, that in WENO-SYM scheme there is an additional weight d_3 corresponding to additional candidate stencil S_3, which provides symmetric discretization for $\hat{f}_{i+1/2}$.

Fig. 2.4 displays the bandwidth properties for the WENO-SYM and the original WENO-JS with optimal weights. This figure we completed by results from the second-order central scheme, from the first order forward scheme, and from six-order Padé scheme. As expected, the numerical dispersion and dissipation are minimal when the Padé scheme is used. For what follows, it is worthwhile to note that dissipation error is much less in the case of WENO-SYM than in the case when WENO-JS is used.

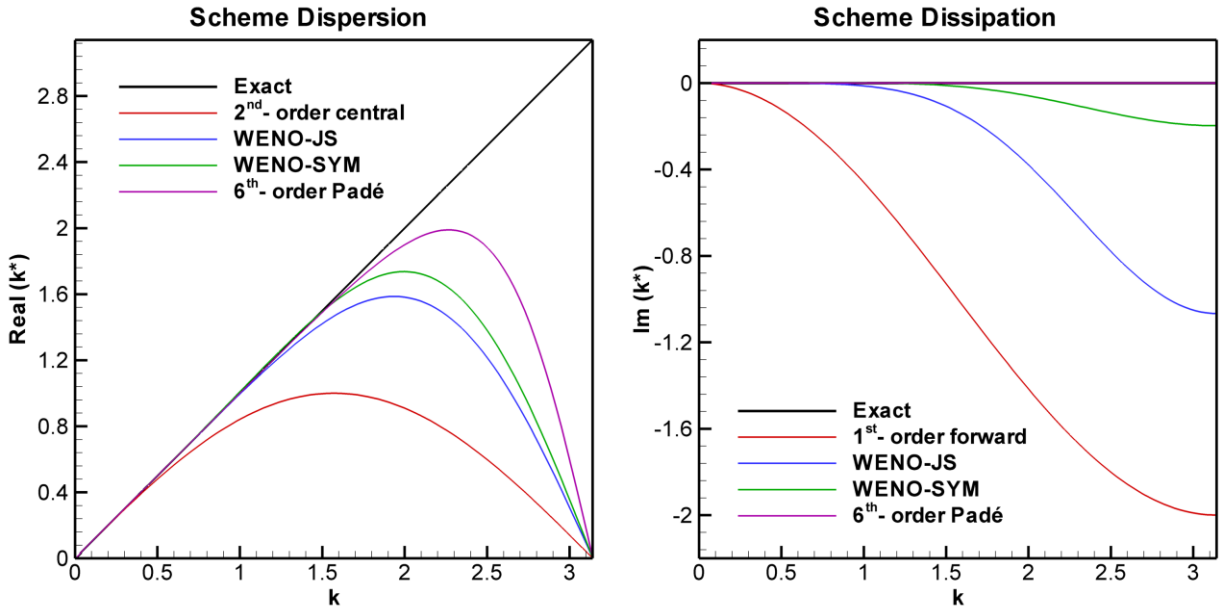


Figure 2.4: Bandwidth efficiency for several numerical schemes.

2.4 Viscous terms

In our algorithm, all spatial gradients, which take place in viscous terms, are calculated explicitly by using the fourth-order accurate central difference scheme. To this end, derivatives in *x*-direction are represented usually as:

$$\frac{\partial}{\partial x}(\mu\frac{\partial u}{\partial x}) = \frac{\partial \mu}{\partial x}\frac{\partial u}{\partial x} + \mu\frac{\partial^2 u}{\partial x^2}. \tag{2.38}$$

In approximation of (2.38), Zhang & Jackson (2009) used compact finite difference to discretize all spatial derivatives. It requires tri-diagonal system of linear equations. In the framework of standard fourth-order central differences, we approximated the first-order derivative by

$$\left(\frac{\partial f}{\partial x}\right)_{i,j} = \frac{-f_{i+2,j} + 8f_{i+1,j} - 8f_{i-1,j} + f_{i-2,j}}{12\Delta x} + O(\Delta x^4), \tag{2.39}$$

and the second-order derivative we approximated by

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_{i,j} = \frac{-f_{i+2,j} + 16f_{i+1,j} - 30f_{i,j} + 16f_{i-1,j} - f_{i-2,j}}{12\Delta x^2} + O(\Delta x^4), \tag{2.40}$$

where $f = u, \mathrm{v}$, or μ. Expressions for $(\partial_y f)_{i,j}$ are derived similarly.

2.5 Time integration scheme

The easiest way to obtain a high-order in time discretization is to use a high-order Runge-Kutta method. However, it has been observed that the classical fourth-order Runge-Kutta method may develop large oscillations in the solution although the space discretization is made by TVD. Therefore, different designs of Runge-Kutta schemes merit careful inspection.

The semi-discretized form of (2.5) by the method of lines yields a system of ordinary differential equations

$$\frac{d\mathbf{U}}{dt} = F(\mathbf{U}), \tag{2.41}$$

where F is the discrete flux vector. This system of differential equations is solved using the Runge-Kutta method. It is performed within one time step Δt

$$\mathbf{U}^{(i)} = \sum_{i=0}^{k-1}\Big[\alpha_k^{(i)}\mathbf{U}^{(k)} + \Delta t\beta_k^{(i)}F(\mathbf{U}^{(k)})\Big], \qquad i = 1,2,\ldots,m \tag{2.42}$$

with $\mathbf{U}^{(0)} = \mathbf{U}(t)$ and $\mathbf{U}(t+\Delta t) = \mathbf{U}^{(m)}$. Coefficients $\alpha_k^{(i)}$ and $\beta_k^{(i)}$ are given from the constraint of requiring the maximal order of accuracy and from the TVD constraint, as follows.

If $\beta_k^{(i)} > 0$ *and the forward Euler method*

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t\, F(\mathbf{U}^n) \tag{2.43}$$

is TVD under the CFL condition $\lambda \le \lambda_0$ $(\lambda = \Delta t/\Delta x)$*, then the method* (2.42) *is TVD under the CFL condition*

$$\lambda \le \lambda_0 \min_{i,k}\frac{\alpha_k^{(i)}}{\beta_k^{(i)}}. \tag{2.44}$$

Indeed, for each Runge-Kutta stage it holds:

$$TV(\mathbf{U}^{(i)}) = \sum_{j=-\infty}^{+\infty}|\Delta_+\mathbf{U}_j^{(i)}| \;\le\; \sum_{j=-\infty}^{+\infty}\sum_{k=0}^{i-1}\alpha_k^{(i)}|\Delta_+(\mathbf{U}_j^{(k)} + \frac{\beta_k^{(i)}}{\alpha_k^{(i)}}\Delta t F(\mathbf{U}_j^{(k)}))| \;\le\; \sum_{k=0}^{i-1}\alpha_k^{(i)}TV(\mathbf{U}^{(k)}),$$

where we used that $\alpha_k^{(i)} > 0$, and that the forward Euler parts in the sum above are TVD under the constraint (2.44). Use induction by assuming that $TV(\mathbf{U}^{(k)}) \le TV(\mathbf{U}^{(0)})$ for all $k \le i$. This is certainly true for $i = 1$. This inequality gives

$$TV(\mathbf{U}^{(i)}) \;\le\; (\sum_{k=0}^{i-1}\alpha_k^{(i)})TV(\mathbf{U}^{(0)}) = TV(\mathbf{U}^{(0)}).$$

Thus $TV(\mathbf{U}^{(m)}) \le TV(\mathbf{U}^{(0)})$ is verified and it represents the TVD condition $TV(\mathbf{U}^{n+1}) \le TV(\mathbf{U}^n)$.

The second order TVD Runge-Kutta is given by (Gottlieb & Shu, 1998):

$$\begin{aligned}\mathbf{U}^{(1)} &= \mathbf{U}^{(0)} + \Delta t F(\mathbf{U}^{(0)}),\\ \mathbf{U}^{(2)} &= \mathbf{U}^{(1)} + \frac{1}{2}\Big[\mathbf{U}^{(0)} + \mathbf{U}^{(1)} + \Delta t F(\mathbf{U}^{(1)})\Big].\end{aligned} \tag{2.45}$$

The third order TVD Runge-Kutta scheme is given by

$$\begin{aligned}
\mathbf{U}^{(1)} &= \mathbf{U}^{(0)} + \Delta t F(\mathbf{U}^{(0)}), \\
\mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^{(0)} + \frac{1}{4}\mathbf{U}^{(1)} + \frac{\Delta t}{4} F(\mathbf{U}^{(1)}), \\
\mathbf{U}^{(3)} &= \frac{1}{2}\mathbf{U}^{(0)} + \frac{2}{3}\mathbf{U}^{(1)} + \frac{2\Delta t}{3} F(\mathbf{U}^{(2)}).
\end{aligned} \tag{2.46}$$

Note that Poisson equation should be solved at the end of each Runge-Kutta stage.

2.6 Stability condition

All terms in our method are discretized explicitly. This requires restriction on a time step to preserve stability. Here we use a condition proposed by Kang et al. (2000). The convective time step restriction is given by

$$\Delta t \left(\frac{|u|_{\max}}{\Delta x} + \frac{|\mathrm{v}|_{\max}}{\Delta y} \right) \le 1 \tag{2.47}$$

where $|u|_{\max}, |\mathrm{v}|_{\max}$ are the maximum magnitudes of the velocities. The viscous time step restriction is given by

$$\Delta t \left(\nu \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} \right) \right) \le 1, \tag{2.48}$$

The effects of any volume force $\vec{F} = (F_x, F_y)$ can be included in the convection estimate noting that $|u|_{\max} + |F_x|_{\max} \Delta t$ is a linear approximation to a bound on the horizontal component of the velocity. Then

$$\Delta t \left(\frac{|u|_{\max} + |F_x|_{\max} \Delta t}{\Delta x} + \frac{|\mathrm{v}|_{\max} + |F_y|_{\max} \Delta t}{\Delta y} \right) \le 1. \tag{2.49}$$

Rewriting (2.47) as $\Delta t C_{cfl} \le 1$ and (2.48) as $\Delta t V_{cfl} \le 1$, one yields

$$0.5\Delta t \left(C_{cfl} + V_{cfl} + \sqrt{(C_{cfl} + V_{cfl})^2 + 4\frac{|F_x|}{\Delta x} + 4\frac{|F_y|}{\Delta y}} \right) \le 1, \tag{2.50}$$

where $\vec{F} = (F_x, F_y)$ is the net acceleration due to forces such as gravity and surface tension.

2.7 Boundary conditions

In order to maintain the high-order accuracy of spatial discretization, we avoided use of lower-order or one-sided discretization scheme near the wall. Instead, we employed three layers of ghost nodes outside the boundaries of the computational domain. This allowed us to retain a single discretization scheme throughout the entire domain. The values at the ghost nodes are calculated near the wall in the normal to wall direction. Schematic of the ghost nodes for boundary treatment is given by Fig. 2.5.

Periodic boundary conditions:

$$\begin{aligned} u_{N_x+1/2+k,j} &= u_{k+1/2,j}, \\ \mathrm{v}_{N_x+k,j+1/2} &= \mathrm{v}_{1+k,j+1/2}. \end{aligned} \tag{2.51}$$

Solid wall (no-slip) boundary conditions:

$$\begin{aligned} u_{N_x+1/2+k,j} &= -u_{N_x+1/2-k,j}, \\ \mathrm{v}_{N_x+k,j+1/2} &= -\mathrm{v}_{N_x-k,j+1/2}. \end{aligned} \tag{2.52}$$

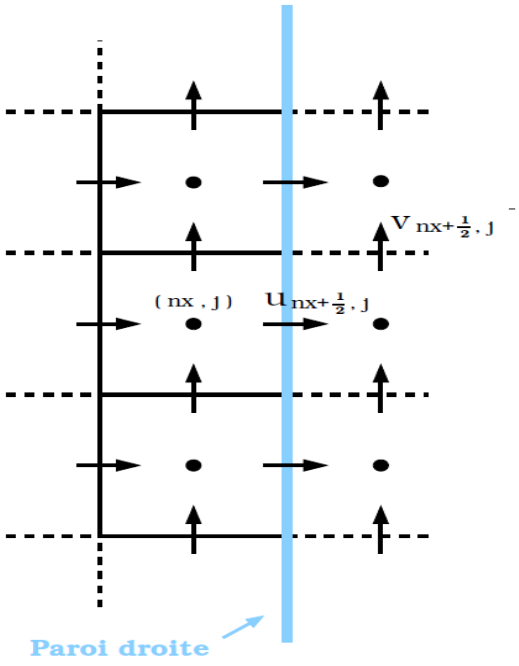


Figure 2.5: Schematics of computational boundary.

2.8 Validation

In this section we presented test computations obtained by described above numerical schemes.

2.8.1. Taylor-Green Vortex problem

First, in order to assess the spatial and temporal accuracy of the overall algorithm, the two-dimensional Taylor-Green Vortex problem is considered. In this case Navier-Stokes equations have analytical solution which is given by:

$$\begin{cases} u(x,y,t) = 1 - A\cos(k(x-t))\sin(k(y-t))e^{-2k^2\nu t}, \\ v(x,y,t) = 1 + A\sin(k(x-t))\cos(k(y-t))e^{-2k^2\nu t}, \\ p(x,y,t) = -0.25A^2(\cos(2k(x-t)) + \cos(2k(y-t)))e^{-4k^2\nu t}. \end{cases} \tag{2.53}$$

In this example, vortices are translated in diagonal direction. Along with such translation their intensity decays exponentially. Coefficient of kinematic viscosity is chosen to be $\nu = 10^{-2}$ which corresponds to Reynolds number $Re = 100$. Vortex amplitude was chosen to be $A = 2.0$ and wavenumber $k = 2\pi$. Fig. 2.6 below represents this solution at initial time and final time $t = 1.0$.

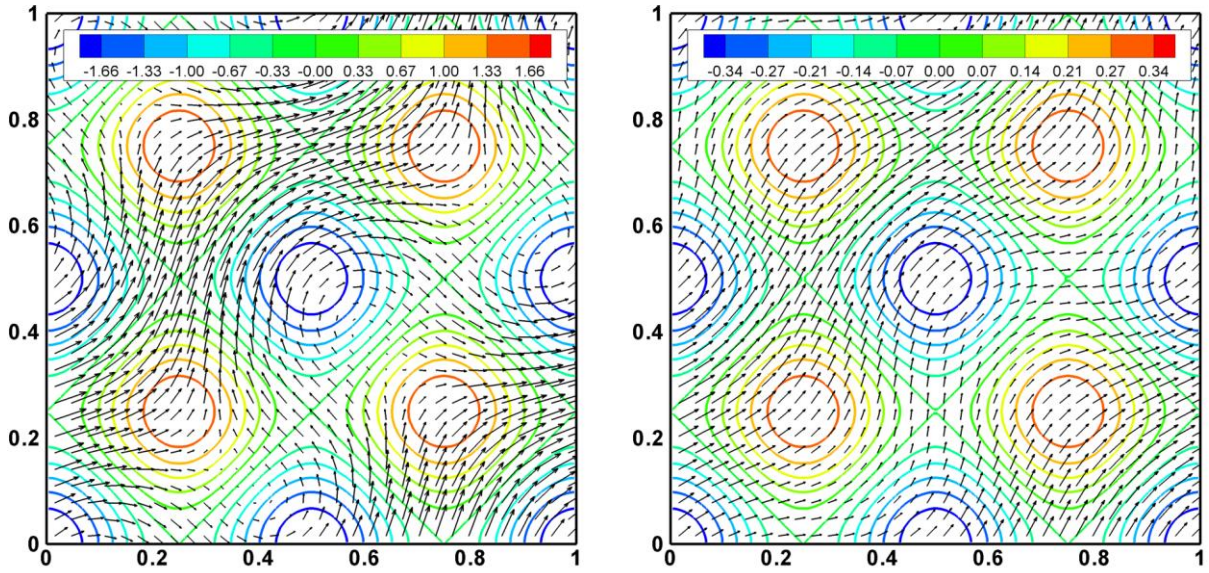


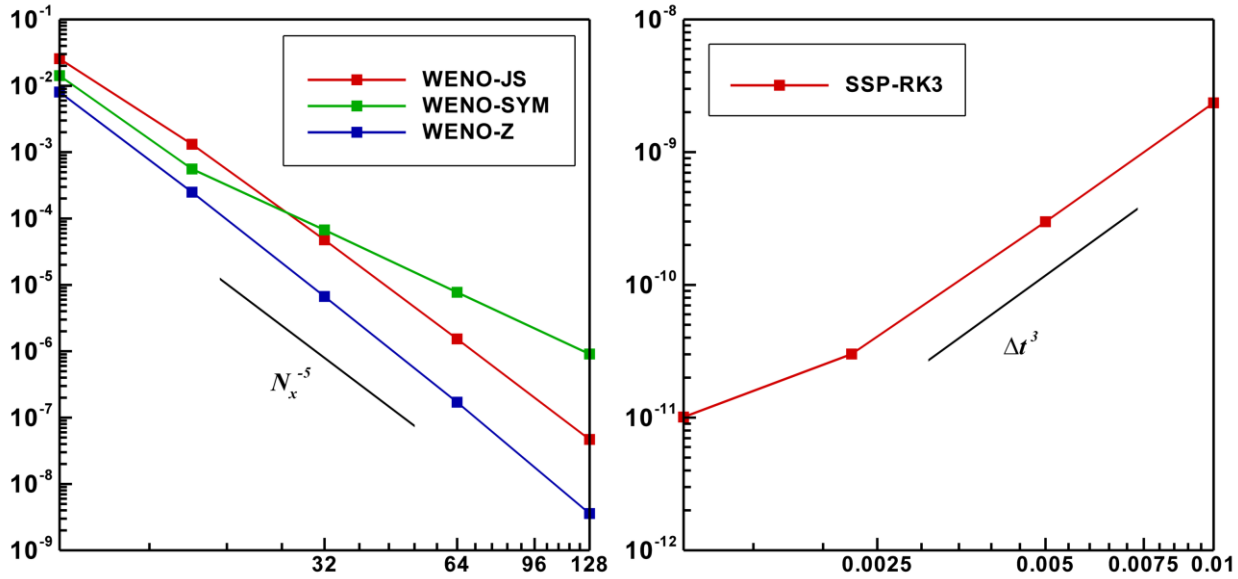
Figure 2.6: Solution $(\vec{u}, p)$ to Taylor-Green Vortex flow at $t = 0$ (left plot); $t = 1.0$ (right plot).

Simulations are performed on the periodic unit square domain using WENO-JS, WENO-SYM and WENO-Z schemes. For each case the L_2 norm of the error in the *u*-component of the velocity is computed to assess the order:

$$\| u \|_{L^2} = \sqrt{\sum (u_{exact} - u)^2 / N}, \tag{2.54}$$

where $N = N_x N_y$ represents the total number of nodes used (N_x and N_y are numbers of nodes in x and y directions, respectively). Because of symmetry of this problem L_2 error norm for the v velocity component $\| v \|_{L^2} = \| u \|_{L^2}$. For assessing the spatial accuracy, the time step is fixed at $\Delta t = 10^{-4}$ and grid resolution chosen for the study is $2^n \times 2^n$ with $n = 3, 4, 5, 6$ and 7. A log-log plot of L_2 error norm vs. number of grids used is shown in Fig. 2.7a. The slope of the plot shows that current algorithm has spatial order of accuracy of 5. Note that for coarse grids (8 and 16 points per wavelength) WENO-SYM is more accurate than classical WENO-JS scheme, while for fine grids the classical scheme has smaller error. The reason is that WENO-SYM scheme is only third order accurate while WENO-JS has fifth order. It is seen that WENO-Z scheme improved the accuracy for coarse and fine grids. The error is about one order of magnitude smaller and fifth order is achieved.

In the next set of simulations, the number of nodes is 256^2, and within the CFL condition, the time step size changes from $\Delta t = 10^{-2}$ to $\Delta t = 1.125 \times 10^{-3}$. The log-log plot of L_2 error norm vs. time step size is shown in Fig. 2.7b. The plot clearly shows third-order accuracy of time integration algorithm of the code. Table 2.1 gives detailed information about spatial error for three different algorithms.



(a) Spatial (b) Temporal

Figure 2.7: L_2 error norm.

Table 2.1: L_2 Error and Convergence Rates for u in Taylor-Green Vortex problem.

	WENO-JS		**WENO-SYM**		**WENO-Z**	
Grid	$\|\|u\|\|_{L^2}$	Rate	$\|\|u\|\|_{L^2}$	Rate	$\|\|u\|\|_{L^2}$	Rate
8×8	2.55×10^{-2}	–	1.42×10^{-2}	–	8.02×10^{-3}	–
16×16	1.31×10^{-3}	4.28	5.57×10^{-4}	4.67	2.49×10^{-4}	5.00
32×32	4.77×10^{-5}	4.78	6.73×10^{-5}	3.05	6.67×10^{-6}	5.22
64×64	1.53×10^{-6}	4.95	7.70×10^{-6}	3.12	1.70×10^{-7}	5.29
128×128	4.67×10^{-8}	5.03	9.06×10^{-7}	3.08	3.58×10^{-9}	5.57

2.8.2. Shear Layer Problem

Next, we perform simulation of Shear Layer test problem. Introduced by Bell *et al*. (1989) and examined more recently in Minion & Brown (1997), the present test concerns two jets in a doubly periodical domain of size $[0,1]\times[0,1]$, to which a sinusoidal perturbation perpendicular to the plane of motion is imposed at the lowest wavenumber resolved by computational mesh. The initial pressure field is uniform $p=0$ and initial velocity field is given by:

$$\begin{aligned} u(x,y) &= \begin{cases} \tanh((y-0.25)/\delta), & \text{for } y \le 0.5, \\ \tanh((0.75-y)/\delta), & \text{for } y > 0.5, \end{cases} \\ v(x,y) &= \varepsilon \sin(2\pi x), \end{aligned} \tag{2.55}$$

where δ is the shear layer width parameter and ε the strength of the initial perturbation. In the absence of any additional perturbations, each of the shear layers rolls up in a single vortex as the flow evolves. Fig. 2.8 depicts components of the velocity field at time $t=0$.

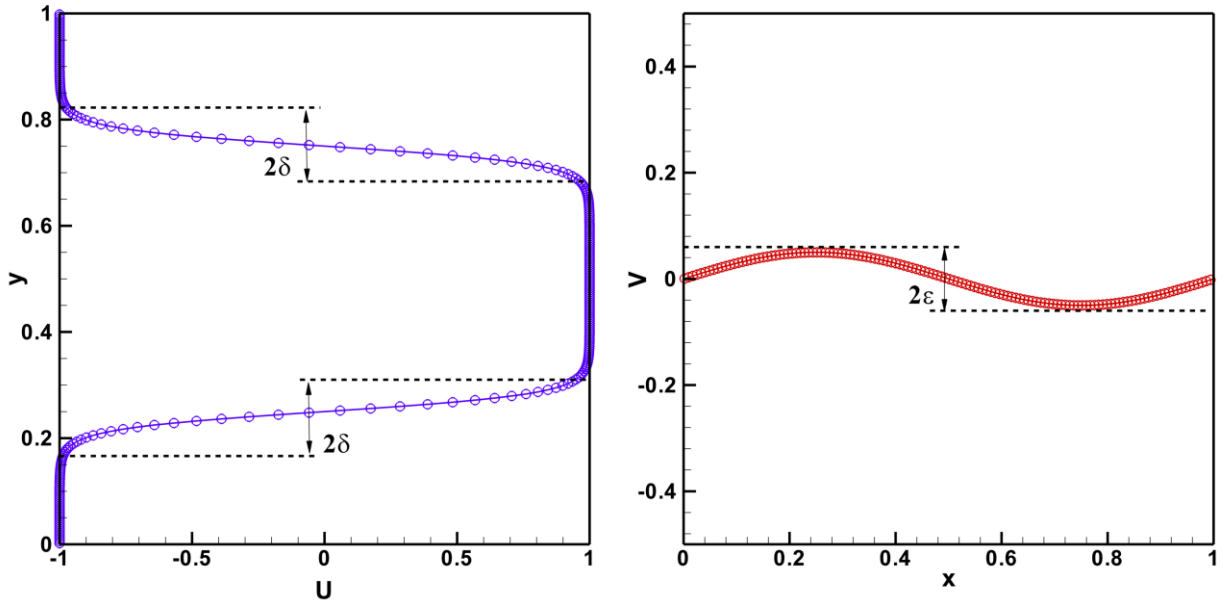


Figure 2.8: Velocity profiles for Shear Layer problem at time $t=0$.

Two cases are considered: thick layer with $\delta=1/30$, $\varepsilon=0.05$ and thin layer with $\delta=1/300$, $\varepsilon=0.05$. In first case there is no viscosity, while the second case is viscous and the corresponding Reynolds number is $Re=10^4$. Calculations are performed on 32×32, 64×64, 128×128 and 256×256 grids. Fig 2.9 shows the vorticity field of the computed solutions on 256×256 grid for different WENO schemes. Qualitatively all schemes give results comparable with the reference solution on 1024×1024 given by spectral method.

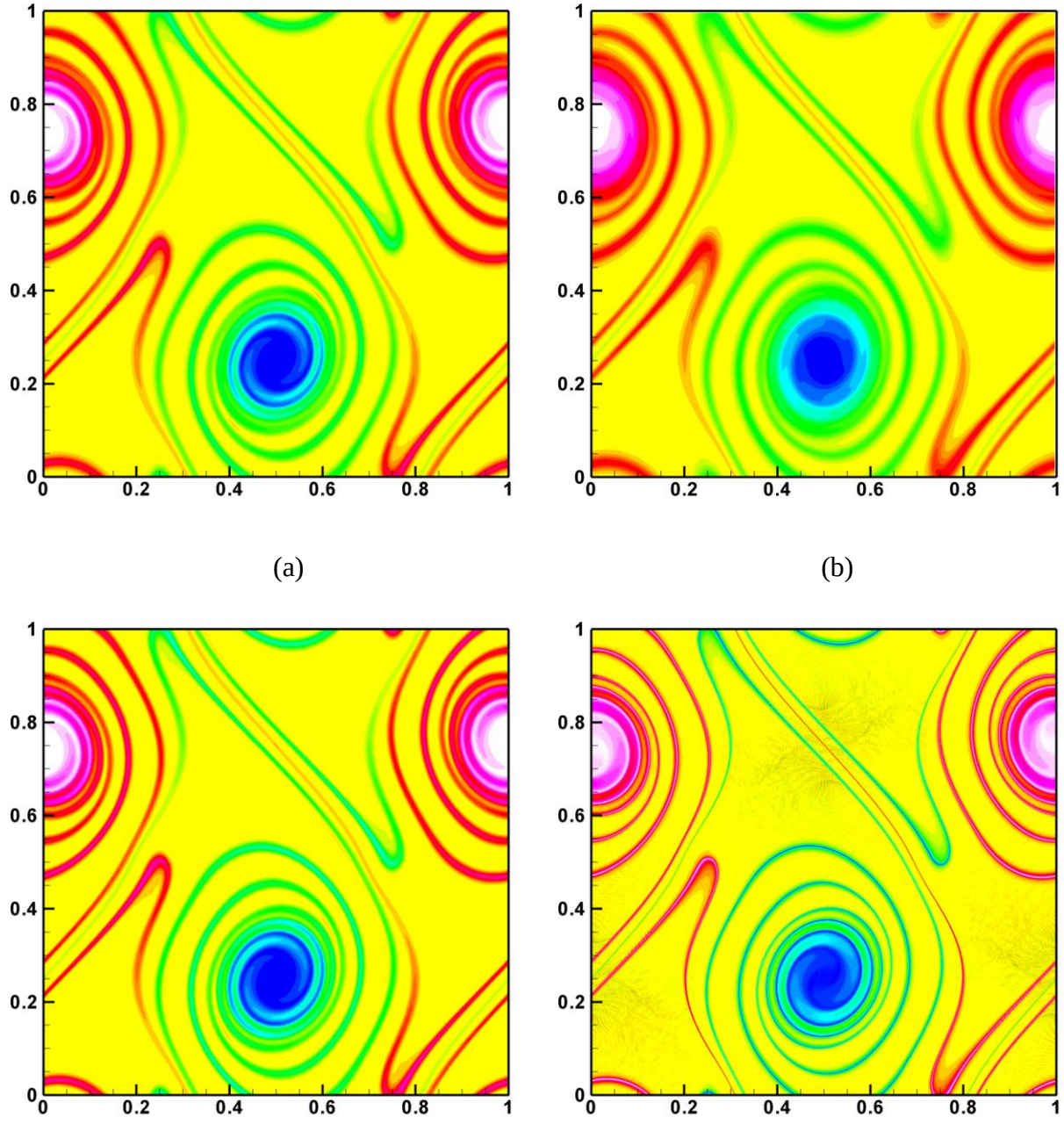


Figure 2.9: Vorticity field in the thick Shear Layer test. Comparison of the numerical schemes at time $t = 2.0$ with a grid resolution of 256×256 points: (a) WENO-JS, (b) WENO-SYM, (c) WENO-Z and (d) Spectral method.

In this case the flow is inviscid, and consequently, the total kinetic energy as well as the total enstrophy should be conserved in time. In the framework of different numerical schemes, we compared evolution of kinetic energy and enstrophy on the coarse grid with 64×64 points. Fig. 2.10 shows the evolution of kinetic energy, and Fig. 2.11 shows the evolution of enstrophy. From both these figures, the advantage of WENO-Z scheme is clearly seen in comparison with WENO-JS and WENO-SYM scheme. The computed loss of kinetic energy on different grids, from 32×32 to 256×256, is presented in Table 2.2.

Next, in Fig. 2.12-2.14, the results are presented for thin shear layer. Here, the flow is viscous. The decay of the total kinetic energy and the enstrophy are compared with the results obtained by spectral method. It is seen that WENO-Z scheme gives better results, while WENO-JS and WENO-SYM have almost the same accuracy in prediction of energy and enstrophy evolutions.

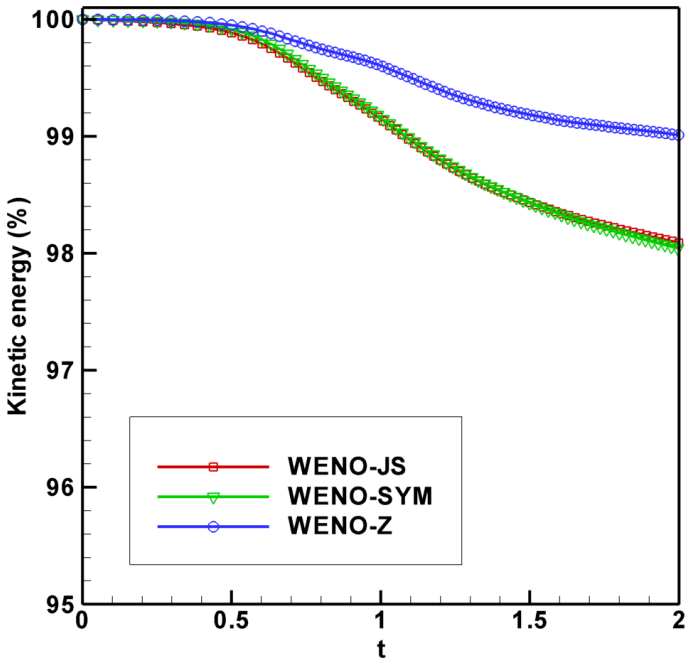


Figure 2.10: Evolution of the total kinetic energy in the thick Shear Layer test. Comparison between WENO-JS, WENO-SYM and WENO-Z. Grid resolution 64×64 points.

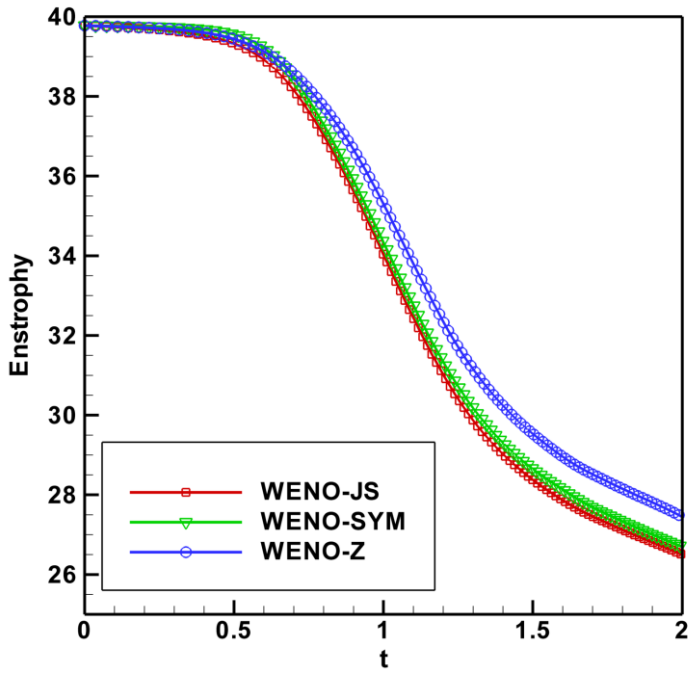


Figure 2.11: Evolution of the total enstrophy in the Shear Layer test. Comparison between WENO-JS, WENO-SYM and WENO-Z. Grid resolution 64×64 points.

Table 2.2: Loss of kinetic energy (%) for the thick shear layer simulations.

Grid	**WENO-JS**	**WENO-SYM**	**WENO-Z**
32×32	6.6554	6.6204	3.6936
64×64	1.9111	1.9591	0.9895
128×128	0.5262	0.5436	0.2518
256×256	0.1305	0.1355	0.0528

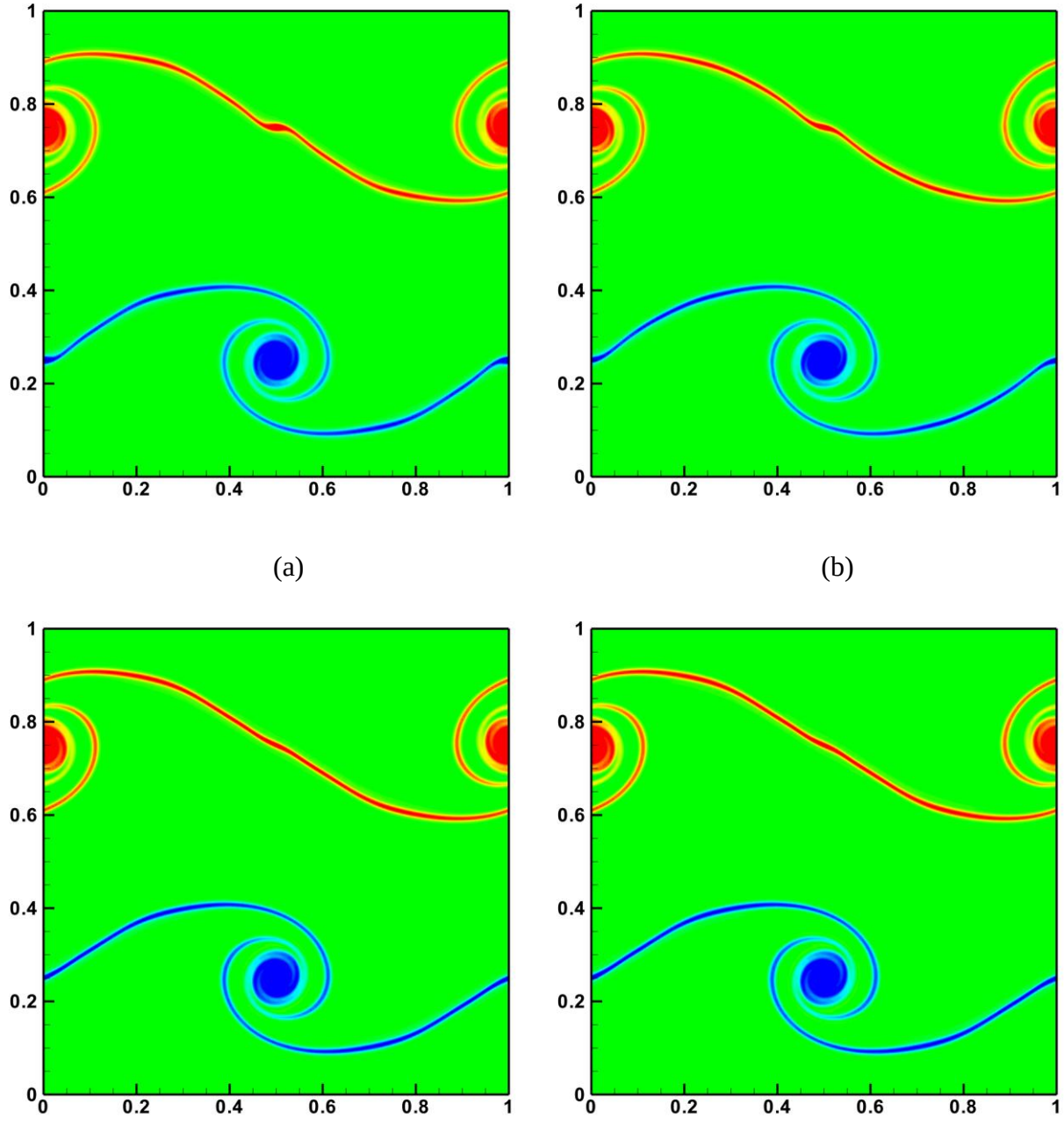


Figure 2.12: Vorticity field in the thin Shear Layer test. Comparison of the numerical schemes at time $t = 0.8$ with a grid resolution of 256×256 points: (a) WENO-JS, (b) WENO-SYM, (c) WENO-Z and (d) Spectral method.

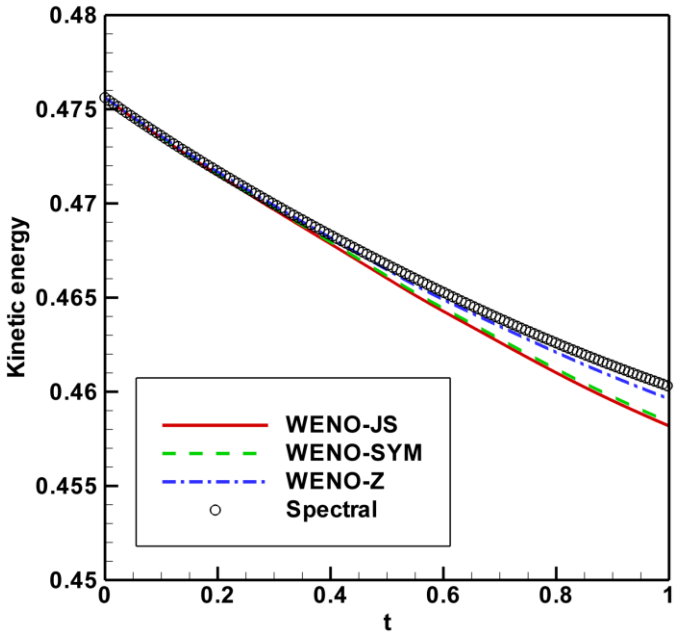


Figure 2.13: Evolution of the total kinetic energy in the thin Shear Layer test. Comparison between WENO-JS, WENO-SYM, WENO-Z and Spectral method. Grid resolution 128×128 points.

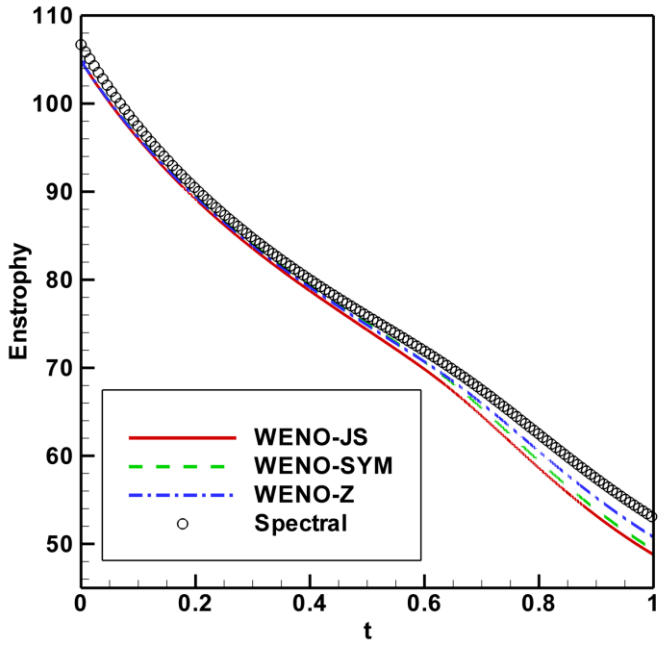


Figure 2.14: Evolution of the total enstrophy in the thin Shear Layer test. Comparison between WENO-JS, WENO-SYM, WENO-Z and Spectral method. Grid resolution 128×128 points.

2.8.3. Driven cavity problem

The next test simulation is performed for two-dimensional driven cavity flow. For assessment of numerical methods for viscous incompressible flows, this problem is the most frequently used benchmark problem. Fig. 2.15 shows its geometry and the boundary conditions.

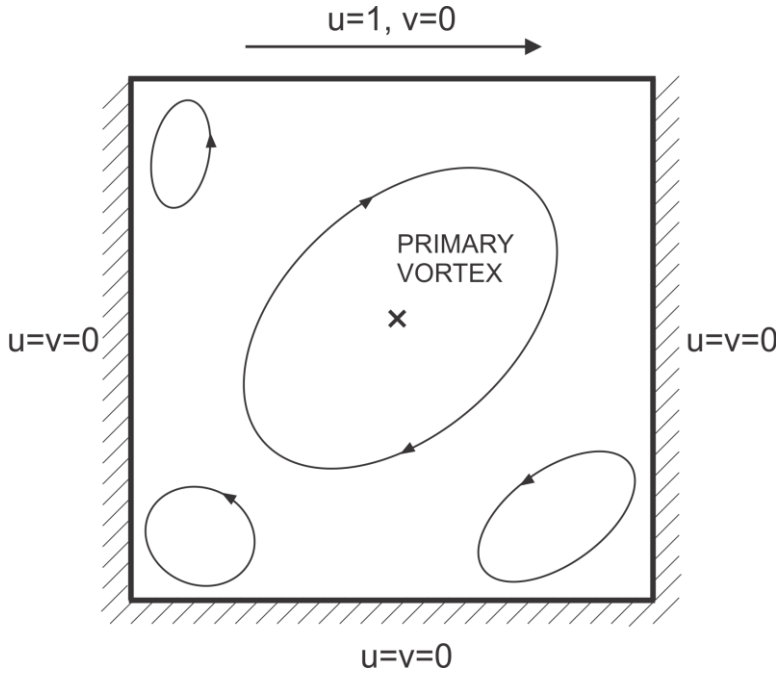


Figure 2.15: Geometry of the driven cavity flow

Flow is driven by upper wall, while three others walls remain fixed. No-slip boundary conditions were employed on all four walls. The moving wall generates vorticity which diffuses inside the cavity thereby forming the driven cavity flow. At low Reynolds number ($Re = 100$), the flow is almost symmetric with respect to the centerline, and two corner eddies are visible. As Reynolds number increases, the center of the main vortex moves toward the downstream corner before it returns toward the center at higher Reynolds numbers. At high Reynolds numbers, several secondary and tertiary vortices begin to appear; the flow structure becomes strongly depending on the Reynolds number.

Fig. 2.16 and Fig. 2.17 show the computed results of streamlines and contours of constant vorticity for several Reynolds numbers $Re = 100$ and 3200. In Fig. 2.18, we present comparisons of the u-component velocities on the vertical centerline and the vertical velocities on the horizontal centerline of the square cavity for $Re = 100$ and 3200, and compare our data

with those of Ghia *et al.* (1982). In each case, obtained velocity profiles exhibit a perfect match with Ghia's results.

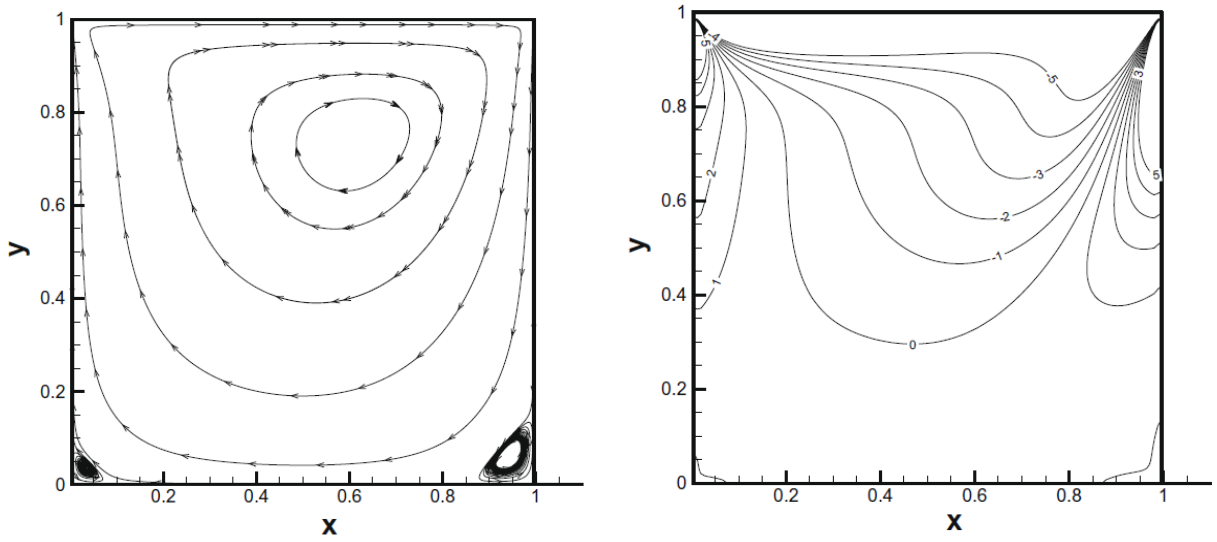


Figure 2.16: Driven cavity test. Streamlines and contours of constant vorticity for $Re = 100$. Grid resolution 128×128 points.

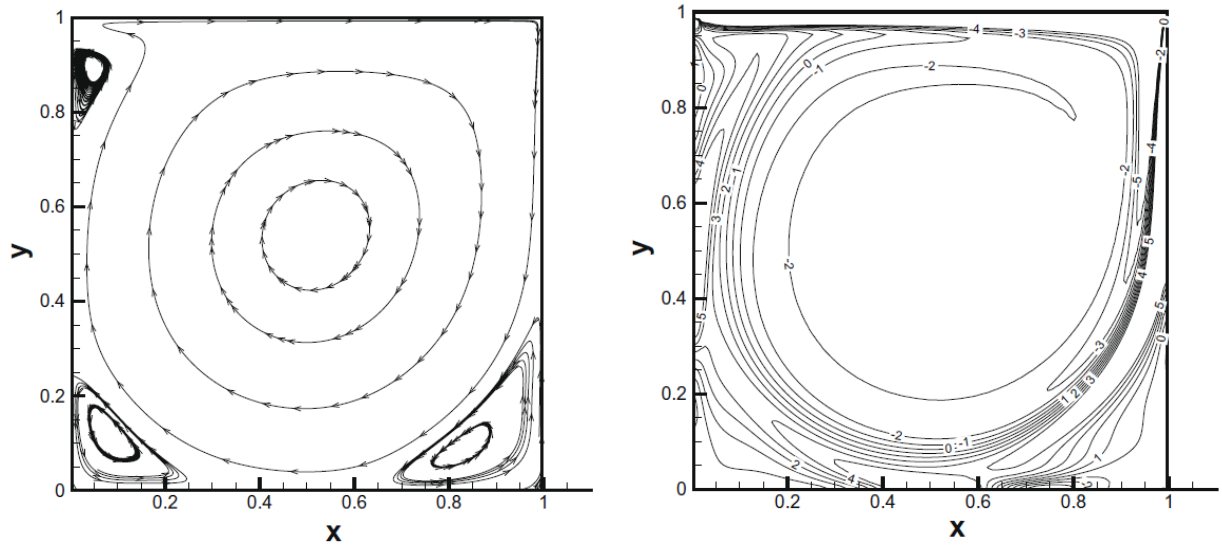


Figure 2.17: Driven cavity test. Streamlines and contours of constant vorticity for $Re = 3200$. Grid resolution 128×128 points.

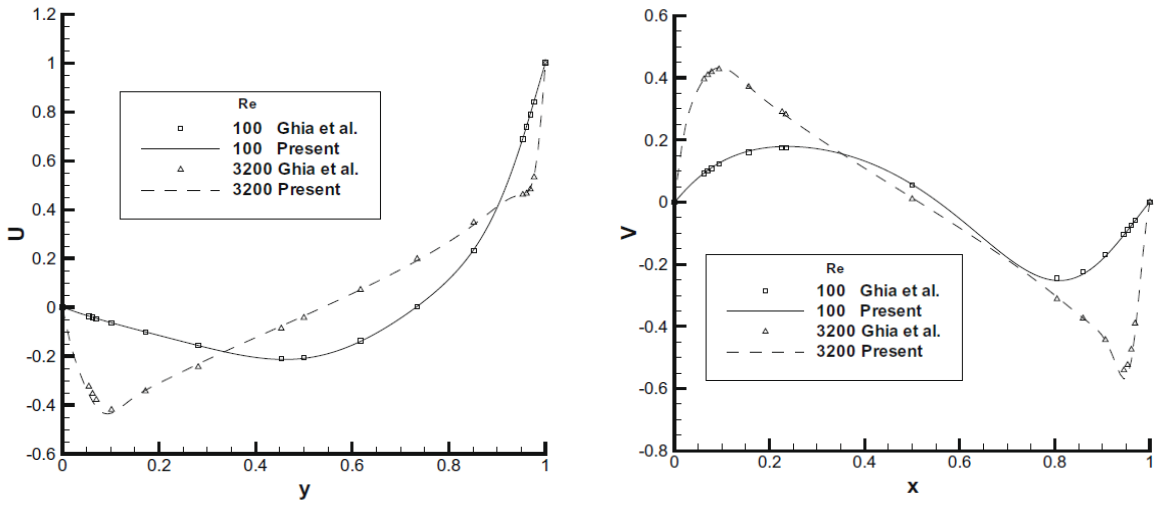


Figure 2.18: Driven cavity test. Centerline slices of velocity: (a) u-component along $y = 0.5$, (b) v-component along $x = 0.5$. Grid resolution 128×128 points.

2.8.4. Two-dimensional decaying turbulence

The accuracy and efficiency of the presented above finite difference approximations is completed by simulation of two-dimensional decaying turbulence. The computational results are compared with those given by spectral method. Parameters of simulation are given by Herring *et al.* (1974). The computational domain is a square with sides of length 2π. Periodic boundary conditions are applied. All runs start from precisely the same initial conditions: a fixed pseudo-random number generator is used for construction of a Gaussian ensemble in two-dimensional incompressible flow with isotropic energy spectrum

$$E(k, t=0) = Ck^4 \exp(-2(k/k_0)^2), \tag{2.56}$$

where $k = |\vec{k}| = \sqrt{k_x^2 + k_y^2}$, C is the constant with value based on initial turbulent energy, and k_0 is the peak of the spectrum. In this study $k_0 = 8$ and the initial turbulent energy is 1.5 in all simulations. The magnitude of velocity Fourier coefficients related to the assumed initial energy spectrum becomes

$$|u(\vec{k})| = \sqrt{\frac{k}{\pi} E(k)} \tag{2.57}$$

The initial velocity distribution in Fourier space is then obtained by introducing a random phase

$$\begin{aligned} u(\vec{k}) &= -\sqrt{\frac{k}{\pi} E(k)}\, \frac{k_y}{k} e^{i\zeta(\vec{k})}, \\ v(\vec{k}) &= \sqrt{\frac{k}{\pi} E(k)}\, \frac{k_x}{k} e^{i\zeta(\vec{k})}. \end{aligned} \tag{2.58}$$

where the phase function is given by $\zeta(\vec{k}) = \xi(\vec{k}) + \eta(\vec{k})$, where $\xi(\vec{k})$ and $\eta(\vec{k})$ are independent random values chosen in $[0, 2\pi]$ at each coordinate point in the first quadrant of the k_x - k_y plane. The conjugate relations for other quadrants are

$$\begin{aligned} \xi(-k_x, k_y) &= -\xi(k_x, k_y), \\ \xi(k_x, -k_y) &= \xi(k_x, k_y), \\ \xi(-k_x, -k_y) &= -\xi(k_x, k_y), \\ \eta(-k_x, k_y) &= \eta(k_x, k_y), \\ \eta(k_x, -k_y) &= -\eta(k_x, k_y), \\ \eta(-k_x, -k_y) &= -\eta(k_x, k_y). \end{aligned} \tag{2.59}$$

The energy spectrum given in (2.56) and initial vorticity field are illustrated in Fig. 2.19. Once the random initial flow field is generated, it is fixed as initial distribution for different numerical schemes assessed hereafter.

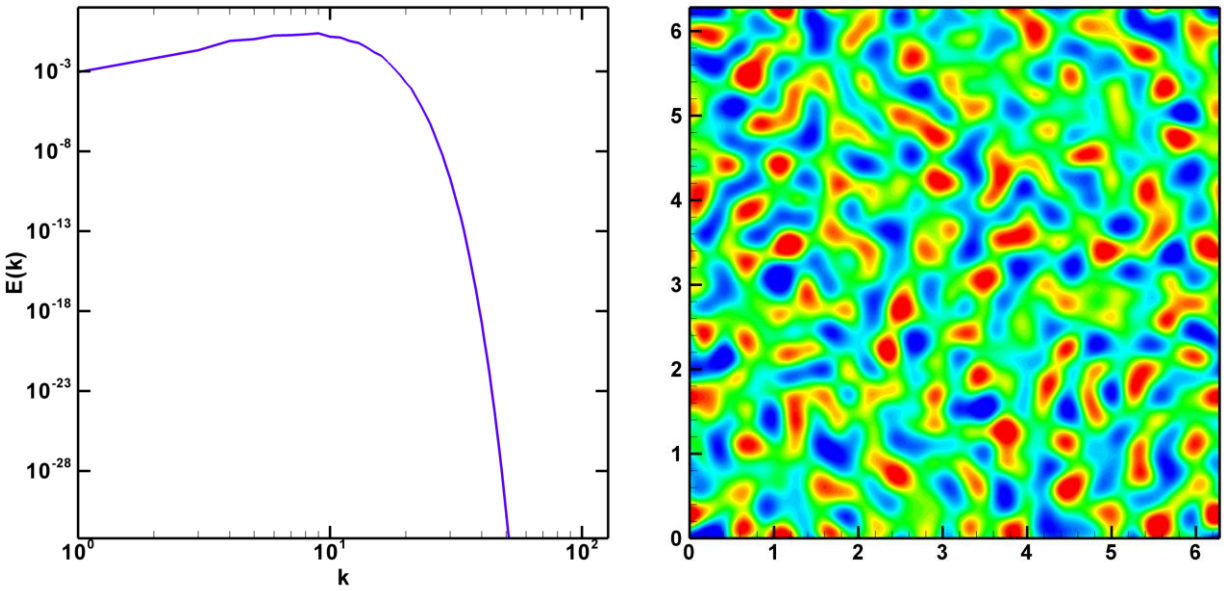


Figure 2.19: Initial energy spectrum (on the left) and vorticity field (on the right) in 2D turbulence test.

In Fig. 2.20, four snapshots of the vorticity-field are exhibited correspondingly to WENO-JS, WENO-SYM, WENO-Z and to Spectral method which we also realized in this work, following the Herring's *et al.* (1974) paper. It is seen that instantaneous vorticity distribution is very similar for all schemes applied. More evidently, this is seen when statistics are calculated. In Fig.2.21, spectra of turbulent energy are given for those numerical schemes at different times. These spectra are very similar to each other. From computed evolution of the velocity field, the following turbulent statistics are compared by different methods: the mean turbulent energy $k_t = \frac{1}{2}\left\langle |\vec{u}|^2 \right\rangle$, the mean enstrophy, $\Omega = \frac{1}{2}\left\langle |\nabla \times \vec{u}|^2 \right\rangle$, the enstrophy dissipation rate, $\eta = \nu \left\langle |\nabla \times (\nabla \times \vec{u})|^2 \right\rangle$, the turbulent macro length scale $L = k_t^{1/2} / \eta^{1/3}$ and the corresponding Reynolds number, $Re_L = k_t / (\nu\eta^{1/3})$. Here, angular brackets denote spatial average. These statistics are presented in Fig 2.22. Due to interpolation errors of velocity components on the grid used here, we did not succeed to have initial statistics in WENO schemes exactly the same as in Spectral method. It is seen in Fig 2.22. However from the decay of turbulent energy it is seen that WENO-JS is more dissipative than WENO-SYM, WENO-Z; two last schemes give very similar decay as one obtained from Spectral method. The same is for the enstrophy decay.

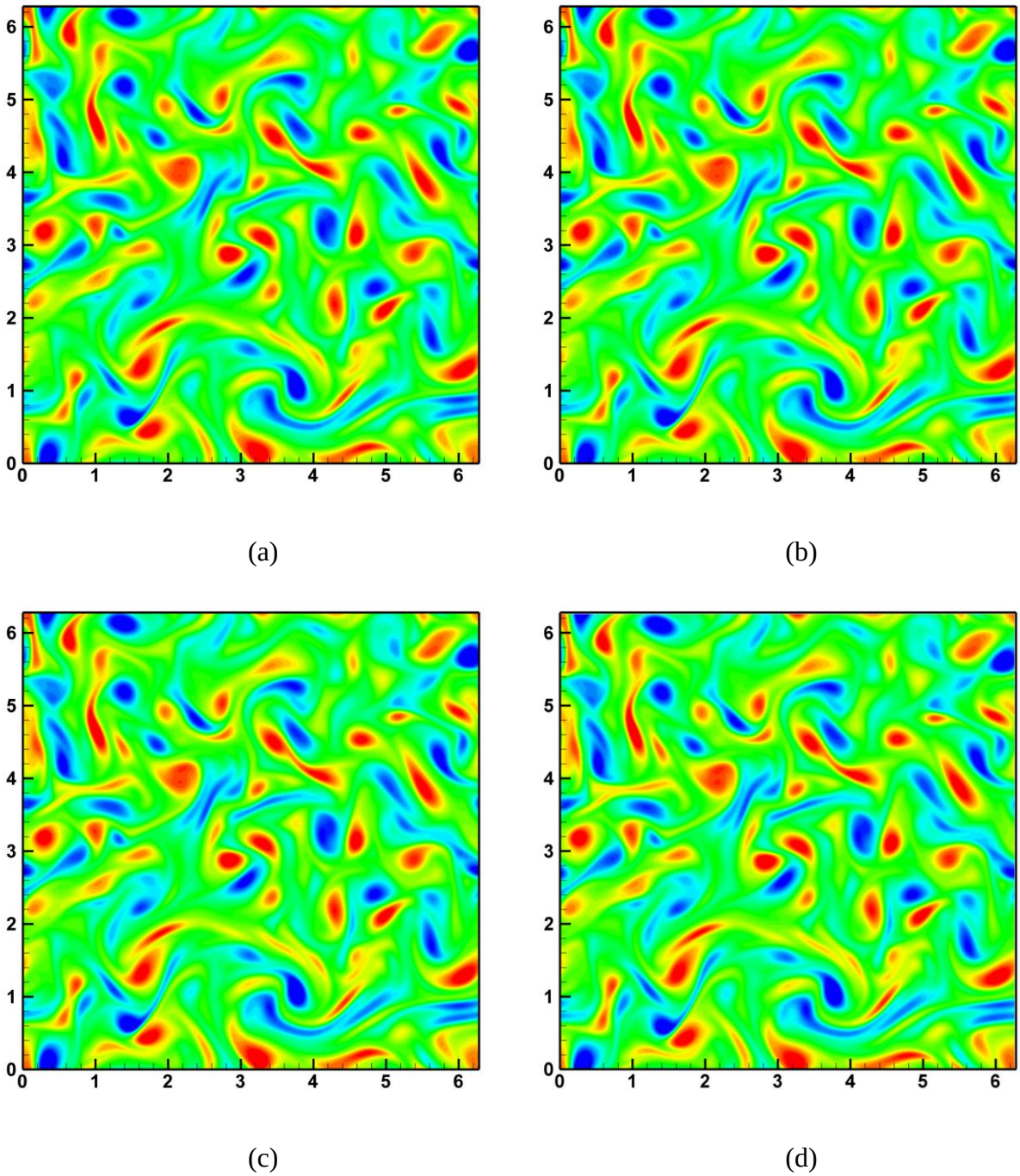


Figure 2.20: Vorticity field in 2D turbulence test. Comparison of the numerical schemes at time $t = 0.8$ for $\nu = 5\times10^{-3}$ with a grid resolution of 256^2 points: (a) WENO-JS, (b) WENO-SYM, (c) WENO-Z and (d) Spectral method.

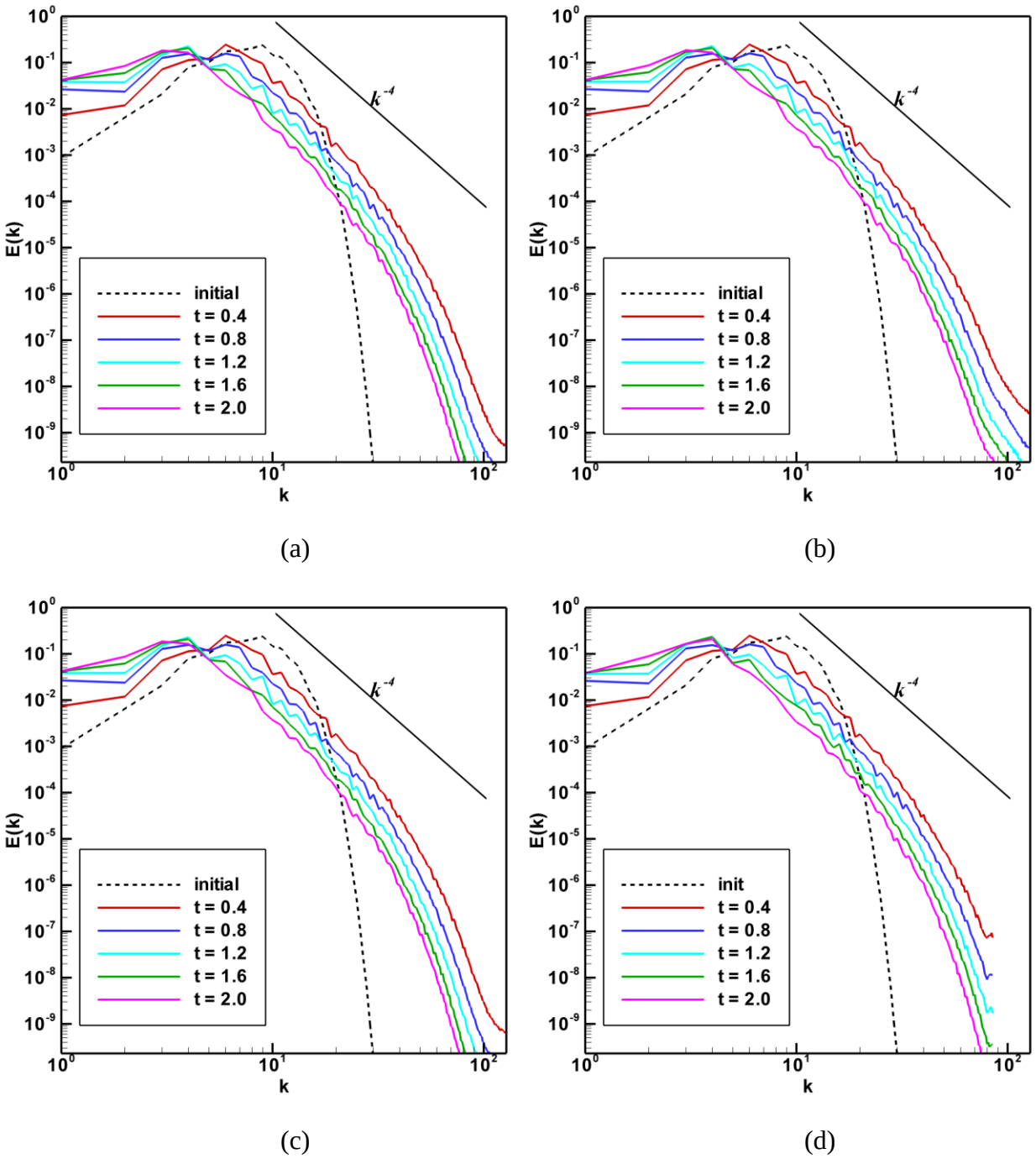


Figure 2.21: 2D turbulence test. Spectrums of energy for different schemes: (a) WENO-JS, (b) WENO-SYM, (c) WENO-Z and (d) Spectral. Grid resolution 256×256.

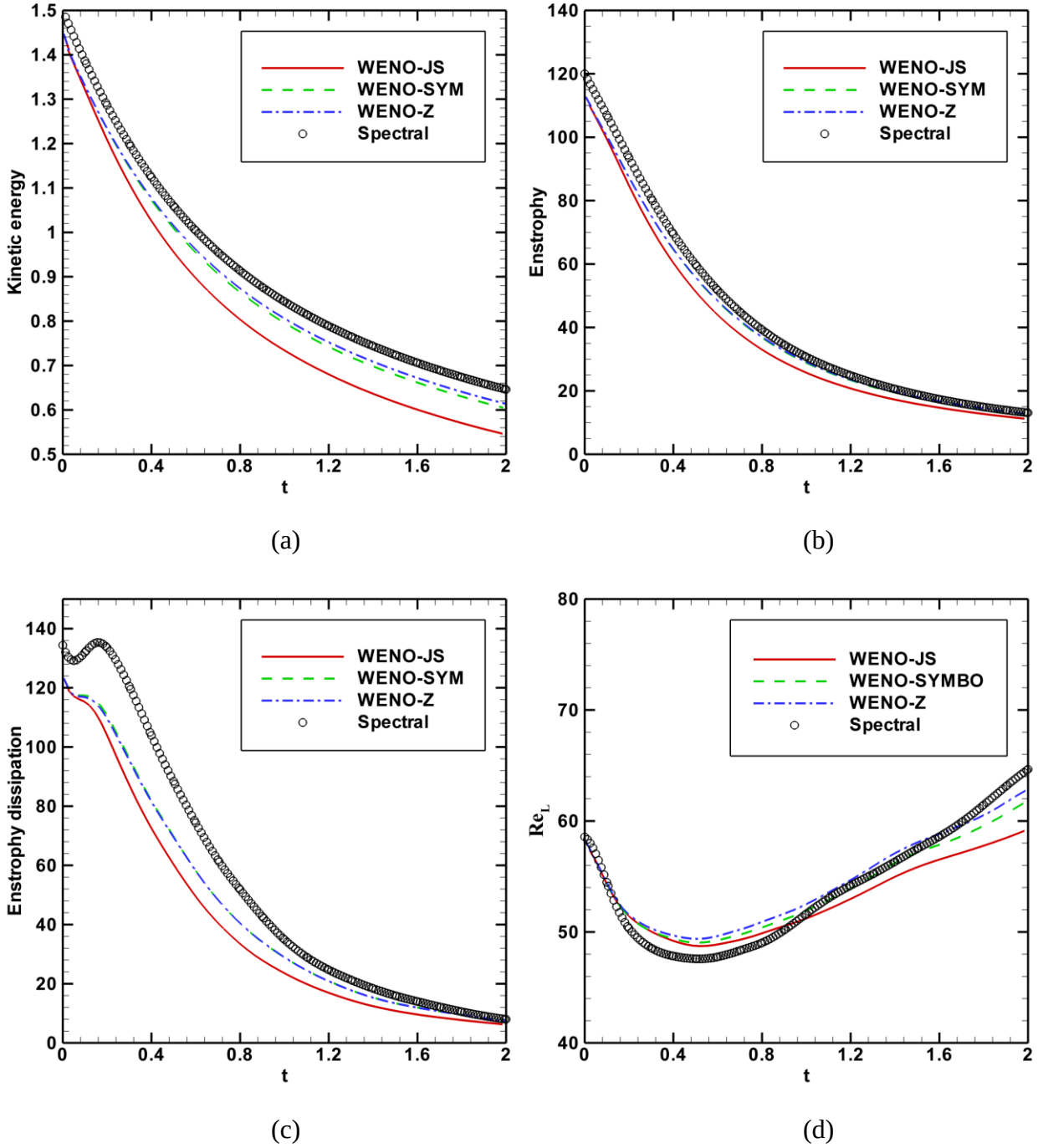


Figure 2.22: 2D turbulence test. (a) Total kinetic energy, (b) total enstrophy, (c) enstrophy dissipation rate and (d) Reynolds number based on integral scale. Comparison of the results for WENO-JS, WENO-SYM and WENO-Z with solution by Spectral method. Grid resolution 128×128.

For a "refined" parameter, such as the enstrophy dissipation, the difference between Spectral method and all appraised here schemes is more pronounced. However it is seen again that WENO-SYM, WENO-Z are closer to Spectral method than WENO-JS. As to the turbulent Reynolds number, no explicit favor can be done to one of WENO schemes; at earlier times WENO-JS is closer to Spectral method, at later times WENO-SYM, WENO-Z are.

2.9 Cost of Poisson solver

This Section concerns numerical solution of Poisson equation. The pressure equation was discretized by second order central differences, and the resulting linear system was solved by using different linear algebraic solvers: Gauss-Seidel method, successive over relaxation (SOR) and incomplete Cholesky conjugate gradients (ICCG). The details about these numerical methods are given in book of Ferziger & Peric (2002), and are not revisited here. In general, Gauss-Seidel or successive over relaxation (SOR) types of iterative algorithms for solving the Poisson equation are of $O(N^2)$, where N is the total number of grid points ($N = N_x N_y$ for two-dimensional problems). The use of these types of iterative Poisson solvers for high-resolution computations, along with long time integration, is not feasible. In order to accelerate these solvers, the multigrid ICCG algorithm has been successfully developed. In this algorithm, the computational effort is reduced to $O(N^{3/2})$. Figure 2.24 demonstrates the required number of iterations to achieve the convergence in Poisson equation for different methods.

Fig. 2.23 shows a comparison between different Poisson solvers. In this figure the dependence of a residual norm on number of iterations is demonstrated for one time step in 2D turbulence test. It is seen that ICCG requires about one order of magnitude smaller number of iterations for convergence than SOR and Gauss-Seidel method. It allows to reduce the total computational cost of our algorithm about 5 times for ICCG in comparison with Gauss-Seidel method. In all simulations presented in this Chapter, as well as in Chapter 5, were performed with ICCG Poisson solver.

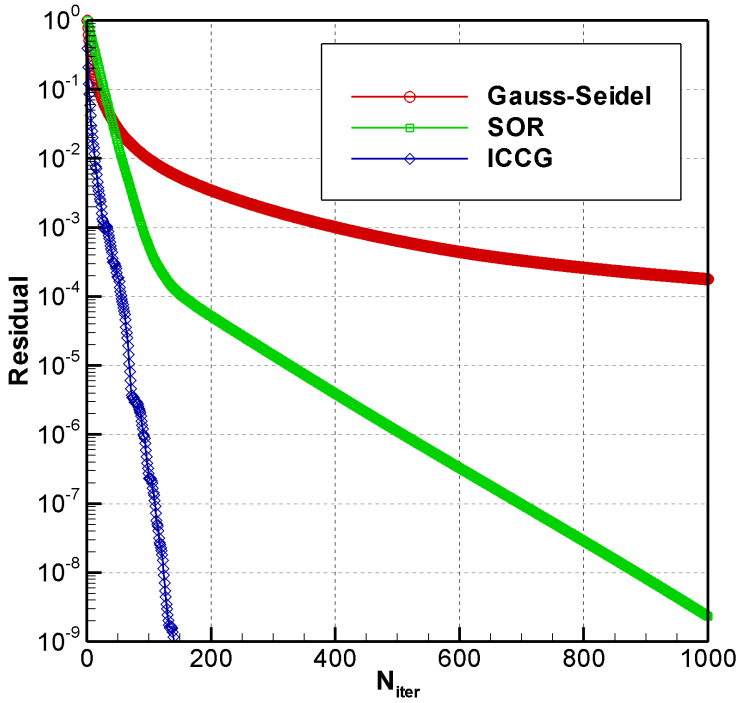


Figure 2.23: Residual versus number of iterations in Poisson solver during one time step in 2D Turbulence test.

2.10 Conclusions

A numerical method for solving the incompressible Navier-Stokes equations with a 5th-order WENO scheme was presented. The WENO scheme is applied to the convective terms in a straight-forward way without artificial compressibility. The algorithm was validated by several numerical tests with exact solutions or previous numerical results, and was shown to be close to 5^{th}-order accurate for velocity variables and 3^{rd}-order in time. We demonstrated that results of WENO-Z scheme for convective terms are fairly better than those obtained for the standard WENO scheme. From other side, the computational cost of the modified scheme remains the same as for the standard WENO scheme. As to WENO-SYM scheme, it shows also improvement in numerical results, but not so significantly as it does by WENO-Z.

Chapter 3

Modified level set equation: derivation

3.1 Source term in the level set equation

To begin, consider the level set equation (1.23), supplemented by a source term proportional to the level set scalar G:

$$\frac{\partial G}{\partial t} + \vec{u} \cdot \nabla G = A(\vec{x},t)G, \tag{3.1}$$

where $A(\vec{x},t)$ is an arbitrary function.

Claim 3.1: The sign of solution of equation (3.1) is conserved in respect to equation (1.23), and consequently, equations (3.1) and (1.23) give the same evolution of the zero level set $G(\vec{x}_f(t),t) = 0$, where $\vec{x}_f(t)$ is the position of points lying on the zero level set surface.

In our derivation this claim plays the key role. It is worthwhile to note that this claim is stated for arbitrary choice of the source term coefficient $A(\vec{x},t)$. It is illustrated hereafter.

Example: Let us consider the case of diagonal translation of a circular interface. Initially G is a signed distance in respect to the circle, and let us choose two following expressions of the source term coefficient: $A_1(\vec{x},t) = \sin(x)\sin(y)$ and $A_2(\vec{x},t) = x + y$. Solutions of Equation (3.1) corresponding to $A_1(\vec{x},t)$ and $A_1(\vec{x},t)$, we denote as $G_1(\vec{x},t)$ and $G_2(\vec{x},t)$, respectively. Fig. 3.1 shows the computed iso-contours of both solutions. It is seen that the zero iso-contour (*i.e.* the predicted interface) remains invariant for both source terms, though outside the interface, the solutions are very different. It is clear that these solutions are immaterial for simulation of interface. However, they have direct impact on accuracy of prediction of interface. Truncation errors of the numerical approximation in the vicinity of the zero-level function depends on the

global distribution of the level set scalar. From numerical practice, it is desirable to keep the level set as signed distance function (Chopp, 1993; Sussman *et al.*, 1994). This allows to obtain a better accuracy in prediction of the interface shape, of interface normal vectors and of its curvature. So, the objective is to have such a form of the source term coefficient that the eikonal equation is satisfied.

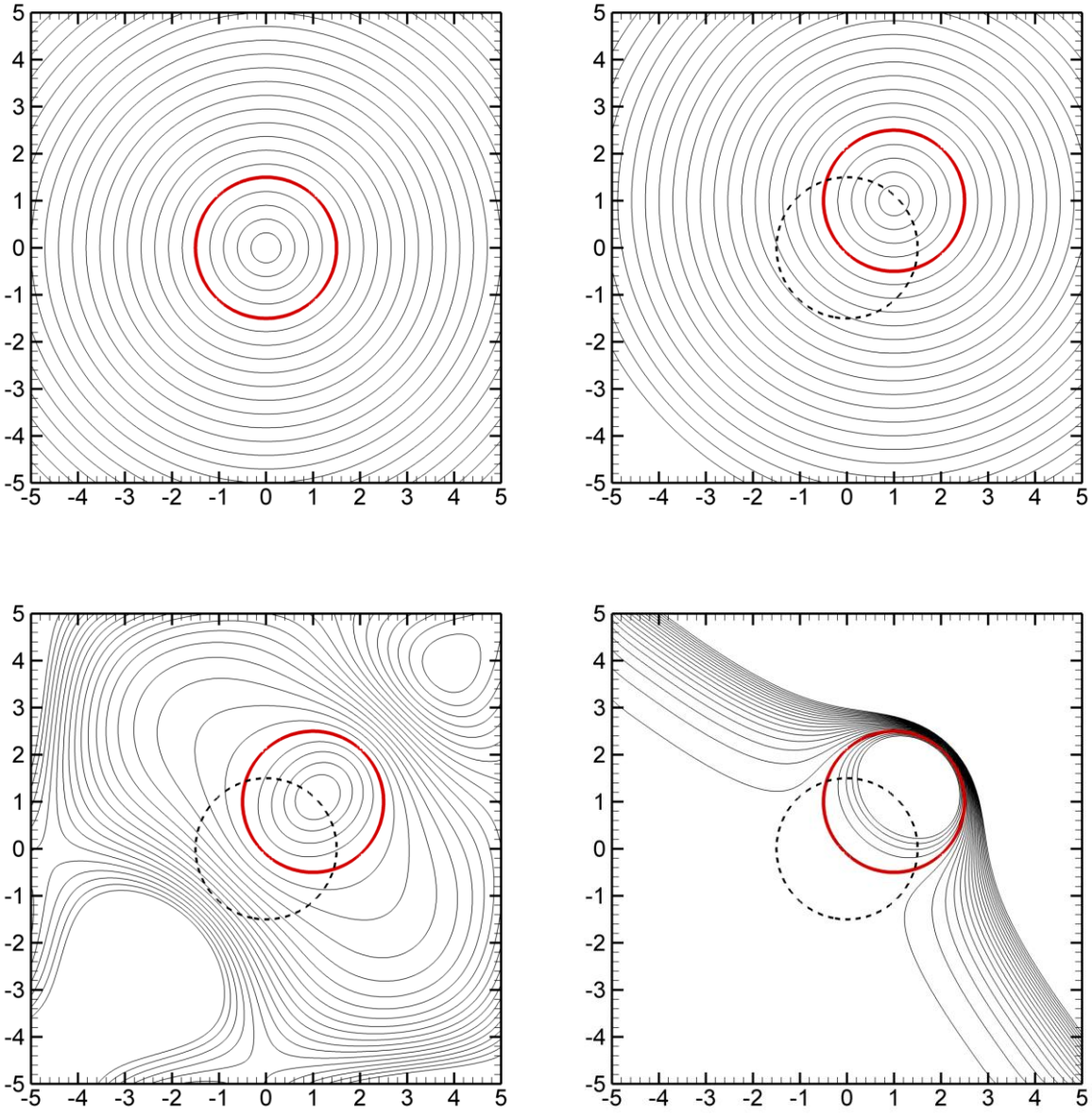


Figure 3.1: Isolines of the level set scalar. From top left to bottom right: initial distribution of G; solutions to (3.1) at time $t = 1.0$ with $A(\vec{x},t) = 0$, $A(\vec{x},t) = x + y$ and $A(\vec{x},t) = \sin(x)\sin(y)$, respectively. Red line represents the zero level set, dashed line represents initial interface position.

3.2 Source term preserving signed-distance solution

In lines of the above claim, we introduce a new scalar field $\varphi(\vec{x},t)$, and let us consider the following initial value problem in suffix notation

$$\frac{\mathrm{D}\varphi}{\mathrm{D}t}=\frac{\partial\varphi}{\partial t}+u_k\frac{\partial\varphi}{\partial x_k}=A(\vec{x},t)\varphi, \tag{3.2}$$

$$\varphi\big|_{t=0}=\varphi_0(\vec{x}), \tag{3.3}$$

where $\varphi_0(\vec{x})$ represents the initial distribution of the level set function and it is taken in form of the signed-distance function, *i.e.* $\left|\nabla\varphi_0(\vec{x})\right|=1$.

Now, let us assume that the solution to (3.2)-(3.3) lies in the class of signed-distance functions, the eikonal equation is satisfied for all times

$$\left|\nabla\varphi(\vec{x},t)\right|=1,\quad \forall t>0 \tag{3.4}$$

The expression for the unit vector, normal to interface, is reduced to $\vec{n}(\vec{x},t)=-\nabla\varphi(\vec{x},t)$. By use of constraint (3.4), the source term coefficient $A(\vec{x},t)$ is determined as follows. Differentiating first equation (3.2) with respect to x_i, and then multiplying it by $2\nabla_i\varphi$, where, by virtue of (3.4), $\nabla_i\varphi\equiv\dfrac{\partial\varphi}{\partial x_i}=-n_i$, we obtain:

$$2\nabla_i\varphi\frac{\partial}{\partial t}\left(\nabla_i\varphi\right)+2\nabla_i\varphi\nabla_i\left(u_k\nabla_k\varphi\right)=2\nabla_i\varphi\nabla_i\left(A\varphi\right) \tag{3.5}$$

Then transforming the second term in (3.5)

$$2\nabla_i\varphi\frac{\partial}{\partial t}\left(\nabla_i\varphi\right)+2\nabla_i\varphi u_k\nabla_i\nabla_k\varphi+2\nabla_i\varphi\nabla_i u_k\nabla_k\varphi=2\nabla_i\varphi\nabla_i A\varphi+2A\nabla_i\varphi\nabla_i\varphi \tag{3.6}$$

Using a change rule for second derivatives, $\nabla_i\nabla_k\varphi=\nabla_k\nabla_i\varphi,$ after summing over the suffix i we see that first two terms in (3.6) are converted in material derivative of $\left|\nabla\varphi\right|^2$,

$$\frac{\mathrm{D}}{\mathrm{D}t}\left|\nabla\varphi\right|^2+2\nabla_i\varphi\nabla_i u_k\nabla_k\varphi=2\nabla_i\varphi\nabla_i A\varphi+2A\left|\nabla\varphi\right|^2 \tag{3.7}$$

By virtue of (3.4), we obtain the governing equation for $A(\vec{x},t)$:

$$\varphi\nabla_i\varphi\nabla_i A + A = \nabla_i\varphi\nabla_i u_k\,\nabla_k\varphi \tag{3.8}$$

or in equivalent form

$$\varphi n_i\frac{\partial A(\vec{x},t)}{\partial x_i} - A(\vec{x},t) = -n_i\frac{\partial u_k}{\partial x_i}n_k \tag{3.9}$$

Equation (3.9) can be integrated analytically by using the method of characteristics. Characteristics of the eikonal equation are the straight lines $\vec{x}_{\varphi=0} = \vec{x} + \varphi\vec{n}$ normal to zero-isosurface $\varphi(\vec{x},t) = 0$. Here $\vec{x}_{\varphi=0}$ implies a point position, at which the characteristics, indexed by the position $\vec{x}$, will land on the zero-isosurface.

For what follows, it is useful to express equation (3.9) in the form of the normal derivatives. To this end consider a distance n from interface along the characteristics, which, again, is a straight line directed normally to the interface. It is clear that along this line, the components of $\vec{n}(\vec{x},t) = -\nabla\varphi\,(\vec{x},t)$ do not change, *i.e.* $\dfrac{\partial n_k}{\partial n} = 0$; besides, by virtue of (3.4), φ aligns n:

$$\varphi = -n \tag{3.10}$$

Then equation (3.9) can be recast as

$$\frac{\partial\left(An - u_k n_k\right)}{\partial n} = 0 \tag{3.11}$$

The exact solution to (3.11) is straightforward:

$$A(\vec{x},t)n = u_k n_k - \left(u_k n_k\right)\Big|_{n=0} \tag{3.12}$$

where the condition $(\;)\big|_{n=0}$ denotes $(\;)\big|_{\varphi(\vec{x},t)=0}$, *i.e.* $\left(u_k\,n_k\right)\big|_{n=0} = u_k(\vec{x}_f(t),t)\,n_k(\vec{x}_f(t),t)$.

It is worthwhile to note that on the zero level set $\varphi(\vec{x},t) = 0$, equation (3.9) takes the following form:

$$A(\vec{x}_{\varphi=0},t)\Big|_{n=0} = n_k\,n_i\frac{\partial u_k}{\partial x_i}\bigg|_{n=0} = n_k\frac{\partial u_k}{\partial n}\bigg|_{n=0} = \frac{\partial u_k\,n_k}{\partial n}\bigg|_{n=0} = \frac{\partial\left(\vec{u}\cdot\vec{n}\right)}{\partial n}\bigg|_{n=0} \tag{3.13}$$

It is seen that at $n \to 0$, (3.12) tends to (3.13).

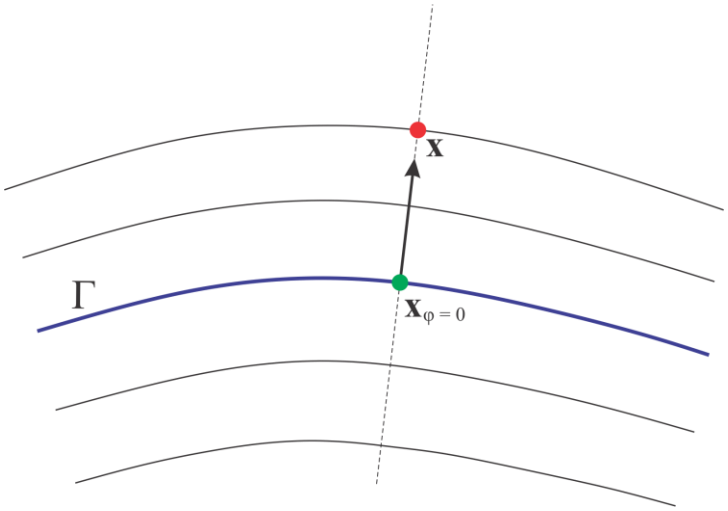


Figure 3.2: Schematic of integration along characteristics of the eikonal equation.

3.3 Local approximations to the source term coefficient in the narrow band

Around the interface of interest, we introduce a narrow band (NB) of a thickness defined by a distance n. Then expanding the speed $u_k n_k$ in (3.12) in a Taylor series about n may provide a useful set of approximations to the source term coefficient $A(\vec{x},t)$. Hereafter the zero- , first- , and second -order approximate forms are derived.

Expansions in a Taylor series for $A(\vec{x},t)$ up to the second-order term

$$A(\vec{x},t) = A_0 + A_1 n + A_2 n^2 + O(n^3) \tag{3.14}$$

and for $u_k n_k$, up to the third-order term

$$u_k n_k = (u_k n_k)\big|_{n=0} + \left(\frac{\partial u_k n_k}{\partial n}\right)\Big|_{n=0} n + \frac{1}{2}\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{n=0} n^2 + \frac{1}{6}\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{n=0} n^3 + O(n^4) \tag{3.15}$$

lead in (3.12) to

$$A_0 + A_1 n + A_2 n^2 + O(n^3) = \left(\frac{\partial u_k n_k}{\partial n}\right)\Big|_{n=0} + \frac{1}{2}\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{n=0} n + \frac{1}{6}\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{n=0} n^2 + O(n^3). \tag{3.16}$$

By equating coefficients at coincident powers of n, the coefficients A_k can be expressed through derivatives $\left(\frac{\partial^{k+1} u_i n_i}{\partial n^{k+1}} \right)\Big|_{n=0}$, $k = 0,1,2$. Thereby setting these expressions for A_k into (3.14), leads to successive approximate forms of $A(\vec{x},t)$ [1].

I. *Zero-order local approximation*

The zero-order coefficient in (3.16) is given by

$$A_0 = \left(\frac{\partial u_k \, n_k}{\partial n} \right)\Big|_{n=0} \tag{3.17}$$

Computation of source term coefficient $A(\vec{x},t) = A_0 + O(n)$ by (3.17) is not convenient for practical purposes, since it includes the derivative of $u_k n_k$ to be taken at the front. Therefore, without changing the order of approximation to $A(\vec{x},t)$, we replace A_0 by its approximation at a spatial point $\vec{x}$ belonging to the narrow band, $\vec{x} \in NB$. Indeed, accurate to the first-order within the narrow band, the first derivative of $u_k n_k$ at the front in (3.17) can be represented by the first derivative of $u_k n_k$ at any $\vec{x} \in NB$:

$$\left(\frac{\partial u_k \, n_k}{\partial n} \right)\Big|_{n=0} = \left(\frac{\partial u_k \, n_k}{\partial n} \right)\Big|_{\vec{x}} + O(n) \tag{3.18}$$

Then an approximation to (3.17) takes the form

$$A_{LA,0}(\vec{x},t) = \left(\frac{\partial u_k \, n_k}{\partial n} \right)\Big|_{\vec{x}} \tag{3.19}$$

Here $A_{LA,0}(\vec{x},t)$ is referred to as *the zero-order local approximation*.

[1] For illustration in Appendix B, expressions for A_k are derived directly from equation (3.9)

II. *First-order local approximation*

From (3.16) including the first-order term, we have

$$A_0 + A_1 n = \left(\frac{\partial u_k n_k}{\partial n} \right)\Big|_{n=0} + \frac{1}{2}\left(\frac{\partial^2 u_k n_k}{\partial n^2} \right)\Big|_{n=0} n \tag{3.20}$$

As previously noted, the fact that coefficients A_0 and A_1 are expressed by velocity derivatives to be taken on the front makes $A(\vec{x},t) = A_0 + A_1 n + O(n^2)$ inconvenient for computation. Then an alternate is the same as for (3.17): without changing the order of approximation to $A(\vec{x} \in NB, t)$, coefficient A_0 and A_1 may be represented by their local approximations within the narrow band. To this end at any $\vec{x} \in NB$, the following Taylor expansions are used to leading order:

$$\left(\frac{\partial u_k n_k}{\partial n} \right)\Big|_{\vec{x}} = \left(\frac{\partial u_k n_k}{\partial n} \right)\Big|_{n=0} + \left(\frac{\partial^2 u_k n_k}{\partial n^2} \right)\Big|_{n=0} n + O(n^2) \tag{3.21}$$

$$\left(\frac{\partial u_k n_k}{\partial n} \right)\Big|_{n=0} = \left(\frac{\partial u_k n_k}{\partial n} \right)\Big|_{\vec{x}} - \left(\frac{\partial^2 u_k n_k}{\partial n^2} \right)\Big|_{n=0} n + O(n^2) \tag{3.21a}$$

$$\left(\frac{\partial^2 u_k n_k}{\partial n^2} \right)\Big|_{\vec{x}} = \left(\frac{\partial^2 u_k n_k}{\partial n^2} \right)\Big|_{n=0} + O(n) \tag{3.22}$$

Then, putting (3.21a), (3.22) into (3.20), the local approximation to $A(\vec{x} \in NB, t)$ takes the following form:

$$A_{LA,1}(\vec{x},t) = \left(\frac{\partial u_k\, n_k}{\partial n} \right)\Big|_{\vec{x}} - \frac{1}{2}\left(\frac{\partial^2 u_k\, n_k}{\partial n^2} \right)\Big|_{\vec{x}} n \tag{3.23}$$

which is referred here to as *the first-order local approximation*.

III. *Second-order local approximation*

From (3.16) we have

$$A(\vec{x},t) = \left(\frac{\partial u_k\, n_k}{\partial n} \right)\Big|_{n=0} + \frac{1}{2}\left(\frac{\partial^2 u_k\, n_k}{\partial n^2} \right)\Big|_{n=0} n + \frac{1}{6}\left(\frac{\partial^3 u_k\, n_k}{\partial n^3} \right)\Big|_{n=0} n^2 + O(n^3) \tag{3.24}$$

Here again, without changing the order of this approximation to $A(\vec{x},t)$, $\vec{x}\in NB$, one can transform coefficients $A_0=\left(\frac{\partial u_k\, n_k}{\partial n}\right)\Big|_{n=0}$, $A_1=\frac{1}{2}\left(\frac{\partial^2 u_k\, n_k}{\partial n^2}\right)\Big|_{n=0}$ and $A_2=\frac{1}{6}\left(\frac{\partial^3 u_k\, n_k}{\partial n^3}\right)\Big|_{n=0}$ in their local approximations using the first, second and third-order derivatives of $u_k n_k$ at any $\vec{x}\in NB$. The following Taylor expansions are used

$$\left(\frac{\partial u_k n_k}{\partial n}\right)\Big|_{\vec{x}}=\left(\frac{\partial u_k n_k}{\partial n}\right)\Big|_{n=0}+\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{n=0} n+\frac{1}{2}\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{n=0} n^2+O(n^3) \tag{3.25}$$

$$\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{\vec{x}}=\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{n=0}+\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{n=0} n+O(n^2) \tag{3.26}$$

$$\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{\vec{x}}=\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{n=0}+O(n) \tag{3.27}$$

We find from (3.25) - (3.27):

$$\left(\frac{\partial u_k n_k}{\partial n}\right)\Big|_{n=0}=\left(\frac{\partial u_k n_k}{\partial n}\right)\Big|_{\vec{x}}-\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{\vec{x}} n+\frac{1}{2}\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{\vec{x}} n^2+O(n^3) \tag{3.25a}$$

$$\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{n=0}=\left(\frac{\partial^2 u_k n_k}{\partial n^2}\right)\Big|_{\vec{x}}-\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{\vec{x}} n+O(n^2) \tag{3.26a}$$

$$\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{n=0}=\left(\frac{\partial^3 u_k n_k}{\partial n^3}\right)\Big|_{\vec{x}}+O(n) \tag{3.27a}$$

With the expressions (3.25a) - (3.27a), *the second-order local approximation* to (3.24) is

$$A_{LA,2}(\vec{x},t)=\left(\frac{\partial u_k\, n_k}{\partial n}\right)\Big|_{\vec{x}}-\frac{1}{2}\left(\frac{\partial^2 u_k\, n_k}{\partial n^2}\right)\Big|_{\vec{x}} n+\frac{1}{6}\left(\frac{\partial^3 u_k\, n_k}{\partial n^3}\right)\Big|_{\vec{x}} n^2 \tag{3.28}$$

With the help of definitions $n_k = -\dfrac{\partial\varphi}{\partial x_k}$, $\dfrac{\partial}{\partial n} = -\dfrac{\partial\varphi}{\partial x_j}\dfrac{\partial}{\partial x_j}$, and taking into account that along the characteristics $\dfrac{\partial n_k}{\partial n} = 0$, $\left(\dfrac{\partial\varphi}{\partial x_k}\right)\Big|_{n=0} = \dfrac{\partial\varphi}{\partial x_k}$ and $n = -\varphi$, expressions (3.19) and (3.23) can be incorporated into level set equation (3.1) in the following form:

$$A_{LA,0}(\vec{x},t) = \frac{\partial\varphi}{\partial x_i}\left(\frac{\partial u_k}{\partial x_i}\right)\frac{\partial\varphi}{\partial x_k} \tag{3.29}$$

$$A_{LA,1}(\vec{x},t) = A_{LA,0}(\vec{x},t) - \frac{1}{2}\frac{\partial\varphi}{\partial x_l}\frac{\partial\varphi}{\partial x_m}\left(\frac{\partial^2 u_k}{\partial x_l\,\partial x_m}\right)\frac{\partial\varphi}{\partial x_k}\varphi \tag{3.30}$$

The expression (3.28) can be also rewritten in terms of φ in the level set equation (3.2):

$$A_{LA,2}(\vec{x},t) = A_{LA,1}(\vec{x},t) + \frac{1}{6}\frac{\partial\varphi}{\partial x_n}\frac{\partial\varphi}{\partial x_l}\frac{\partial\varphi}{\partial x_m}\left(\frac{\partial^3 u_k}{\partial x_n\,\partial x_l\,\partial x_m}\right)\frac{\partial\varphi}{\partial x_k}\varphi^2 \tag{3.31}$$

However this expression involves the velocity third-order derivative; this, along with the non-linearity, presents difficulty in practical calculations. In this work, we assessed only the zero-order (3.29) and the first-order (3.30) local approximations to the source term coefficient, as well as its exact expression (3.12). Note that in terms of the level set function φ, the exact expression (3.12) is

$$A(\vec{x},t)\varphi = \left[u_k - \left(u_k\right)\Big|_{n=0}\right]\frac{\partial\varphi}{\partial x_k} \tag{3.32}$$

Finally, the level set equation (3.2) to be integrated with (3.32), and with (3.29), (3.30) in the narrow band, takes the following forms respectively:

$$\frac{\partial\varphi}{\partial t} + u_k\frac{\partial\varphi}{\partial x_k} = \left[u_k - \left(u_k\right)\Big|_{n=0}\right]\frac{\partial\varphi}{\partial x_k} \tag{3.33}$$

$$\frac{\partial\varphi}{\partial t} + u_k\frac{\partial\varphi}{\partial x_k} = \frac{\partial\varphi}{\partial x_j}\left(\frac{\partial u_k}{\partial x_j}\right)\frac{\partial\varphi}{\partial x_k}\varphi \tag{3.34}$$

$$\frac{\partial\varphi}{\partial t} + u_k\frac{\partial\varphi}{\partial x_k} = \left(\frac{\partial u_k}{\partial x_j} - \frac{1}{2}\varphi\frac{\partial\varphi}{\partial x_l}\frac{\partial^2 u_k}{\partial x_l\partial x_j}\right)\frac{\partial\varphi}{\partial x_j}\frac{\partial\varphi}{\partial x_k}\varphi \tag{3.35}$$

Note again that when (3.33) is integrated, there is no need in the re-initialization procedure, but at each time and spatial position, one needs to determine the corresponding value of $\left(u_k\right)\big|_{n=0}$. When (3.34) and (3.35) are integrated, the front needs to be re-initialized. However in comparison to standard approach (1.23) with (1.32), the number of re-initializations is expected to be reduced. Thus the objective in next Chapter is to assess the results of integration (3.33) - (3.35) for different tests configuration.

Remark (*Relation to the extension velocity formulation*). As we mentioned in Chapter 1, the extension velocity method (Adalsteinsson & Sethian, 1999) consists of

$$\frac{\partial \varphi}{\partial t} + \vec{u}^{\,\mathrm{ext}} \cdot \nabla \varphi = 0, \tag{3.36}$$

where extension velocity $\vec{u}^{\,\mathrm{ext}}$ is determined from two conditions:

$$\frac{\partial}{\partial n}\left(\vec{u}^{\,\mathrm{ext}} \cdot \vec{n}\right) = 0 \quad \text{and} \quad \vec{u}^{\,\mathrm{ext}}\big|_{\varphi=0} = \vec{u}\big|_{\varphi=0}. \tag{3.37}$$

It is seen, that our approach with exact source term is equivalent to the extension velocity formulation, where

$$\vec{u}^{\,\mathrm{ext}} = \vec{u} + A\varphi\vec{n}. \tag{3.38}$$

Here, the source term coefficient is given by (3.12). The advantage of our formulation is: it allows naturally to obtain the approximate solutions of $A(\vec{x},t)$ in the vicinity of the isosurface $\varphi(x,t)=0$.

3.4 Particular case: homogeneous strain

Consider the flow produced by homogeneous strain. In this case the velocity field is given by

$$u_i(\vec{x},t) = S_{ij}(t)x_j, \;\; i=1,2,3 \tag{3.39}$$

Here the components of the velocity gradient tensor, S_{ij}, depend at most on time. All spatial derivatives of the velocity of order higher than one are equal to zero, and consequently, all-order local approximations to the source term coefficient, as well as its exact expression, are coincident, and are equal to the zero-order local approximation, *i.e.:*

$$u_k n_k - \left(u_k n_k\right)\Big|_{n=0} = S_{ij}\frac{\partial\varphi}{\partial x_i}\frac{\partial\varphi}{\partial x_j}\varphi \tag{3.40}$$

Indeed, consider a value of the velocity components on the interface $\varphi(\vec{x}_f(t),t)=0$:

$$\left(u_k\right)\Big|_{n=0} = u_k\left(\vec{x}-\varphi\nabla\varphi\right) = S_{kl}\left(x_l - \varphi\frac{\partial\varphi}{\partial x_l}\right) \tag{3.41}$$

By using (3.40) and the property that $n_k = \left(n_k\right)\Big|_{n=0}$ along characteristics of the eikonal equation, we may rewrite the LHS in (3.40) in the following form:

$$u_k n_k - \left(u_k n_k\right)\Big|_{n=0} = \left(u_k - \left(u_k\right)\Big|_{n=0}\right)n_k = \left(S_{kl}x_l - S_{kl}\left(x_l - \varphi\frac{\partial\varphi}{\partial x_l}\right)\right)\frac{\partial\varphi}{\partial x_k} = S_{kl}\frac{\partial\varphi}{\partial x_l}\frac{\partial\varphi}{\partial x_k}\varphi \tag{3.42}$$

Chapter 4

Assessment of the modified level set equation

In this chapter, results of computations obtained by the non-modified level set equation are referred to as "standard approach". These results are compared with results from integration of (3.33) - (3.35), which are referred to as "new approach". Thereafter the results of computation are referred to as "source term 1" for the zero-order local approximation (3.34), "source term 2" for the first-order local approximation (3.35), and "source term 3" for the exact expression of the source term coefficient (3.33). A series of calculations including well-known test problems and several new test cases will be presented.

4.1 Numerical implementation

Integration procedure. Numerical integration of the level set equation was performed using a uniform Cartesian mesh. The two first (simplest) test problems were resolved on the whole computational domain, whereas for other test cases we introduced the adaptive narrow band around the interface (Adalsteinsson & Sethian, 1995; Peng *et al*., 1999). Its schematic is illustrated in Fig. 4.1: the thickness of the narrow band is controlled by presumed parameters $\gamma = 12 \cdot \Delta x$ - the external layer width, and $\beta = 9 \cdot \Delta x$ - the external layer width. In order to avoid the development of spurious oscillations from edges of the narrow band, the cutoff function for the velocity field was introduced in the form of the smoothed Heaviside function (Peng *et al*., 1999). The first-order extrapolation of boundary conditions at the edge of narrow band was done according to (Adalsteinsson & Sethian, 1995).

The level set equation was discretized in time using the 3-stage third-order TVD Runge-Kutta scheme (Shu & Osher, 1988; Gottlieb & Shu, 1998). For spatial discretization of the convective term, the fifth-order WENO scheme (Jiang & Shu, 1996) was used. Concerning discretization of the source term in (3.34) and (3.35), we used the fifth-order WENO scheme for Hamilton–Jacobi equations, which was proposed in Jiang & Peng (2000). In this scheme, the Hamiltonian $H = \alpha_{ij} \partial_i \varphi \partial_j \varphi, \;\; \alpha_{ij} = \varphi \partial_i u_j$ (*e.g.* for zero-order local approximation) is approximated by global

Lax-Friedrichs flux function. Our experience showed that such approximation to the source term eliminates the high-frequency numerical oscillations; later are developed if the central-difference scheme is used.

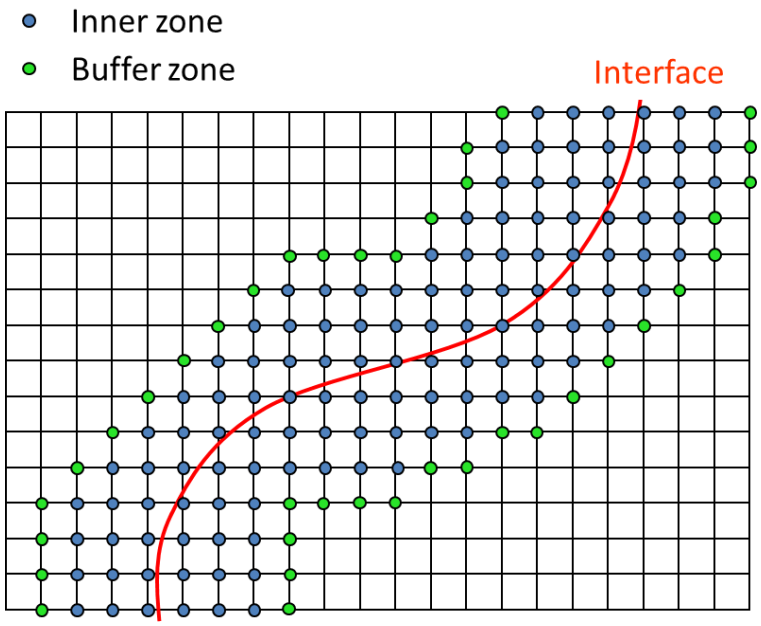


Figure 4.1: Schematic of narrow band: inner and buffer zone.

As to the RHS in (3.33), the main difficulty is to embed $\vec{u}|_{n=0}$ into computation of $\varphi(\vec{x},t)$, *i.e.* to determine the interface propagation velocity in the direction of the characteristics indexed by the position of interest $\vec{x}$. Schematically, our calculation of $\vec{u}|_{n=0}$ is illustrated in Fig.4.2. Similar to Gomes & Faugeras (2000), it is also based on computation of $\vec{x}|_{\varphi=0} = \vec{x} - \varphi\nabla\varphi$ first, and then of $\vec{u}|_{n=0}$, corresponding to $\vec{x}|_{\varphi=0}$. However, the direct computation of $\vec{x}|_{\varphi=0} = \vec{x} - \varphi\nabla\varphi$ reduces the robustness of the algorithms: the errors in $\nabla\varphi$ may introduce errors in $\vec{x}|_{\varphi=0}$. This motivated us to use the iterative gradient descent procedure, as follows:

$$\mathbf{x}_{\varphi=0}^{0} = \mathbf{x},$$

$$\mathbf{x}_{\varphi=0}^{k+1} = \begin{cases} \mathbf{x}_{\varphi=0}^{k} - \max\{\varphi(\mathbf{x}_{\varphi=0}^{k}),\ \mathrm{sgn}\,(\varphi(\mathbf{x}_{\varphi=0}^{k}))\,\Delta\mathbf{x}\}\nabla\varphi(\mathbf{x}_{\varphi=0}^{k}) & \text{if} \quad \varphi(\mathbf{x}_{\varphi=0}^{k}) < 0 \\ \mathbf{x}_{\varphi=0}^{k} - \min\{\varphi(\mathbf{x}_{\varphi=0}^{k}),\ \mathrm{sgn}\,(\varphi(\mathbf{x}_{\varphi=0}^{k}))\,\Delta\mathbf{x}\}\nabla\varphi(\mathbf{x}_{\varphi=0}^{k}) & \text{if} \quad \varphi(\mathbf{x}_{\varphi=0}^{k}) \geq 0 \end{cases} \tag{4.1}$$

Here the superscript k is the number of iterative step. Once the point $\vec{x}\big|_{\varphi=0}$ is computed, the bi-cubic interpolation was used in order to determine velocity in point $\vec{u}\big|_{n=0}$ by velocities in neighboring grid cells.

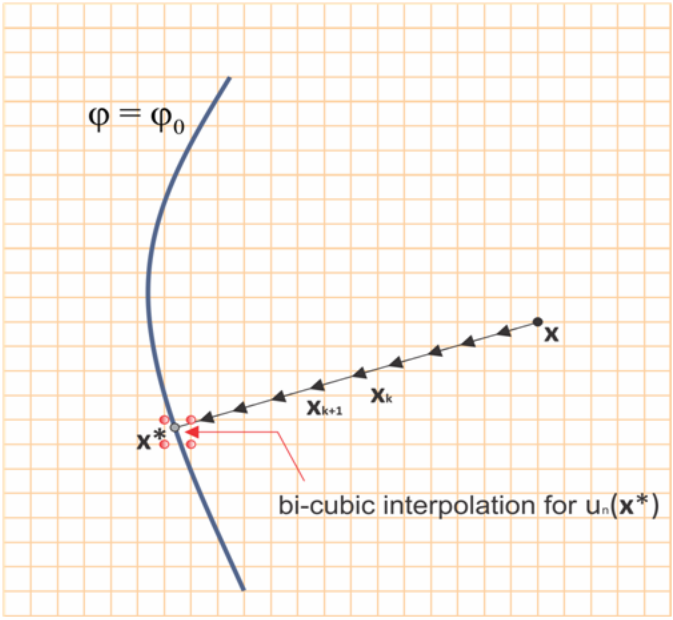


Figure 4.2: Schematic of calculation $\vec{x}_{\varphi=0}$ and $u\big|_{n=0}$.

Accuracy and parameters. When the exact solution to the level set equation is known, the mean computational error in the shape of the interface may be estimated by:

$$E_{shape} = \frac{1}{N}\sum_{k=1}^{N}\varphi_{\text{exact}}(\tilde{x}_k, \tilde{y}_k), \tag{4.2}$$

where N is the number of the interface points $(\tilde{x}_k, \tilde{y}_k)$ computed by linear interpolation between cells adjacent to the zero level set: $\varphi_{i,j}\varphi_{i\pm1,j} \le 0$ or $\varphi_{i,j}\varphi_{i,j\pm1} \le 0$. The mean deviation from the signed distance was estimated by

$$E_{|\nabla\varphi|} = \frac{1}{N}\sum_{i,j}\left(1-\left|\nabla\varphi\right|_{ij}\right)^2. \tag{4.3}$$

The interface curvature in two dimensions is given by

$$\kappa = \nabla \cdot \vec{n} = -\frac{\varphi_x^2 \varphi_{yy} - 2\varphi_x \varphi_y \varphi_{xy} + \varphi_y^2 \varphi_{xx}}{\left(\varphi_x^2 + \varphi_y^2\right)^{3/2}}. \tag{4.4}$$

All spatial derivatives in (4.3) and (4.4) are calculated by second-order central differences. In order to explore the mass conservation property, we estimate the area of the interior region using the first-order accurate approximation to the integral:

$$A = \int_\Omega H(-\varphi)\, dx\, dy. \tag{4.5}$$

where H is a numerically smeared out Heaviside function given by (1.27). Finally, the convergence criterion for reinitialization was used in the following form:

$$\frac{\sum_{|\varphi_{i,j}^n|<\alpha} |\, \varphi_{i,j}^{n,\nu+1} - \varphi_{i,j}^{n,\nu}\,|}{N_\alpha} < \Delta t' \Delta x^p. \tag{4.6}$$

where N_α is the number of grid points where $|\varphi_{i,j}^n| < \alpha$ and the prescribed parameters are chosen as: $\alpha = 1.5 \cdot \Delta x$, $\Delta t' = 0.1 \cdot \Delta x$ and $p = 2$.

4.2 Homogeneous strain

As it was shown in the previous chapter (Section 3.4) for flow produced by homogeneous strain (see equation (3.39)) all local approximations to $A(\vec{x}, t)$, as well as its exact expression, are equivalent, and are equal to the zero-order local approximation. Here we compare our numerical results with analytical solutions. To this end, we consider two simple cases:

(i) Flow is one-dimensional, and it is produced by a homogeneous strain

$$u = -kx, \ \mathrm{v} = 0. \tag{4.7}$$

The initial distribution of the level set function is given by

$$\varphi_0(x) = -x + x_0, \ x_0 = -0.3. \tag{4.8}$$

It is easy to obtain the exact solution of the level set equation (without source term), which is given by

$$\varphi(x,t) = -x \exp(kt) + x_0, \tag{4.9}$$

and exact location of front corresponding to (4.9) is the following

$$x_f(t) = x_0 \exp(-kt). \tag{4.10}$$

(ii) Two-dimensional flow is also stretched homogeneously, but simultaneously, it is rotating. We prescribe the following velocity components:

$$u = x - y,\ \mathrm{v} = 2x - y. \tag{4.11}$$

The initially circular interface, with radius $R = 0.15$ and center $(x, y) = (0, 0)$, is represented by the following level set function:

$$\varphi_0(x, y) = \sqrt{x^2 + y^2} - R. \tag{4.12}$$

In this case exact solution can be obtained by using the method of characteristics:

$$\begin{aligned} &\frac{dx}{dt} = x - y, \quad \frac{dy}{dt} = 2x - y; \\ &x\big|_{t=0} = \xi_1, \quad y\big|_{t=0} = \xi_2. \end{aligned} \tag{4.13}$$

or in vector form of (4.13):

$$\vec{x}' = A\vec{x}, \ \ A = \begin{pmatrix} 1 & -1 \\ 2 & -1 \end{pmatrix}. \tag{4.14}$$

Solution to (4.14) is given by the matrix exponential. Let us determine e^{At}. First, we find the eigenvalues of A:

$$\det(A - \lambda I) = \begin{pmatrix} 1-\lambda & -1 \\ 2 & -1-\lambda \end{pmatrix} = \lambda^2 + 1 = 0 \ \Rightarrow \ \lambda_1 = i, \ \ \lambda_2 = -i. \tag{4.15}$$

Here I is the identity matrix. The eigenvector $v_1 = (v_{11}, v_{21})^T$, corresponding to the eigenvalue λ_1, is determined by

$$\begin{pmatrix} 1-i & -1 \\ 2 & -1-i \end{pmatrix} \cdot \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = 0, \ \ \Rightarrow \ 2v_{11} - (1+i)v_{21} = 0. \tag{4.16}$$

Setting $v_{21} = s$ we obtain $v_{11} = (1+i)s/2.$ Thus the first eigenvector is given by

$$\vec{v}_1 = \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = \begin{pmatrix} (1+i)s/2 \\ s \end{pmatrix} \begin{pmatrix} 1+i \\ 2 \end{pmatrix}. \tag{4.17}$$

Similarly, we find the eigenvector $v_2 = (v_{12}, v_{22})^T$, associated with the eigenvalue λ_2:

$$\begin{pmatrix} 1+i & -1 \\ 2 & -1+i \end{pmatrix} \cdot \begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} = 0, \quad \Rightarrow \; 2v_{12} - (1-i)v_{22} = 0. \tag{4.18}$$

The second eigenvector is given by

$$\vec{v}_2 = \begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} = \begin{pmatrix} 1-i \\ 2 \end{pmatrix}. \tag{4.19}$$

Now the transition matrix can be written as

$$T = (\vec{v}_1, \vec{v}_2) = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} = \begin{pmatrix} 1+i & 1-i \\ 2 & 2 \end{pmatrix}. \tag{4.20}$$

Exponential matrix is determined by

$$e^{At} = Te^{Jt}T^{-1}, \; J = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}. \tag{4.21}$$

Here J is the Jordan form of A. Solution to (4.13) is given by

$$\vec{x} = e^{At}\vec{\xi}. \tag{4.22}$$

Characteristic variables $\vec{\xi} = (\xi_1, \xi_2)^T$ can be obtained directly from (4.22)

$$\xi_1 = x(\cos t - \sin t) + y \sin t, \quad \xi_2 = -2x \sin t + y(\cos t + \sin t). \tag{4.23}$$

Solution to level set equation now is determined in terms of the initial data and characteristic variables as

$$\varphi(x, y, t) = \varphi_0(\xi_1, \xi_2). \tag{4.24}$$

Finally, let us check that (4.24) with (4.23) give us the exact solution. By substitution (4.23) and (4.24) into the level set equation, one yields:

$$\frac{D\varphi}{Dt} = \frac{\partial\varphi_0}{\partial\xi}\left(\frac{\partial\xi}{\partial t} + u\frac{\partial\xi}{\partial x} + v\frac{\partial\xi}{\partial y}\right) + \frac{\partial\varphi_0}{\partial\eta}\left(\frac{\partial\eta}{\partial t} + u\frac{\partial\eta}{\partial x} + v\frac{\partial\eta}{\partial y}\right) =$$

$$= \frac{\partial\varphi_0}{\partial\xi}\left(-x(\sin t + \cos t) + y\cos t + (x-y)(\cos t - \sin t) + (2x-y)\sin t\right) +$$

$$+ \frac{\partial\varphi_0}{\partial\eta}\left(-2x\cos t) + y(\cos t - \sin t) - 2(x-y)\sin t + (2x-y)(\cos t + \sin t)\right) \equiv 0$$

Thus, analytical solutions for both cases of homogeneous strain were obtained. Making use of (4.2), these solutions may give estimation of the mean computational error in interface shape.

Fig. 4.3a and Fig. 4.3b show the initial shape of the interface and the vector plot of the prescribed velocity field corresponding these two cases, respectively. In this case, computations have been performed on a whole computational domain. In the framework of new approach, level sets were not re-initialized. The new approach was compared with results from the standard one without and with re-initialization procedure.

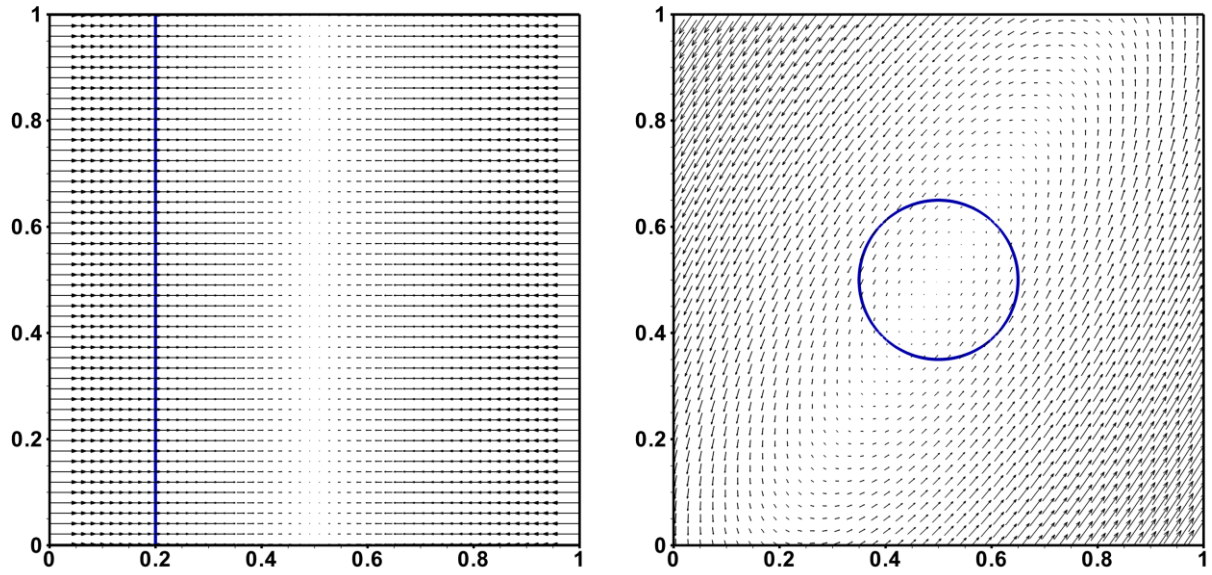


Figure 4.3: Initial location of the interface and the velocity field produced by homogeneous strain; on the left – one-dimensional case (4.8); on the right – two-dimensional case (4.12).

***Case* (i):** The computational domain is $\Omega=[-0.5,0.5]\times[-0.5,0.5]$ with grid resolution 128×128, and in (4.7), $k=2$. For time corresponding to $t=1.0$, Fig. 4.4 shows the level set distribution obtained by standard approach without re-initialization (on the left), and by new approach (on the right). As expected, the new approach conserved the distance between level sets, whereas a high density of level sets is accumulated closely to interface in the case of the standard approach. This is also illustrated in Fig. 4.5 by level set distributions at different times. It is seen that new approach provides the translation of level sets with time, contrary to standard approach; the later inclines level sets with time as $\nabla\varphi(t)\sim e^{kt}$. In Fig. 4.6, we compared results given by new approach with those given by standard one but with re-initializations of level sets. It is seen that standard approach requires about 50 iterations in the reinitialization procedure against zero re-initializations in new approach. Interesting is that although the eikonal equation is satisfied within both approaches, in the standard one with re-initializations, and in the new one without re-initializations, the new approach predicts the analytical position of interface much better. This is demonstrated on the right part in Fig. 4.6. Such an improvement in prediction of front location is also confirmed in Table 4.1 for different grid resolutions.

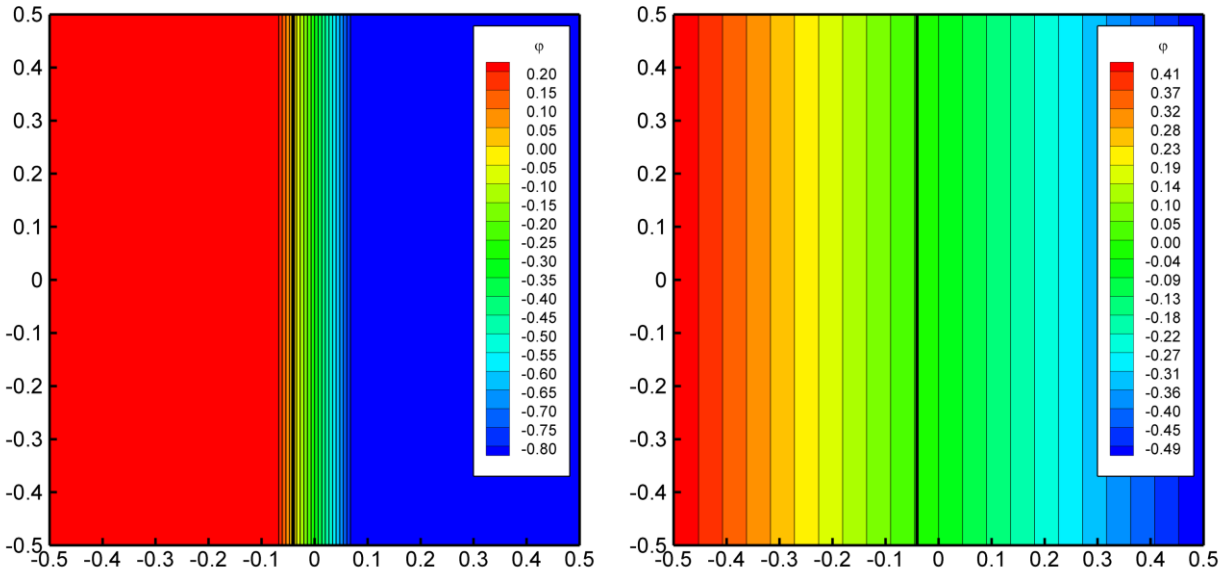


Figure 4.4: Level set isolines at time $t=1.0$ in one-dimensional flow produced by a homogeneous strain; on the left: standard method, on the right: new approach; thick line represents the zero level set.

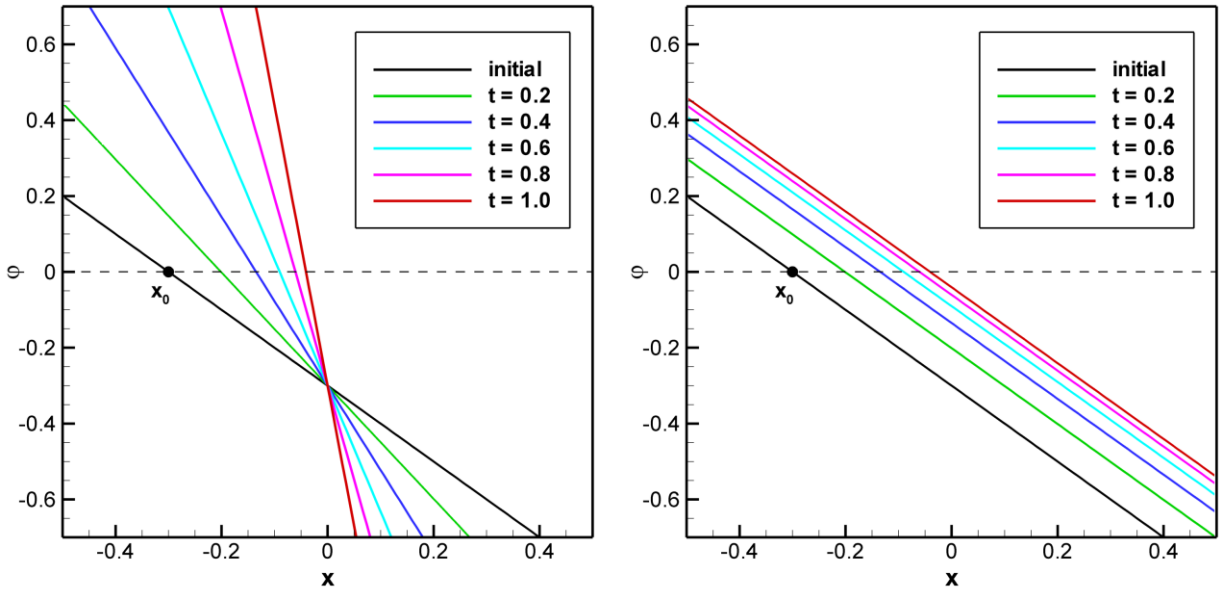


Figure 4.5: Level sets at different times in one-dimensional flow produced by a homogeneous strain; on the left: standard method, on the right: new approach.

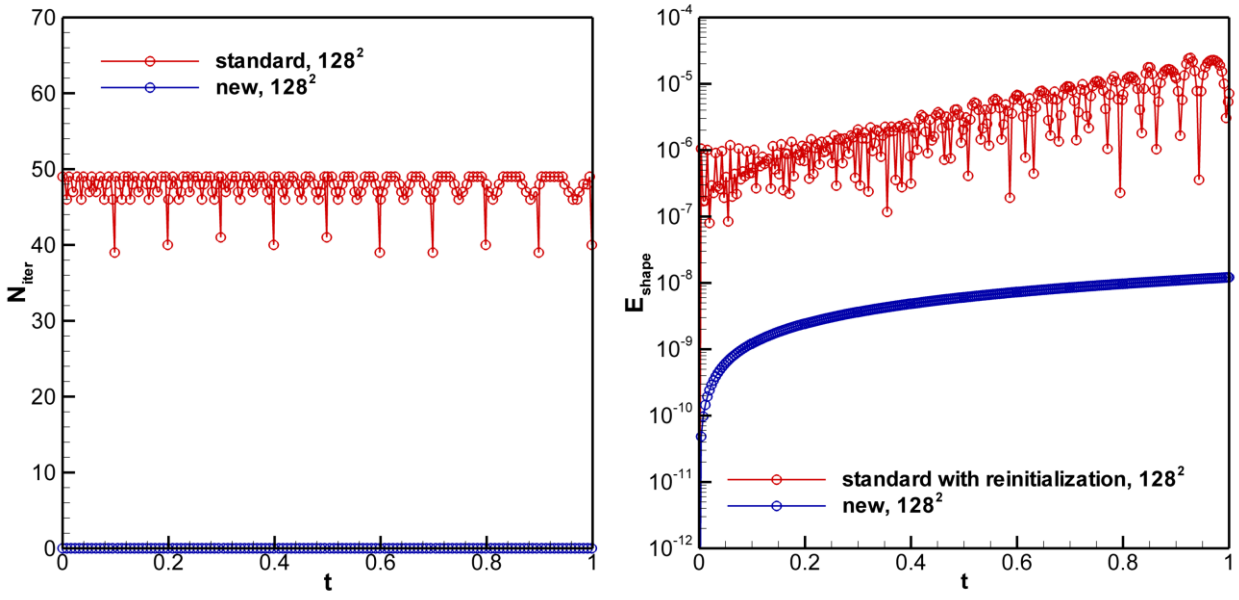


Figure 4.6: One-dimensional flow produced by a homogeneous strain; on the left: comparison of number of iterations in the reinitialization procedure; on the right: mean computational error in the interface location.

Table 4.1. Case (i): mean shape error for different meshes at time $t = 1.0$

mesh	64×64	128×128	256×256	512×512
standard no reinitialization	9.50×10^{-8}	1.19×10^{-8}	1.50×10^{-9}	1.86×10^{-10}
standard with reinitialization	1.28×10^{-5}	9.65×10^{-7}	6.43×10^{-7}	9.23×10^{-8}
new approach	1.31×10^{-8}	1.63×10^{-9}	2.04×10^{-10}	2.53×10^{-11}

From Table 4.1 it is seen that the smallest error was obtained for new approach. Let us explain these results. First, note that in this test case $\frac{\partial^n \varphi}{\partial x^n} = 0$ for $n = 2,3,\ldots$. It means that a truncation error of the numerical scheme is determined only by temporal discretization. In standard approach without reinitialization we have the following estimation $\left|\frac{\partial^n \varphi}{\partial t^n}\right| \le x_{\max} k^n e^{kt}$, where $x_{\max}$ is the maximum value of x. At the same time, in new approach, all temporal derivatives are characterized by smaller absolute value, $\left|\frac{\partial^n \varphi}{\partial t^n}\right| = |x_0| k^n e^{-kt}$ ($k > 1$). Consequently, the truncation error of temporal discretization for modified equation is smaller than it is for standard level set equation. In standard approach with reinitialization procedure a total error is dominated by reinitialization. Zero interface moves during iterations (Russo & Smereka, 2000). Fig. 4.7 illustrates how zero level set perturbs during reinitialization step.

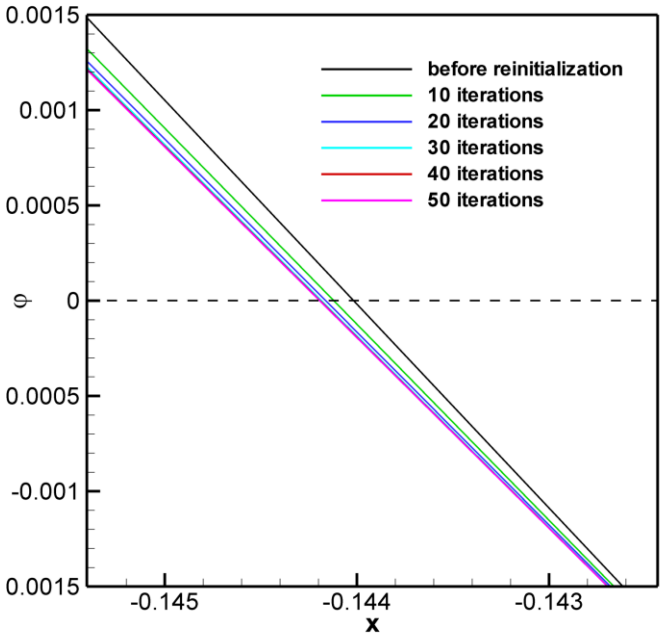


Figure 4.7: One-dimensional flow produced by a homogeneous strain. Front motion during reinitialization.

***Case* (ii):** In this case the velocity field is divergence free, and contrary to the case (i), the area enclosed by the interface should be conserved. The first illustration here is to compare the new and the standard approaches, the both without re-initialization procedure. For different times, and for grid resolution 128×128, Fig. 4.8 shows distributions of the level set and of its gradient norm (in color). The standard approach without re-initializations is presented on the left, and the new approach on the right. Obviously, the new approach preserved the distance between level sets, and the gradient norm is equal to unity in most part of the computational domain. Noticeable exceptions represent points inside the zero level set, where discontinuities in $\nabla\varphi$-distribution, or skeleton points, may appear on the intersection of normal to interface directions. From other side, the standard approach displays a complex distribution around the zero-level set, with increased and decreased densities of level sets in the minor and major directions, respectively. For these two considered approaches, Fig. 4.9 shows the area enclosed by zero level set. It is seen that after a certain time ($t=0.3$ for 64×64; $t=0.5$ for 128×128; $t=0.7$ for 256×256), the flow is so strongly stretched that due to approximations to source terms in (3.33) - (3.35), the new approach fails to keep this area constant. Simultaneously, solution from standard approach without re-initializations exhibits large oscillations in computation of this area. Otherwise when the re-initialization procedure is activated in the standard approach, the numerical oscillations are suppressed, but it incurred the supplementary loss of accuracy in the area computation, in comparison to the new approach. This is shown in Fig. 4.10.

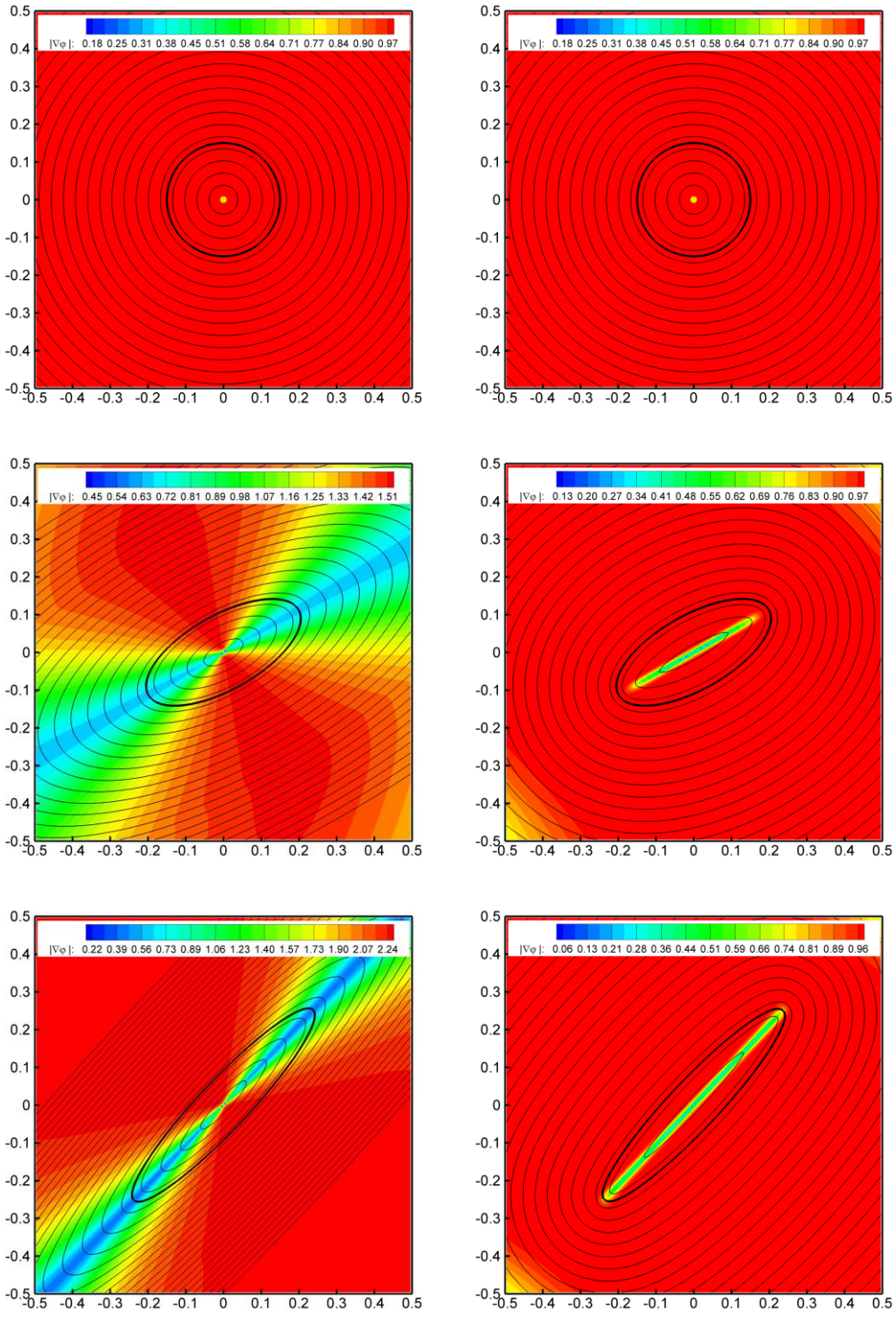


Figure 4.8: Case (ii): isolines of the level set function and its gradient norm (in color); from top to bottom: $t = 0,\ 0.4,\ 1$. Left column: standard method; right column: new approach, the both without re-initializations; thick line corresponds to the zero level set.

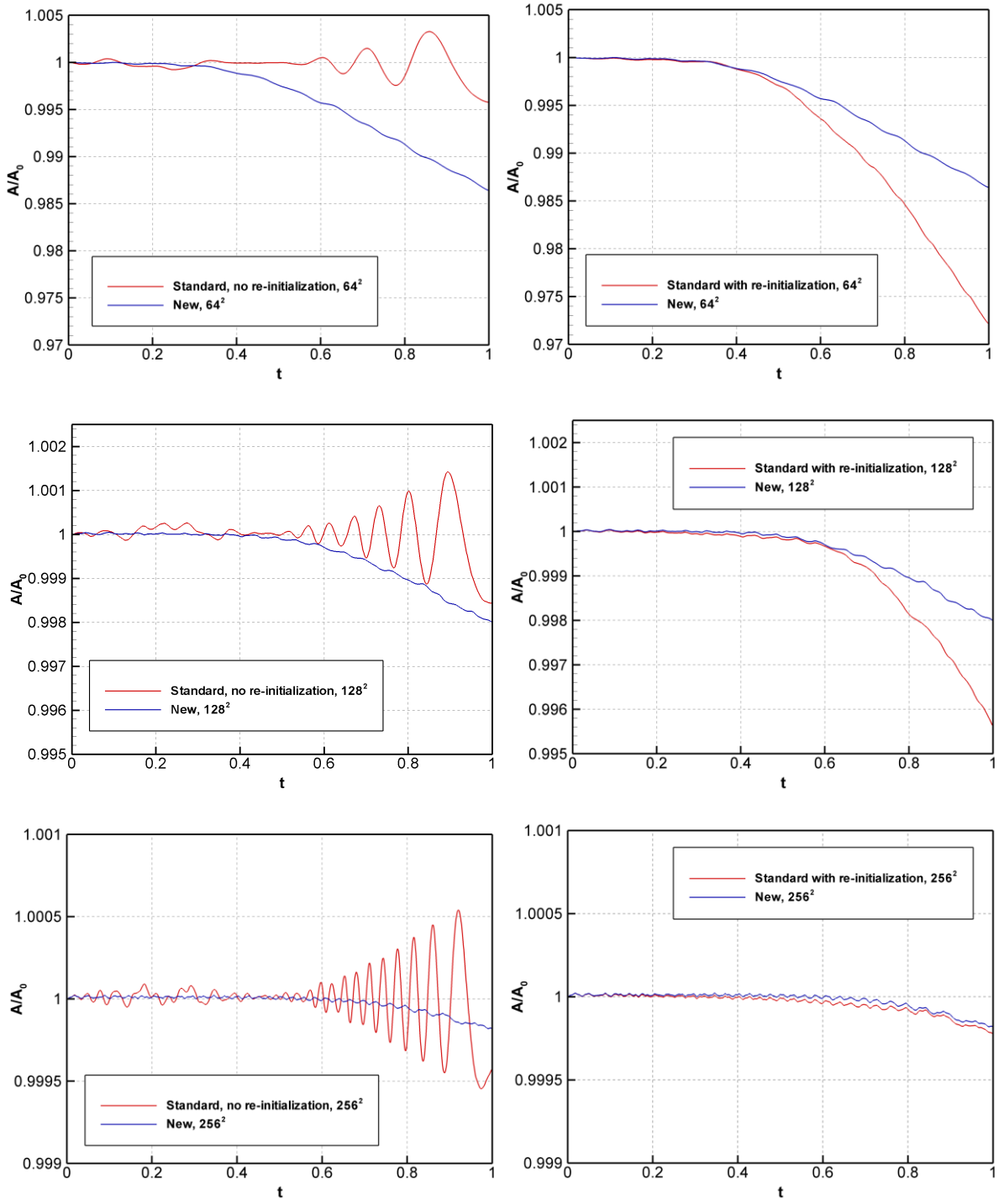


Figure 4.9: Case (ii): evolution of the normalized area in new and standard approaches, the both without re-initializations (on the left); evolution of the normalized area in new and standard approaches, the later with re-initializations (on the right).

Additionally to this comparison, the number of iterations employed with time into re-initialization procedure is given in Fig. 4.10 for grid resolution 128×128. This number is almost $30\div40$ against zero in the new approach. Table 4.2 shows results for mean shape error for different grid resolutions.

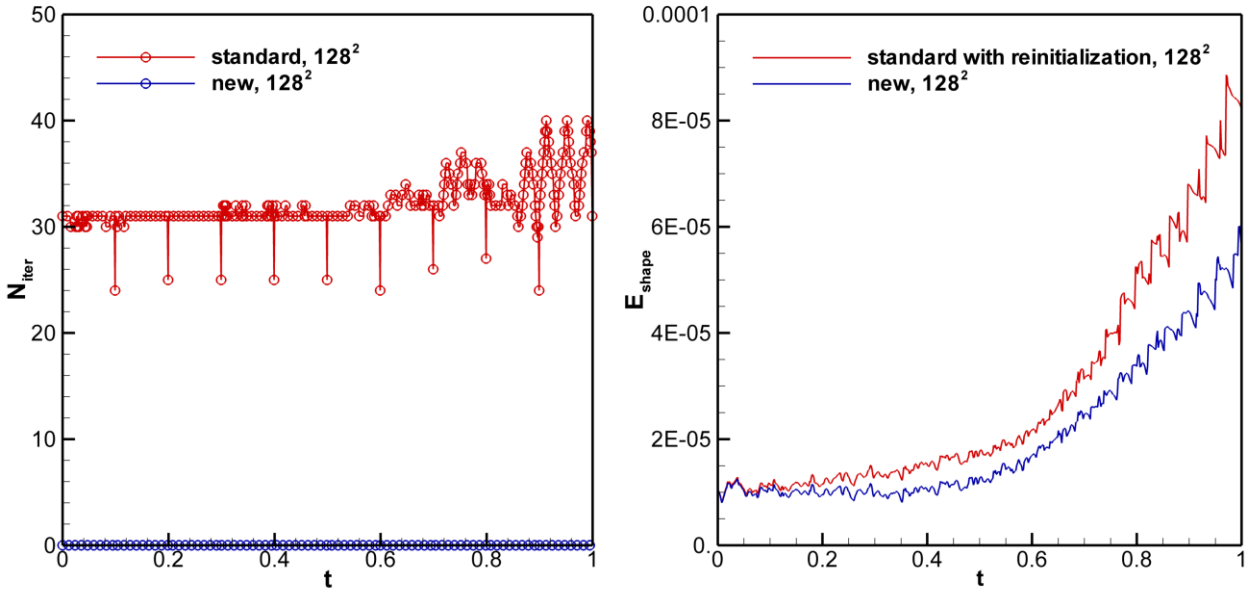


Figure 4.10: Case (ii): Number of iterations employed in re-initialization procedure for standard and new approach.

Table 4.2. Case (ii): mean shape error for different meshes at time $t=1.0$

mesh	64×64	128×128	256×256	512×512
standard no reinitialization	1.28×10^{-4}	3.26×10^{-5}	7.88×10^{-6}	1.93×10^{-6}
standard with reinitialization	6.71×10^{-4}	8.23×10^{-5}	7.91×10^{-6}	3.11×10^{-6}
new approach	4.67×10^{-4}	5.69×10^{-5}	5.89×10^{-6}	5.81×10^{-7}

We complete assessment of this case by important remark about discretization of the source term. As it was noted before, in all our calculations we used the 5^{th}-order WENO scheme with global Lax-Friedrichs approximation for Hamiltonian (Jiang & Peng, 2000). Our experience showed that discretization of the source term by central schemes often leads to unstable results. In Fig. 4.11, results obtained by 4^{th}-order central scheme are compared versus 5^{th}-order WENO scheme. In this figure, the results for distribution of the level set function and of its gradient norm (in color) are shown. Central difference scheme is presented on the left, and WENO scheme on the right. For central difference scheme we see that oscillations occur in points of discontinuity in $\nabla\varphi$-distribution, or skeleton points. From other side, WENO scheme shows stable results. Due to numerical diffusion segment of skeleton points is smeared out by 3-4 grid cells. It is expected that the use of less diffusive approximations for Hamiltonian, such as Roe flux or Godunov flux, will give a sharper resolution of segment with skeleton points.

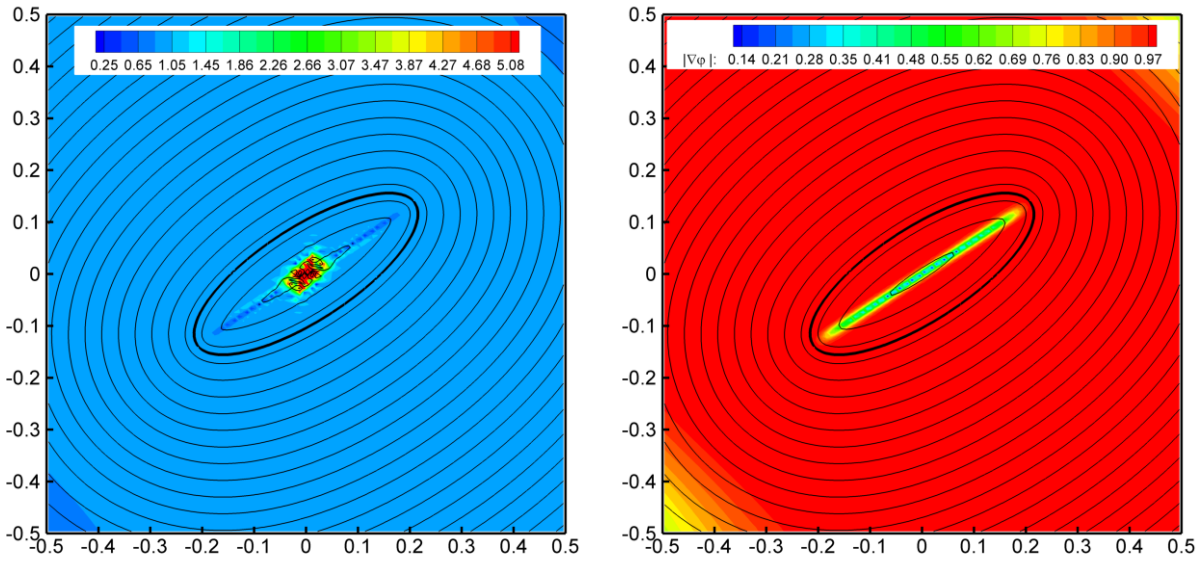


Figure 4.10: Isolines of the level set function and its gradient norm (in color) for new approach at time $t = 0.5$. On the left: 4^{th}-order central scheme; on the right: 5^{th}-order Hamilton-Jacobi WENO scheme.

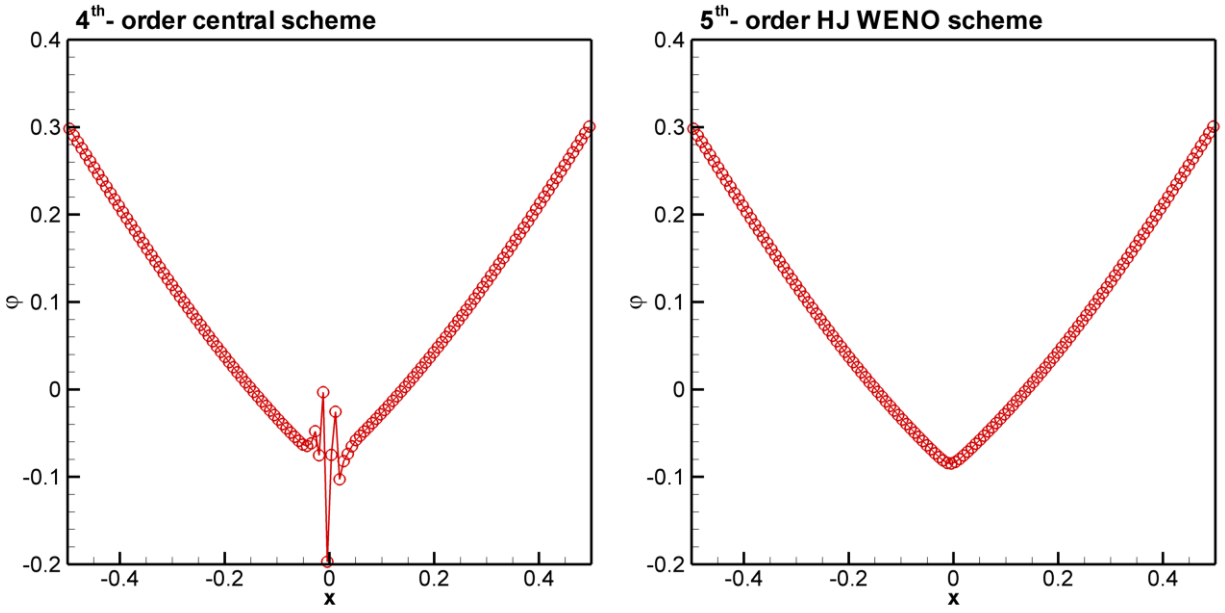


Figure 4.12: Profile of the level set function along $y = 0.5$ for new approach at time $t = 0.5$. On the left: 4th-order central scheme; on the right: 5th-order Hamilton-Jacobi WENO scheme.

4.3 Interface stretching by single vortex

For this classical test, the conditions are taken from Rider & Kothe (1998). The initial zero level set is a circle, centred at $(x, y) = (0.5, 0.75)$, and with radius $R = 0.15$:

$$\varphi_0(x, y) = \sqrt{(x-0.5)^2 + (y-0.75)^2} - R \tag{4.25}$$

This interface is stretched in a computational domain $\Omega : [0,1] \times [0,1]$ by an external flow field which is defined by the following stream function:

$$\psi(x, y) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \tag{4.26}$$

After time $t = 3.0$ we change the velocity sign, and our calculations are continued until time $t = 6.0$. Thereby the aim is to observe how the zero level set is reversed to its initial position. Computations with new approach were performed using the zero- order and the first-order local approximations (equations (3.34) and (3.35), respectively), as well the exact expression of the source term coefficient, i.e. (3.33). In all computations, the CFL number (based on the maximum velocity) is set to 0.5. First, the reference solution is obtained on refined mesh 1024×1024 by

standard level set approach without reinitialization. The initial configuration and the reference solution at $t = 3.0$ is given in Fig. 4.13.

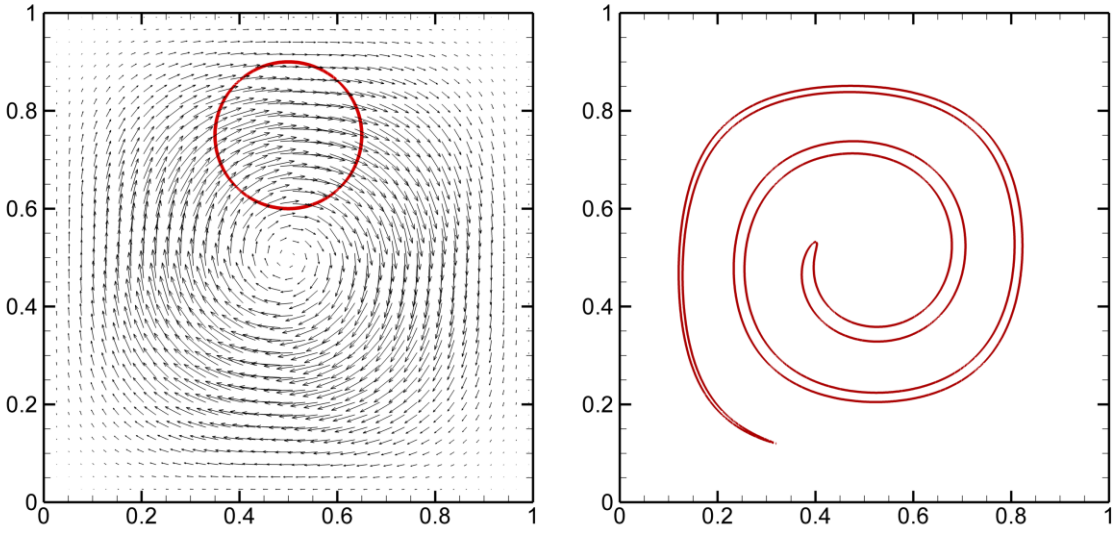


Figure 4.13: Single vortex test. On the left: initial zero level function and the velocity field plot; on the right: reference solution at $t = 3.0$ obtained by 1024×1024 computational grid without re-initializations of level sets.

At time $t = 3.0$, we compared this reference solution with computations on a coarser mesh containing 256×256 grid points. The computations included the standard approach with the re-initialization procedure, and the new one with different expressions of the source term coefficient. In Fig. 4.14, fifteen isolines of level set function are shown in the domain of narrow band; the thick red line represents the zero level set of φ. It is seen that though the zero-order local approximation (3.34) gives resolution similar to the standard method, the first-order local approximation (3.35) and the exact expression of the source term (3.33) resolve the stretched filament fairly better. However, when we observed the reversed shape at time $t = 6.0$, all four methods produced approximately the same configurations of zero level set, with still smoother shapes in cases „source term 2" and „source term 3". The results are demonstrated in Fig. 4.15.

The convergence criterion in the re-initialization procedure for all appraising here approaches is taken the same, but the number of iterations is observed different: an advantage of the new method is manifested by smaller number of iterations required for the convergence criterion. For different times, Fig. 4.16 shows a comparison of iterations number in standard approach and in the case when the first-order local approximation (3.35) is applied. The maximal number of iterations was limited by 30. It is seen that until time $t = 0.5$ the new approach requires a very small number of re-initializations compared to 19 iterations in standard approach.

After this time, a strong deformation of the front requires more iteration in both approaches, but still in the case of the new approach the iteration number is considerably reduced. In Table 4.3, the mean number of iterations, the mean computational cost per time-step and the mean shape error are demonstrated for all four approaches employed. In comparison with standard approach, it is seen that computational efficiency is progressively improved when successively higher order of local approximation is employed by (3.33) - (3.35).

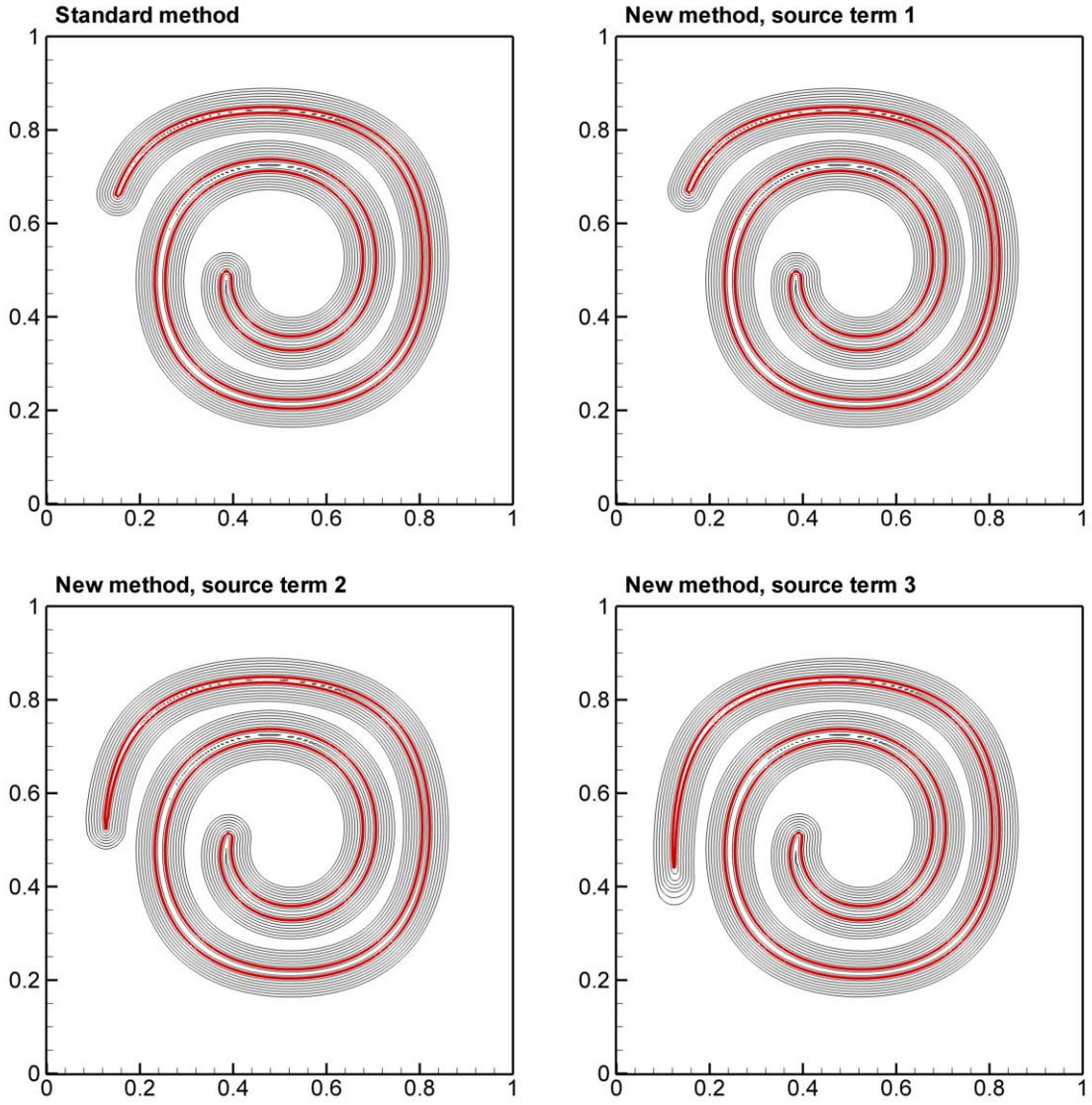


Figure 4.14: Single vortex test. Isolines of level set function at $t = 3.0$ using standard (with re-initializations) and new approaches. New approach: the zero-order local approximation - source term 1; the first-order local approximation - source term 2; the exact expression of the source term coefficient - source term 3. Here, 15 isolines of φ are shown in the narrow band domain; thick red line represents the zero level set of φ. Grid resolution: 256×256 points.

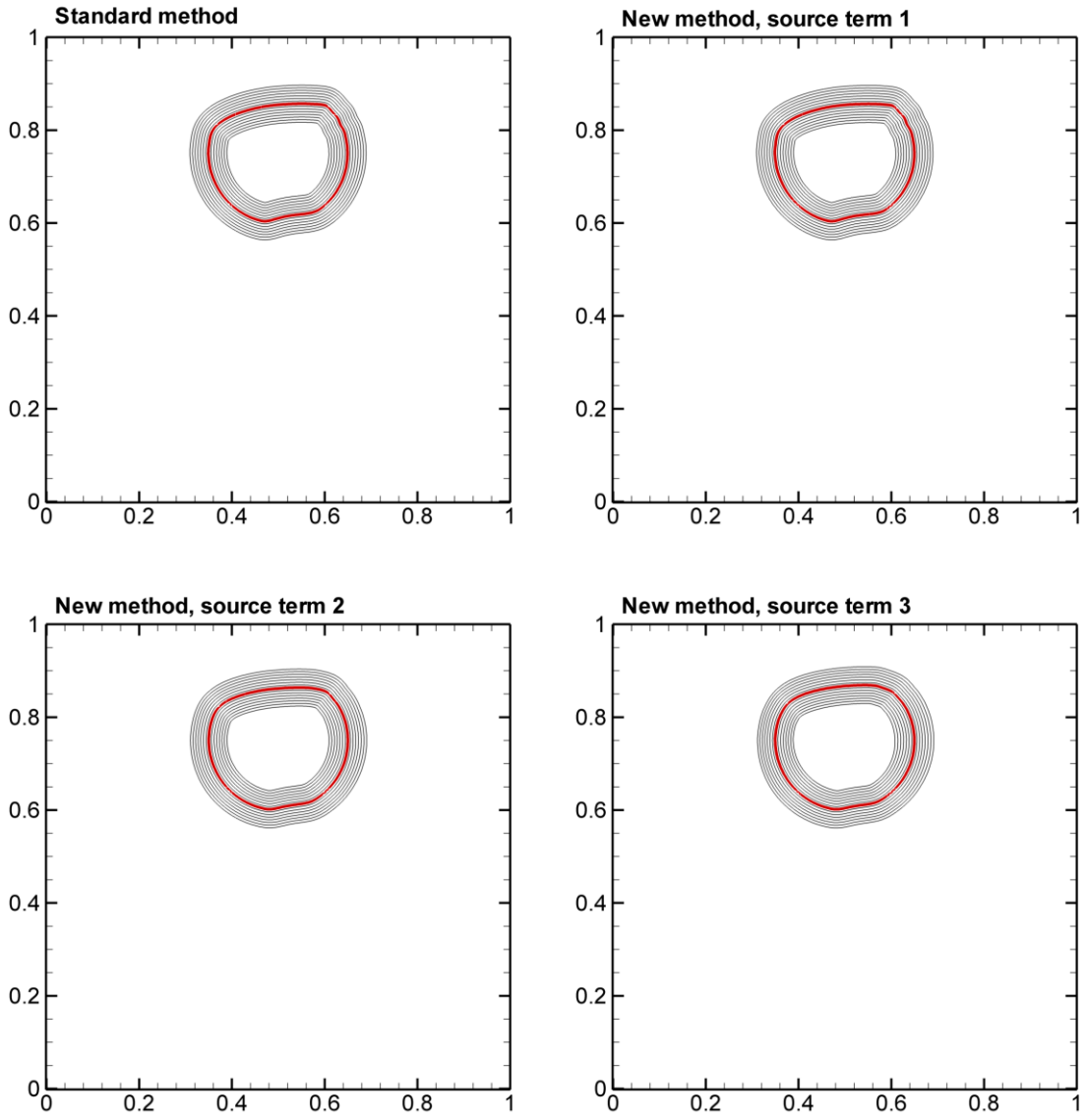


Figure 4.15: The reversed shape at time $t = 6.0$ in single vortex test. Isolines of level set function obtained by standard (with re-initializations) and new approaches. New approach: the zero-order local approximation - source term 1; the first-order local approximation - source term 2; the exact expression of the source term coefficient - source term 3. Here, 15 isolines of φ are shown in the narrow band domain; thick red line represents the zero level set of φ. Grid resolution: 256×256 points.

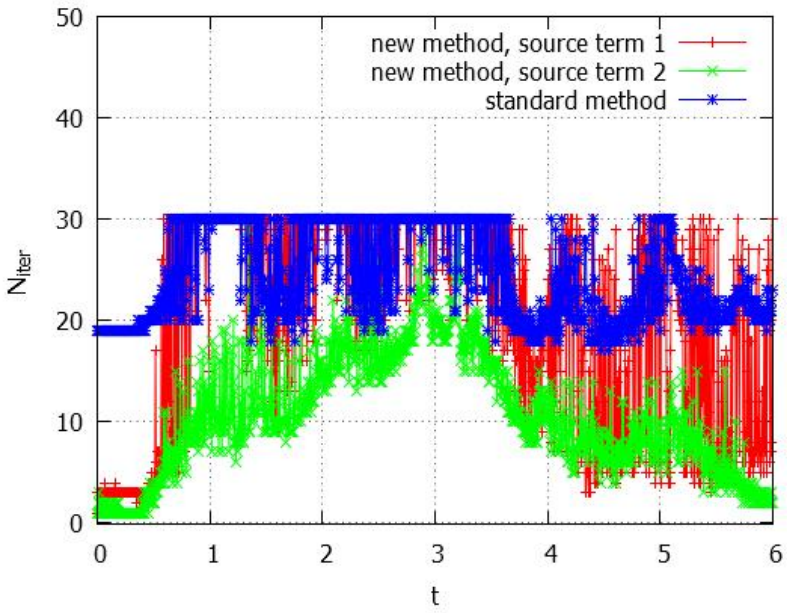


Figure 4.16: Single vortex test. Number of iterations employed with time in the re-initialization procedure; new approach is presented by the first-order local approximation.

Table 4.3: Single vortex test. Averaged number of iterations, mean computational cost per time-step and mean shape error. Grid resolution: 256×256 points.

Method	$<N_{iter}>$	CPU time, sec	shape error
standard method	24.44	0.4521	6.76×10^{-2}
“source term 1”	20.16	0.3954	6.74×10^{-2}
“source term 2”	10.46	0.2317	4.93×10^{-2}
“source term 3”	-	0.2878	4.21×10^{-2}

4.4 Oscillating circle test

This test case was proposed in Hartmann *et al*. (2008): in a computational domain $\Omega:[-5,5]\times[-5,5]$, the initially circular interface, with radius $R=3$ and with center $(x,y)=(0,0)$, is represented by level set function:

$$\varphi_0(x,y)=\sqrt{x^2+y^2}-R. \tag{4.27}$$

The level set function is subject to a presumed velocity field, in which only the normal component is non-zero. The modulus of this component is defined by trigonometric functions of azimuth and time:

$$\vec{u}=s\vec{n},\ s=\cos(8\theta)\sin(\omega t),\ \theta=arctg\left|\frac{y}{x}\right|,\ \omega=\frac{2\pi}{T} \tag{4.28}$$

The period of oscillation is set to $T=5$. Parameters of simulation are chosen the same as in the original paper (Hartmann *et al*., 2008). The time step $\Delta t=\Delta x/4$ corresponds to CFL number of 0.25, and the grid resolution is based on 256×256 points. The normal vector $\vec{n}$ is approximated using the second-order centered difference scheme.

The oscillating circle test is a particularly difficult test, since the interface shape is strongly coupled with the interface motion. Small perturbations of the normal to interface vector $\vec{n}$ may lead to errors in level set function φ, and consequently, to modification of the external velocity vector $\vec{u}$. As it was noted in (Hartmann *et al*., 2008), this test case requires a large number of re-initializations of the level set function. Otherwise, when standard approach is used without re-initialization, the interface shape suffers from strong oscillations.

The results of computation and comparison between appraising approaches are illustrated in Fig. 4.17 and Fig. 4.18 at times $t=2.5$ and $t=5.0$, respectively. Time $t=5.0$ corresponds to return of convected interface back to its initial circled shape. Here, ten iso-lines of φ are shown in the narrow band domain, and the thick line represents the zero level set. Concerning time $t=2.5$, all considered approaches give the same configuration. However an advantage of new approach is clearly seen in Fig. 4.18. This figure exhibits the returned circled shape at $t=5.0$. It is seen that including the zero-order local approximation, the modified level set equation leads to the better predicted circled shape. In the next figure, Fig. 4.19, such an improvement is demonstrated by calculation of curvature along the returned shape at $t=5.0$. It is seen that spurious modes, developed in computation of curvature by standard approach, are remarkably reduced in the new approach.

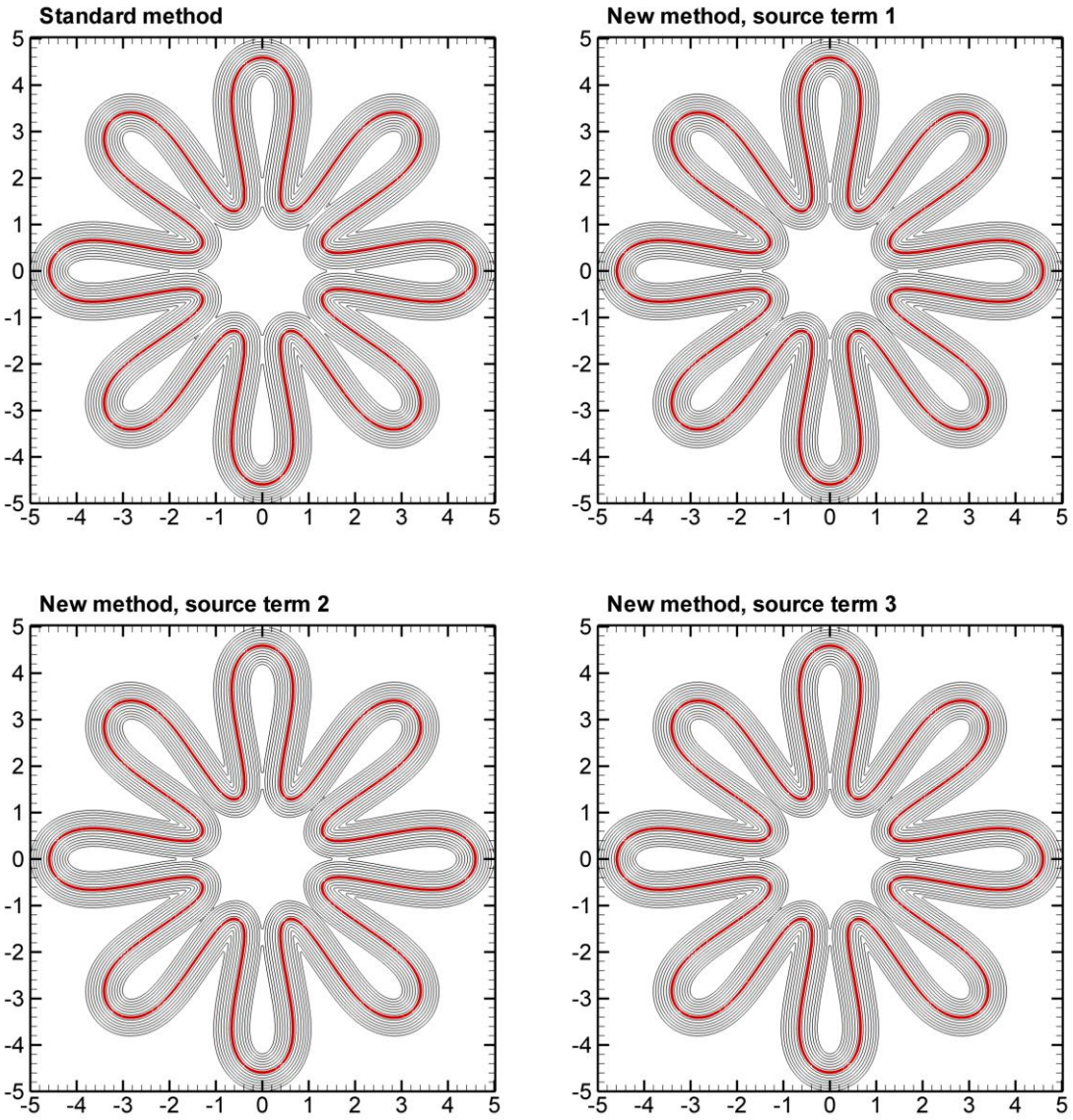


Figure 4.17: Oscillating circle test. Isolines of level set function are computed by standard and new approaches. New approach: zero-order local approximation - source term 1; first-order local approximation - source term 2; exact expression of the source term coefficient - source term 3. Here, 10 isolines of φ are shown in the narrow band domain, thick red line represents the zero level set of φ. Grid resolution: 256×256 points.

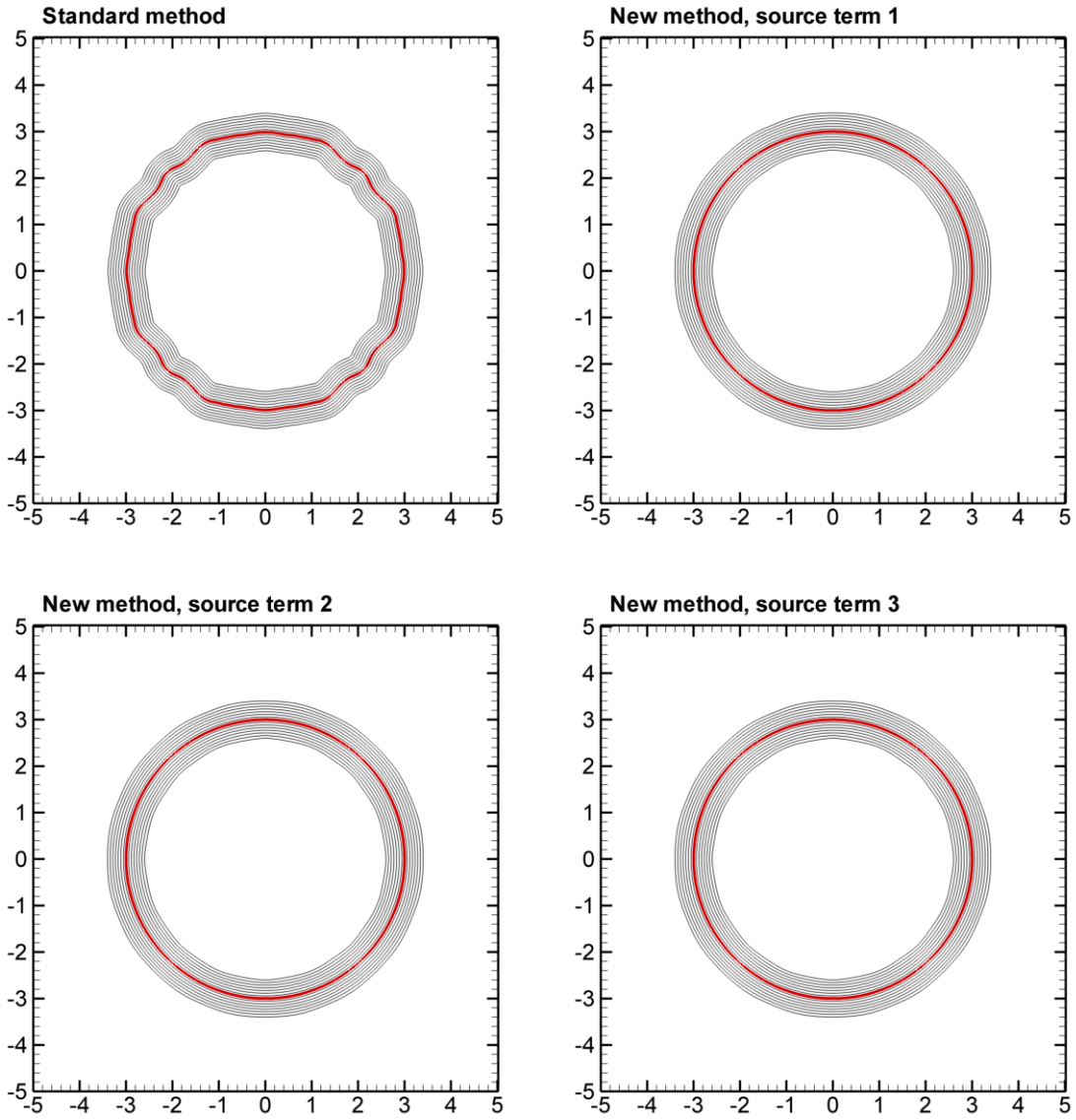


Figure 4.18: Return to the circled shape ($t = 5.0$) in oscillating circle test. Isolines of level set function are computed by standard and new approaches. New approach: zero-order local approximation - source term 1; first-order local approximation - source term 2; exact expression of the source term coefficient - source term 3. Here, 10 isolines of φ are shown in the narrow band domain, thick red line represents the zero level set of φ. Grid resolution: 256×256 points.

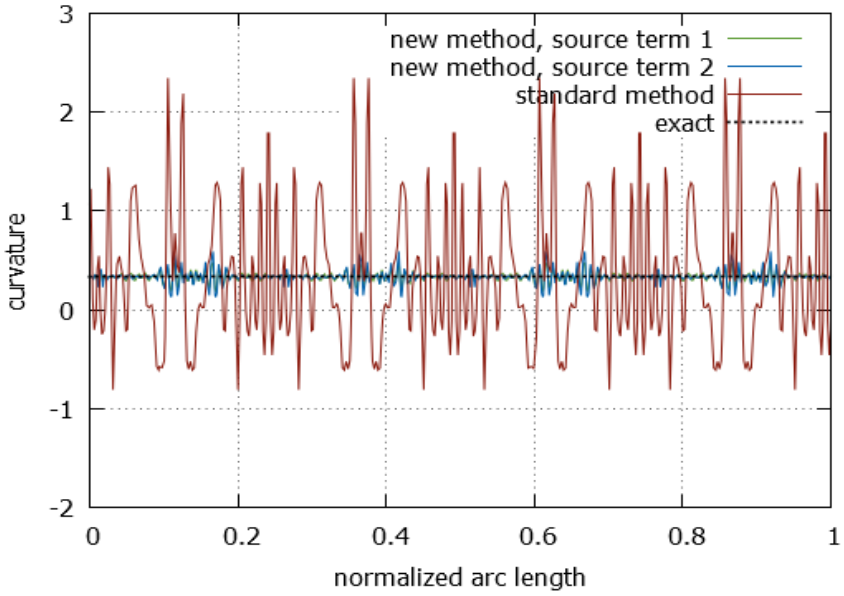


Figure 4.19: Interface curvature in the returned circled shape ($t = 5.0$) in oscillating circle test.

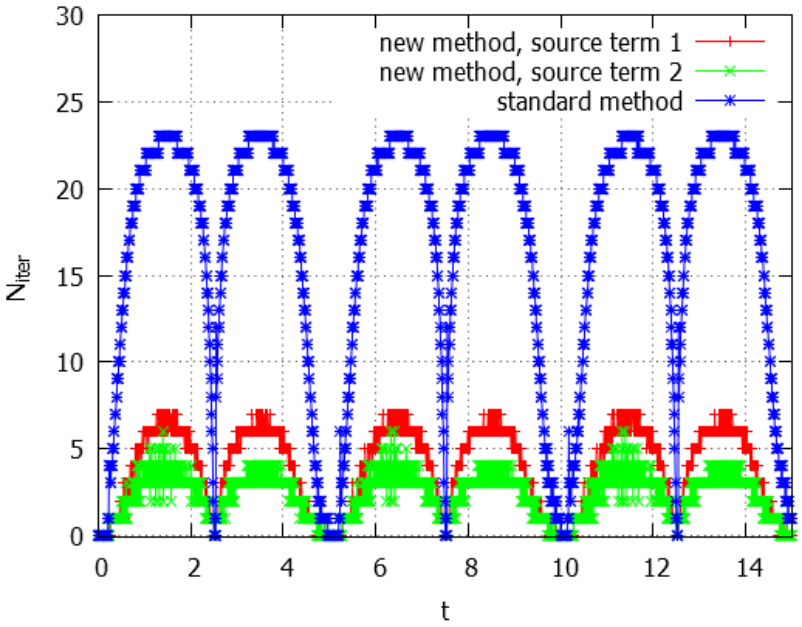


Figure 4.20: Oscillating circle test. Number of iterations in the reinitialization procedure employed with time in appraising approaches.

We observed it at different grid resolutions: 64×64; 128×128; 256×256; 512×512, and we compared our computations with results issued from the method proposed in (Hartmann *et al*., 2010). The mean shape error is given in Table 4.4 from standard approach, from zero-order local approximation and from computation in (Hartmann *et al*., 2010). Starting from 128×128 grid points, the zero-order local approximation gives the mean shape error an order less compared to other results. Additionally to this improved computation of curvature, the number of iterations in the re-initialization procedure is approximately five times less in the new approach than in the standard one. This is shown in Fig. 4.20 for the zero- and first-order local approximations, as well as for standard approach. As expected, it is seen in this figure that higher order local approximation requires less of re-initializations. Table 4.5 concludes the oscillating circle test: the mean number of iteartions, the mean computational cost per time-step and the mean shape error are given for the returned shape at $t=5.0$ applying standard approach and the zero- order local approximation (3.34). Computational efficiency of later is clearly seen.

Table 4.4: Oscillating circle test. Mean shape error at $t=5.0$ for different grid resolution.

Method	64×64	128×128	256×256	512×512
standard approach	1.57×10^{-2}	6.72×10^{-3}	3.10×10^{-3}	1.07×10^{-3}
source term 1	1.22×10^{-2}	5.65×10^{-4}	4.74×10^{-5}	3.87×10^{-5}
Hartmann *et al.* (2010)	7.40×10^{-3}	2.58×10^{-3}	1.26×10^{-3}	6.26×10^{-4}

Table 4.5: Oscillating circle test. Averaged number of iterations, mean computational cost per time-step and mean shape error. Grid resolution: 256×256 points.

Method	$<N_{iter}>$	CPU time, sec	shape error
standard method	16.46	0.5758	3.10×10^{-3}
source term 1	3.89	0.1960	4.74×10^{-5}

4.5 Propagation of the premixed flame front

Another interesting example, when the interface configuration is coupled with its advancement, is the propagation of the premixed flame in the framework of G-equation approach. In one-dimensional formulation, the flame front is governed by the following evolution equation:

$$\frac{\partial \varphi}{\partial t} + \vec{u} \cdot \nabla \varphi = u_F \left| \nabla \varphi \right| \tag{4.29}$$

with the following initial condition:

$$\varphi_0(x, y) = -x + x_0. \tag{4.30}$$

Here u_F is a proper speed of the flame front, $\vec{u}$ determines the external flow field, x_0 corresponds to initial position of the flame front. It is assumed that $\varphi(x, y, t) > 0$ in the burnt gases. Consequently, the flame propagates from the left to the right. At initial time, according to (4.33), all level surfaces of $\varphi(\vec{x}, t)$ are planes normal to the x direction. Another form of (4.29) is this:

$$\frac{\partial \varphi}{\partial t} + (\vec{u} + u_F \vec{n}) \cdot \nabla \varphi = 0 \tag{4.31}$$

It is seen that advection of $\varphi(\vec{x}, t)$ is controlled by prescribed velocity field and by the front propagation speed, which in turn, depends on the normal vector $\vec{n}$. In our calculation we presumed the shear velocity field:

$$\vec{u} = (u, v)^T = (\lambda(1 + \eta \cos(ky)) \cos(2ky), 0)^T \tag{4.32}$$

As previously, a computational domain $\Omega : [0,1] \times [0,1]$ is discretized by uniform grid. Parameters of calculations are as follows: the flame front speed is $u_F = 1.0$, the shear intensity parameter is $\lambda = 0.4$, the intensity of modulation $\eta = 0.5$, and the wave number is $k = 8\pi$. The final time of computation is $t = 0.5$.

Fig. 4.21 shows the solution of (4.31) at different times obtained in the whole computational domain on the fine grid 512×512 by standard approach without re-initialization.

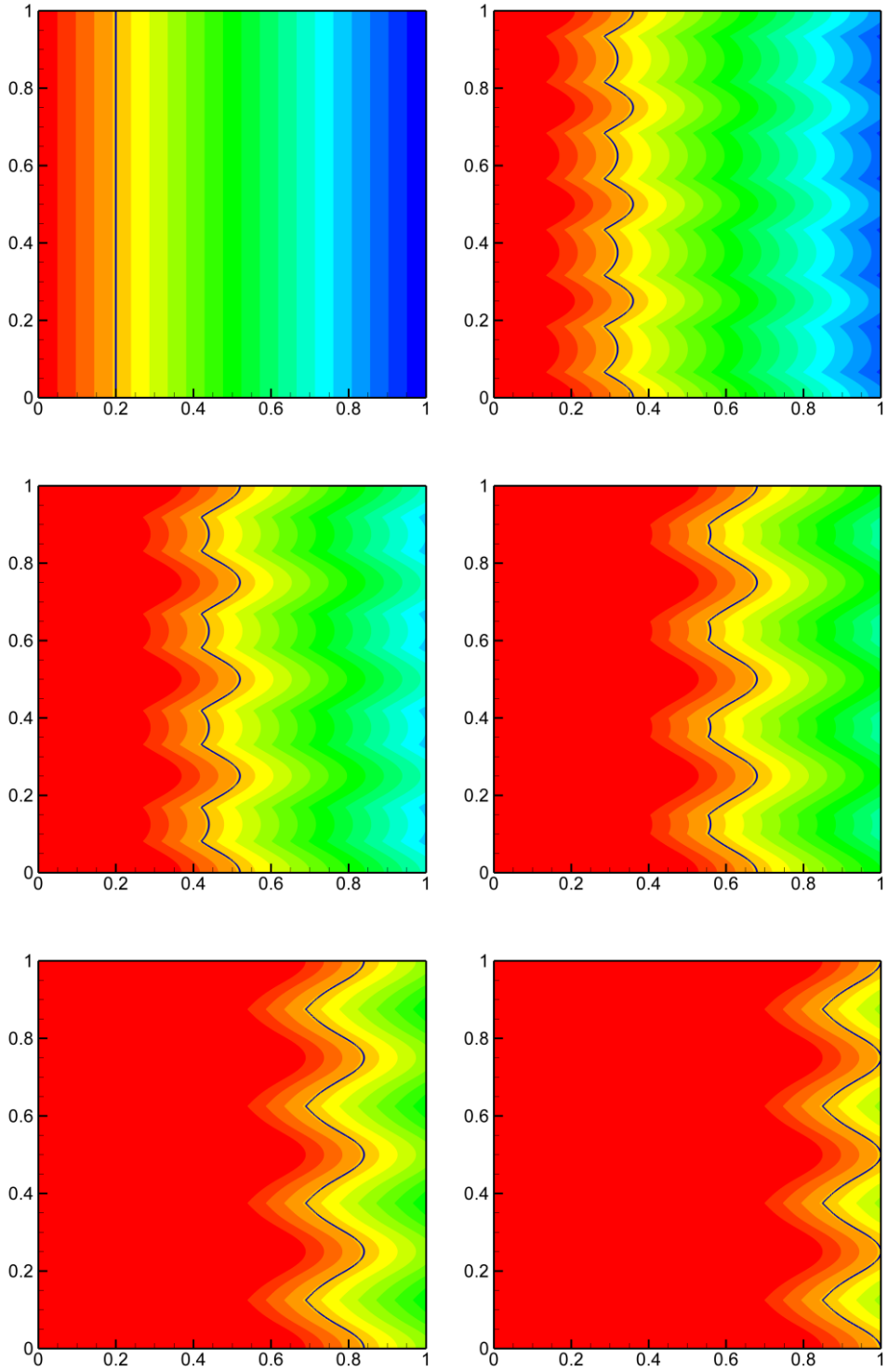
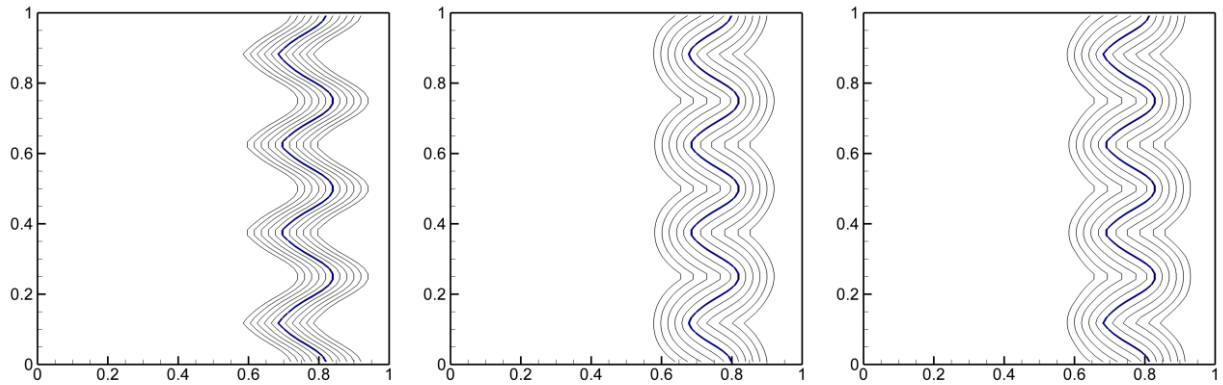


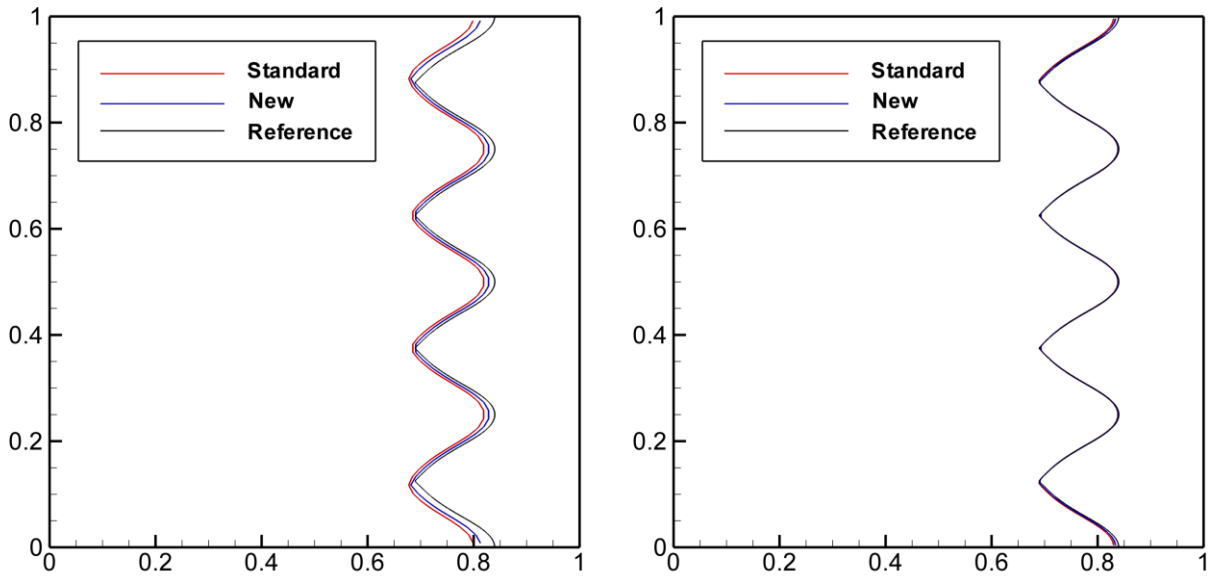
Figure 4.21: Flame propagation test. The reference solution at different times obtained without re-initializations of level sets by standard approach in the whole domain with 512×512 points.

In Fig. 4.22, three plots of the premixed flame front are shown from numerical integration of (4.31): results on the left are obtained on the mesh 512×512 by standard approach without re-initializations (referred to as reference solution); in the middle - on the mesh 64×64 by standard approach with re-initializations; on the right - on the mesh 64×64 by the zero-order local approximation. Below these plots, three fronts are put together for comparison. A slight difference in the front location is seen: computations with new approach give the front location slightly closer to reference solution than in the case of standard approach. The same was observed for other times.



reference solution (512×512) | standard, re-initializations (64×64) | new approach, source term 1 (64×64)

Figure 4.22: Flame propagation test. Isolines of the level set function at time t = 0.4. Interface location is represented by the thick line; on the bottom: three fronts are put together.



64×64 128×128

Figure 4.23: Flame propagation test. Interface location at time t = 0.4. Comparison of the standard re-initialization approach and new approach with reference solution. Grid resolution: 64×64 (left plot) and 128×128 (right plot).

The difference is also seen from the curvature computation at the leading edge of the front in Fig. 4.24. The cusped zones of the front are better resolved by new approach. However in general for this test case, the difference between approaches is not very pronounced. In the future, this motivates to study the more complex situations with premixed flames. What is remarkable at the moment, it is the number of iterations employed in the reinitialization procedure. This number is fairly different: the new approach requires much smaller number of iterations, and as in the previous test, the number of iterations is decreasing with application of higher order of the local approximation to the source term coefficient. This is illustrated in Fig. 4.25.

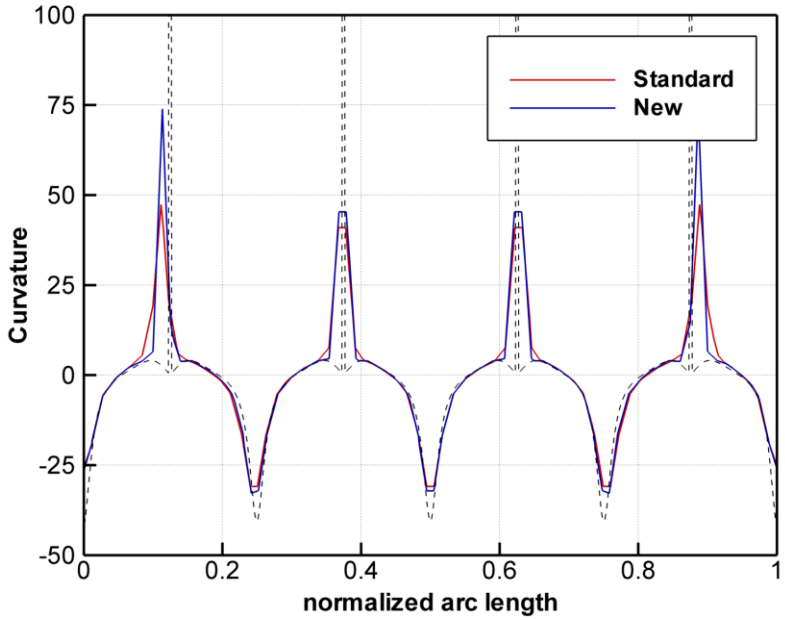


Figure 4.24: Flame propagation test. Interface curvature at t = 0.4 for the standard re-initialization and new approach on 64×64 grid. Dashed line corresponds to the reference solution.

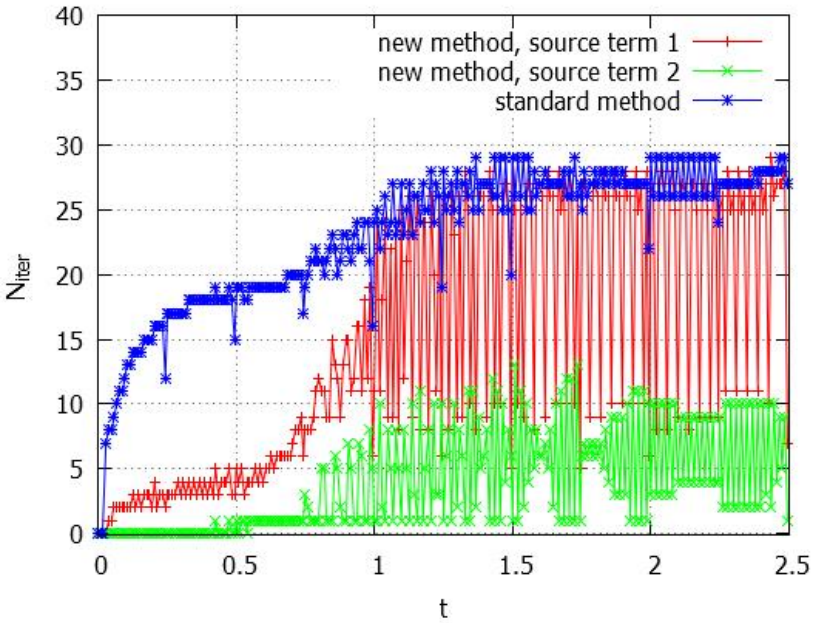


Figure 4.25: Flame propagation test. Number of iterations employed with time in the re-initialization procedure.

4.6 Assessment of different numerical schemes for advection term

4.6.1 Convergence rate by different WENO schemes in the single vortex test

In this Section, we compare different WENO schemes described in Chapter 2. Our computations here are based only on the standard form of the level set equation without re-initializations, and without source term on the right-hand side. Both forms of the level set equation are considered: the first one is the divergent form $\partial_t \varphi + \nabla \cdot (\vec{u}\,\varphi) = 0$ (we will refer as conservative form), and the second one is the advective form $\partial_t \varphi + \vec{u} \cdot \nabla \varphi = 0$ (non-conservative).

For single vortex test, Fig. 4.26 and Fig. 4.27 show the interface location for different schemes at times $t = 3.0$ and $t = 6.0$, respectively. Dashed line corresponds to the reference solution. We see that the choice of advection schemes has a strong influence on the interface resolution. The most resolved solution was obtained with WENO-Z scheme in conservative form. Figure 4.28 illustrates the capacity of different schemes to preserve the area enclosed by the zero level function. For example at the moment of maximum stretching area error for the standard WENO-JS (n-cons.) is about 40%, while WENO-SYM and WENO-Z schemes give only 2-3%. For comparison, the particle level set method of Enright (2002) gives 0.71% of error on the same grid resolution.

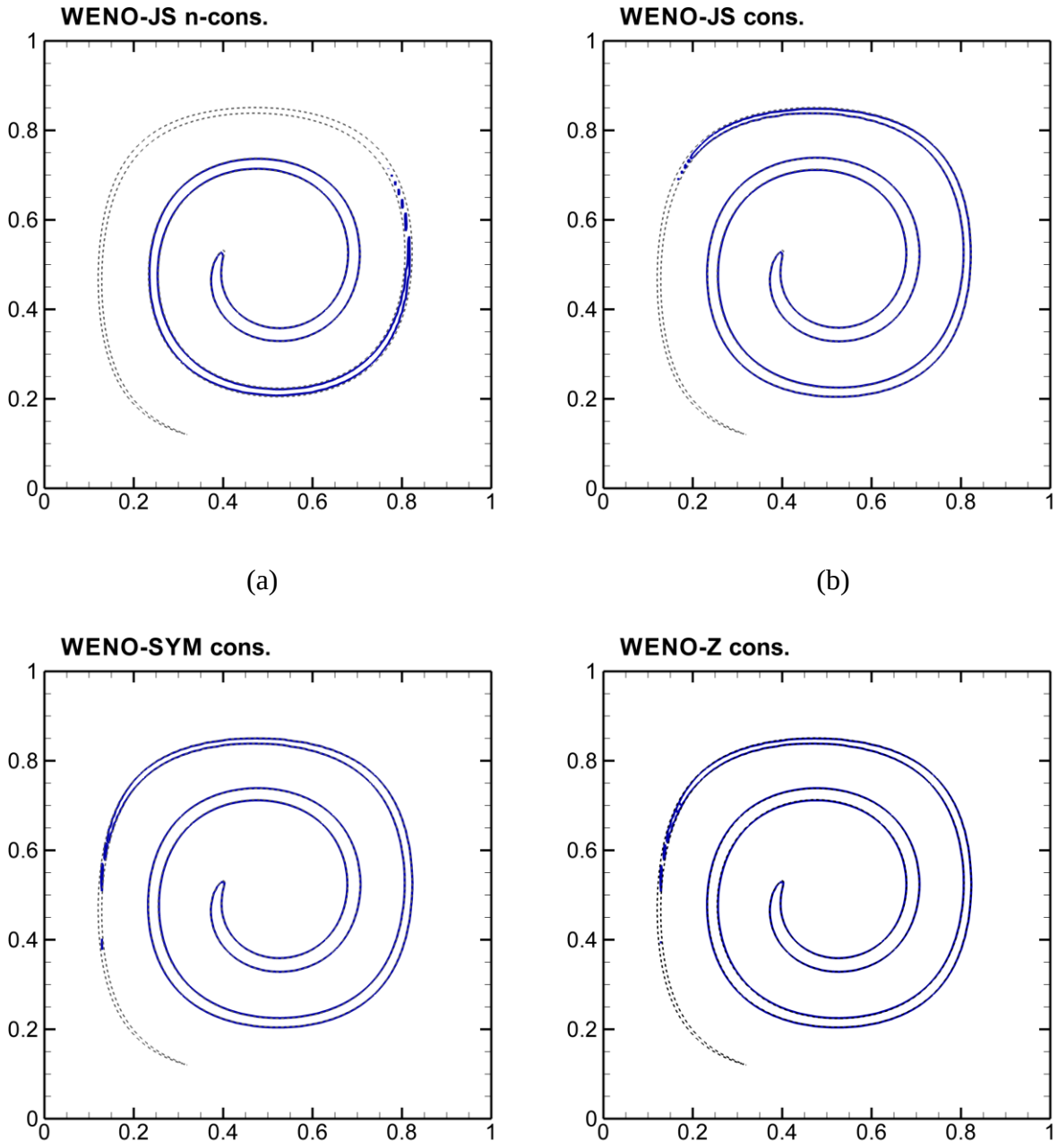


Figure 4.26: Single vortex test. Interface at time $t = 3.0$. Comparison of different schemes for convective term in the level set equation: (a) WENO-JS n-cons., (b) WENO-JS cons, (c) WENO-Z cons. and (d) WENO-Z cons. Grid resolution 128×128 points.

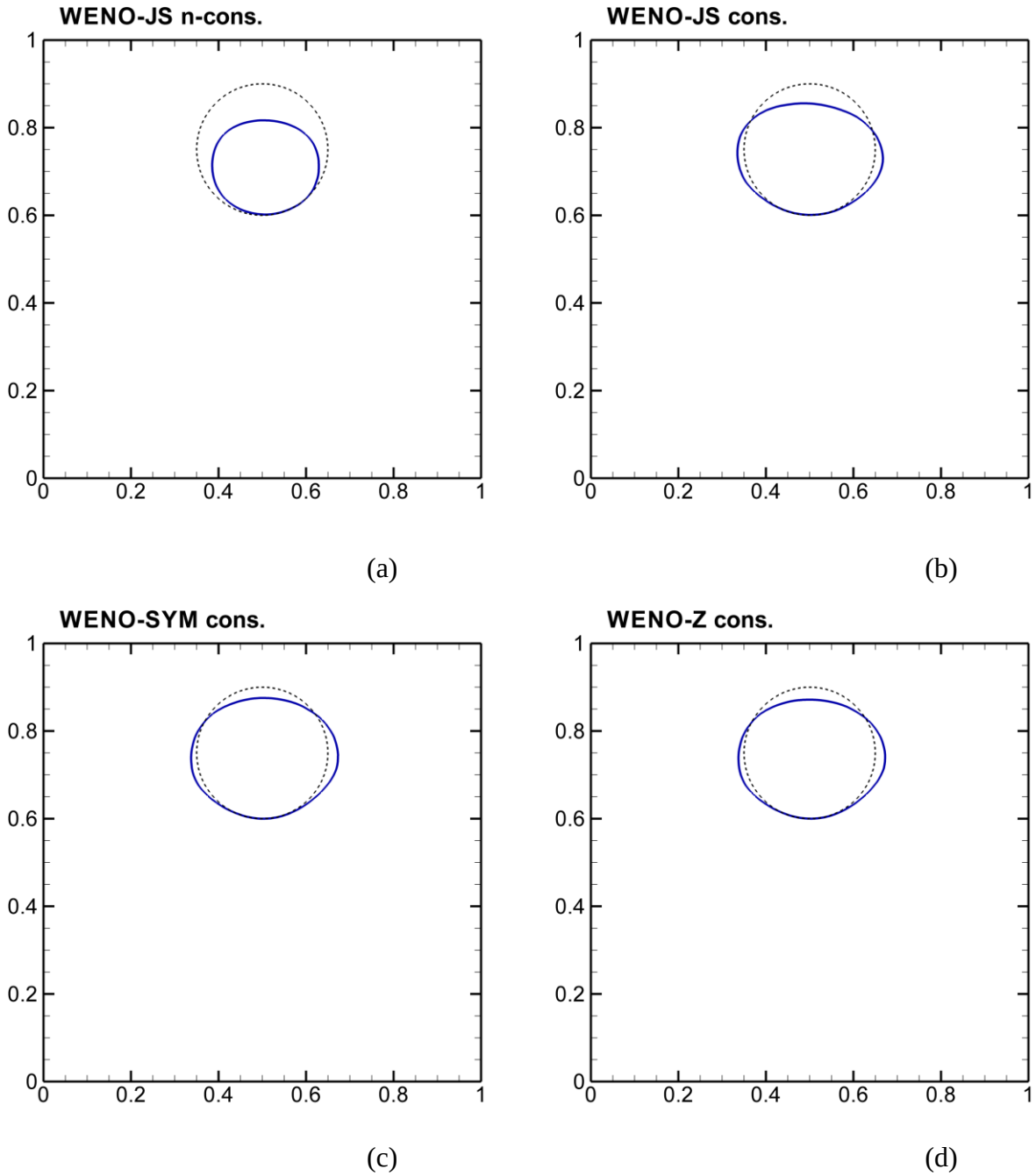


Figure 4.27: Single vortex test. Interface at time $t = 6.0$. Comparison of different schemes for convective term in the level set equation: (a) WENO-JS n-cons., (b) WENO-JS cons, (c) WENO-Z cons. and (d) WENO-Z cons. Grid resolution 128×128 points.

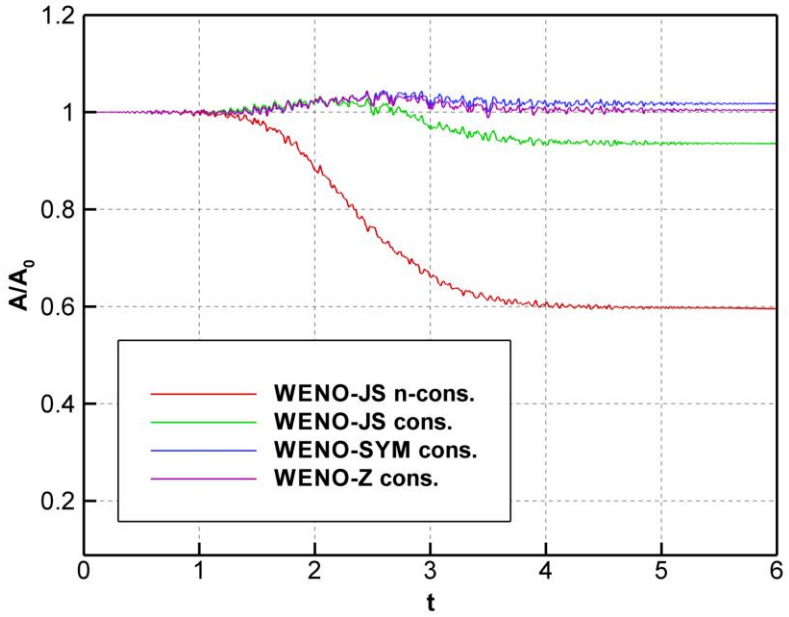


Figure 4.28: The temporal evolution of the normalized area in single vortex test.. Grid resolution: 128×128 points.

4.6.2 Zalesak's test

The next step is examination of the ability in applied schemes to resolve the sharp corners and the thin structures. To this end, we consider the widely used test problem proposed by Zalesak (1979). This test represents a rigid body rotation of disk with a thin slot in prescribed time-independent velocity field. The initial interface has a shape of slotted circle centered at $(x, y) = (50, 75)$ with radius $R = 15$, the slot width of 5 and the slot length of 25. The corresponding initial data for the level set function is the following:

$$\varphi_0(x, y) = \max\left\{\min\left\{x_c + 2.5, -x_c + 2.5, -y_c + 10\right\}, (x_c^2 + y_c^2)^{1/2} - R\right\} \tag{4.33}$$

where $x_c = x - 50,\ y_c = y - 75.$ The velocity field is given by

$$u = \frac{\pi}{314}\left(50 - y\right), \quad \mathrm{v} = \frac{\pi}{314}\left(x - 50\right), \tag{4.34}$$

This velocity fields provides the disk to complete one revolution every 628 time units. In this case, the right hand side in (3.9) is equal to zero, $n_i \dfrac{\partial u_k}{\partial x_i} n_k = 0$, and consequently, the source term in the modified level set equation is as well equal to zero. This implies that in the Zalesak's problem the both approaches, new and standard, have identical formulation, and hence, the numerical results are given hereafter in the framework of standard form of the level set equation, without re-initialization procedure.

Computational domain is a square [0,100]×[0,100] discretized on 100×100 grid. Only 5 cells are used to resolve the disk's slot. Similarly as for previous test problem (Section 4.2) we can obtain the exact solution. It is given by

$$\varphi(x, y, t) = \varphi_0(\xi_1, \xi_2), \tag{4.35}$$

where the characteristic variables are

$$\begin{aligned} \xi_1 &= \left(x - 50\right)\cos\left(\lambda t\right) + \left(y - 50\right)\sin\left(\lambda t\right), \\ \xi_2 &= \left(y - 50\right)\cos\left(\lambda t\right) - \left(x - 50\right)\sin\left(\lambda t\right), \\ \lambda &= \pi / 314. \end{aligned} \tag{4.36}$$

The initial configuration and the exact solution at different times are given in Fig. 4.29.

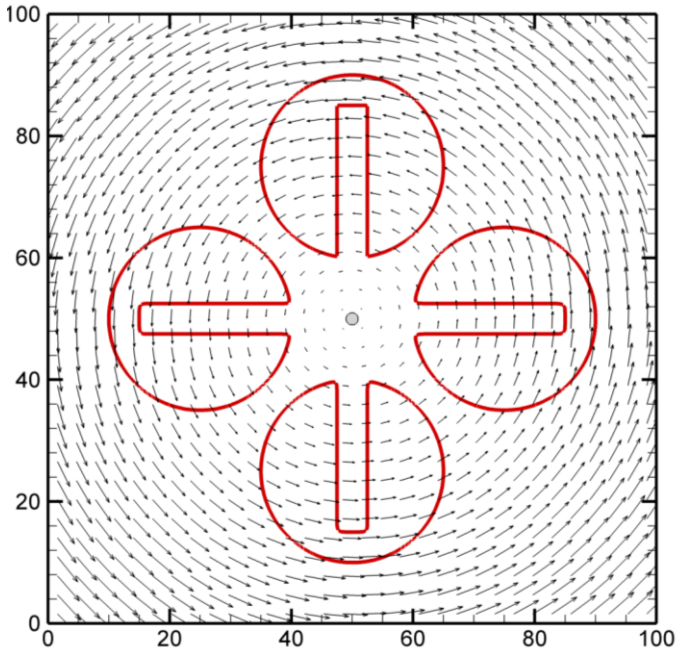


Figure 4.29: The velocity field and exact interface location at different times in the Zalesak's test.

First, as for previous test case, we compare different level set transport schemes, namely WENO-JS n-cons., WENO-JS cons., WENO-SYM cons. and WENO-Z cons schemes. Fig. 4.30 and Fig.4.31 compare the interface location after one and ten full rotation, respectively. Results for different level set transport schemes are presented here and compared with the exact solution (dashed line). It is clearly seen that WENO-SYM cons. and WENO-Z cons. lead to more accurate solution. This is due to increased accuracy of these schemes in comparison to WENO-JS, and due to the fact that the conservative form of the level set equation has been used. For different advection schemes, Fig. 4.32 shows the evolution in time of the normalized area enclosed by the interface. It can be seen that for this test problem the conservation errors remain very small for all considered schemes. The maximum deviation is less than 0.2%. For comparison, the same error was obtained with the ACLS method (Desjardins *et al.*, 2008), where the error was estimated by 0.0352%.

The next step consists in estimation of grid convergence. This analysis is performed using both 50×50 and 200×200 grids in addition to the 100×100 grid applied earlier. The CFL number is kept constant (0.5) for the different cases. Table 4.7 summarizes the L_2 norm of error for φ and the convergence rate for three grid resolutions. As previously, it can be seen that WENO-SYM and WENO-Z schemes lead to more accurate solution. However, as expected due to sharp corners in the solution, all schemes give at most the first order of convergence rate. Discontinuities in the gradient of solution to Hamilton-Jacobi equation affect the convergence

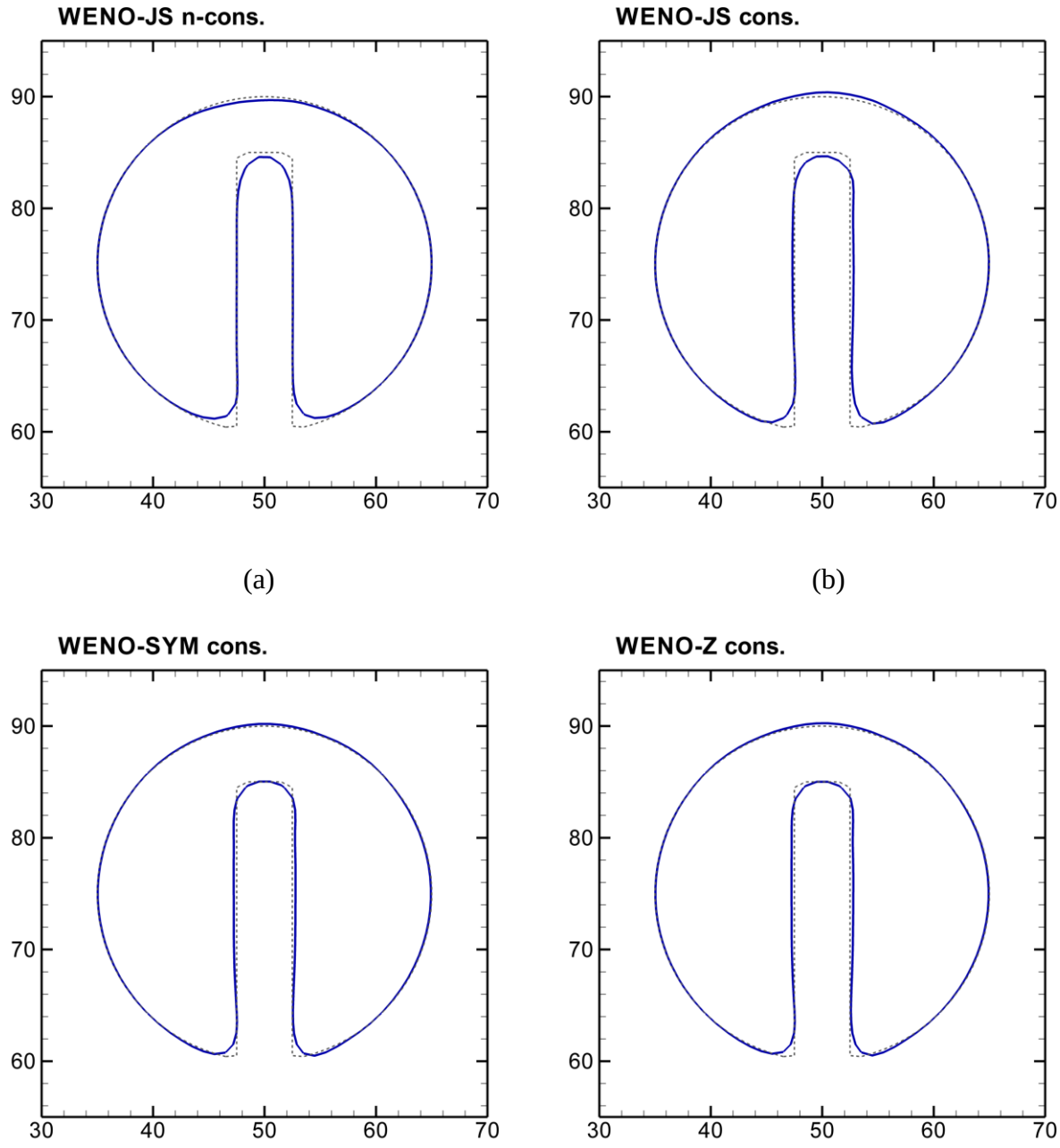


Figure 4.30: Zalesak's test. Interface after 1 full rotation. Comparison of different schemes for convective term in the level set equation: (a) WENO-JS n-cons., (b) WENO-JS cons, (c) WENO-Z cons. and (d) WENO-Z cons. Grid resolution 100×100 points.

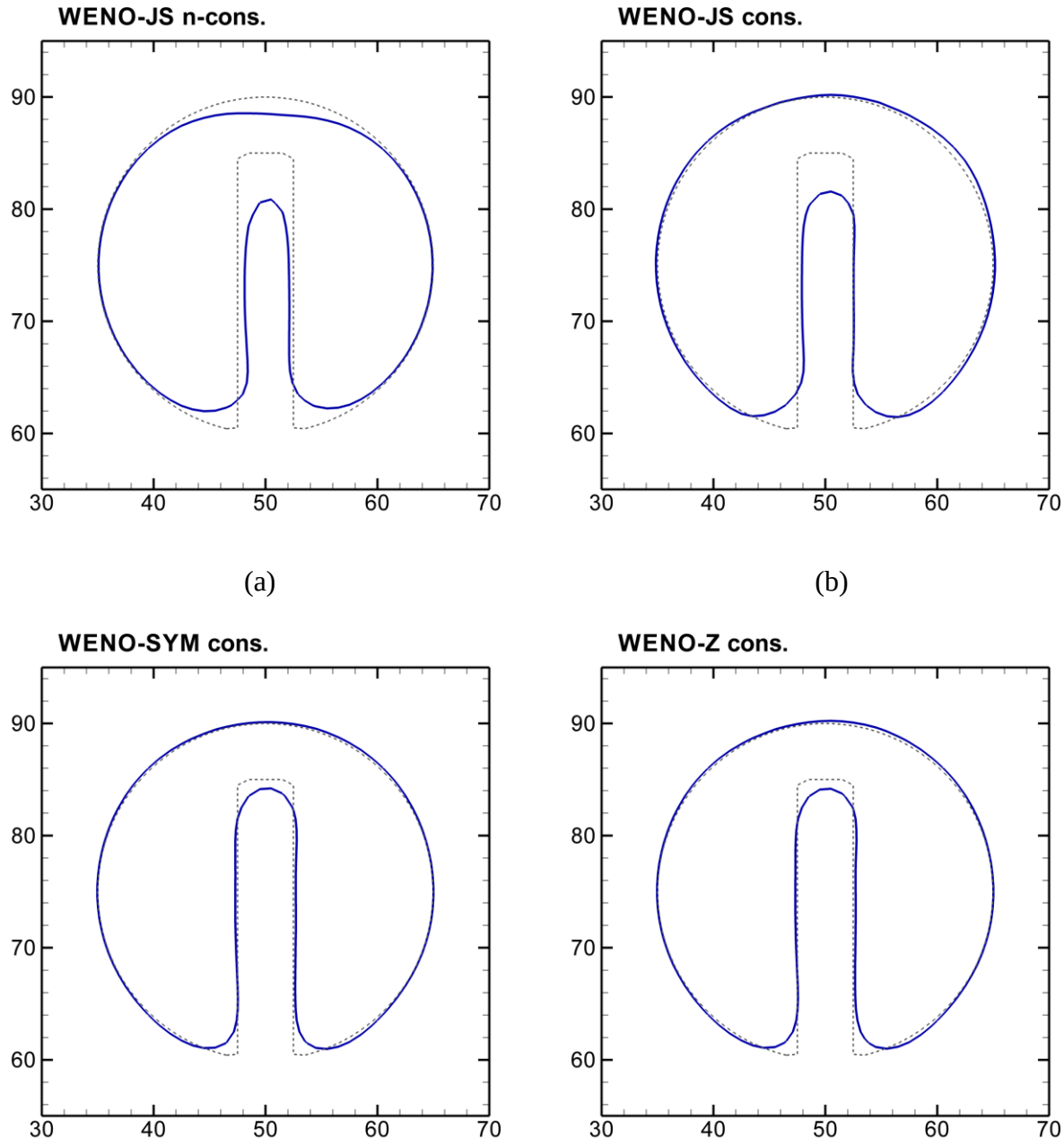


Figure 4.31: Zalesak's test. Interface after 10 full rotations. Comparison of different schemes for convective term in the level set equation: (a) WENO-JS n-cons., (b) WENO-JS cons, (c) WENO-Z cons. and (d) WENO-Z cons. Grid resolution 100×100 points.

rate, similar as it does when in computation of compressible flow the shocks appear. In the vicinity of shocks, the shock-capturing schemes have at most the first order accuracy. The papers by Engquist & Sjogreen (1999) and Ostapenko (2010) are devoted to this issue.

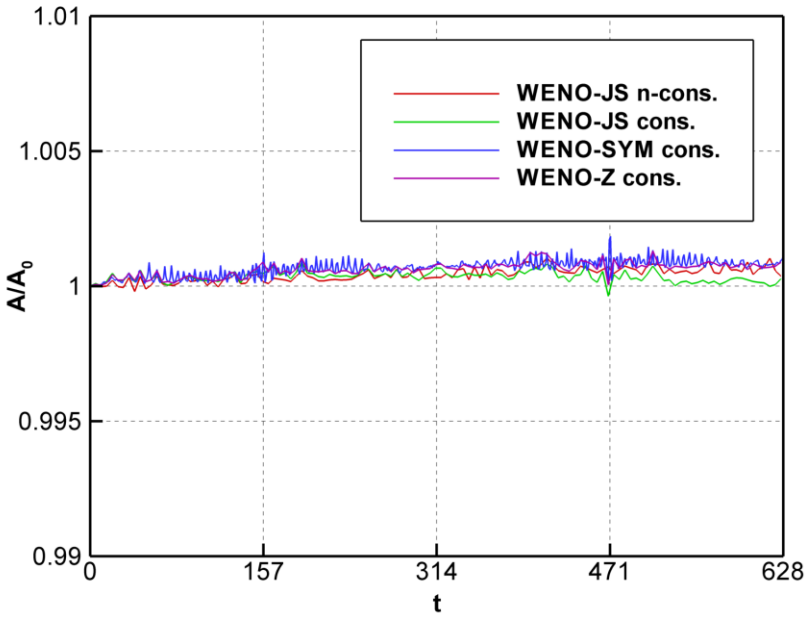


Figure 4.32:. The temporal evolution of the normalized area in the Zalesak's test. Grid resolution: 100×100 points.

Table 4.7: L_2 norm of error for φ and convergence rates for Zalesak's disk after 10 full rotations.

	50×50	100×100		200×200	
Scheme	$\|\| e_\varphi \|\|_{L_2}$	$\|\| e_\varphi \|\|_{L_2}$	Rate	$\|\| e_\varphi \|\|_{L_2}$	Rate
WENO-JS n-cons.	1.22	4.89×10^{-1}	1.31	2.87×10^{-1}	0.77
WENO-JS cons.	7.69×10^{-1}	3.52×10^{-1}	1.12	2.40×10^{-1}	0.55
WENO-SYM cons.	5.74×10^{-1}	3.06×10^{-1}	0.91	2.36×10^{-1}	0.37
WENO-Z cons.	6.09×10^{-1}	3.08×10^{-1}	0.98	2.36×10^{-1}	0.38

4.7 Conclusions

We examined the ability of the new method with different formulations of the source term. For all assessed test cases, we observed that use of derived approximations to the source term in the level set equation can significantly reduce the number of iterations. This implies that modifications proposed for the level set equation provide the interface being more precisely predicted, and the computational cost is lower when compared to standard approach with the re-initialization procedure. When the external velocity field depends on the derivatives of the level set function, this advantage of the new approach is fairly noticeable. "Oscillating circle test" and test examples with premixed combustion showed clearly this. However, when the external velocity field is independent of the level set field, and when the interface is very strongly deformed, both approaches require a relatively large number of iterations to converge re-initializations. Although the computation of two-dimensional flow, produced by the homogeneous strain, showed that area enclosed by the zero level function is preserved better by new approach, the accuracy in prediction of the analytical solution of the zero level function is of the same order in the new approach and in the standard one with re-initializations.

In addition to comparison of different forms of the level set equation, we compared also different high-resolution schemes applied to the level set transport. We showed an explicit advantage of the WENO-SYM and WENO-Z schemes versus the standard 5^{th}-order WENO-JS scheme.

Chapter 5

Examples with two-phase flows

In this chapter we present three examples of flows with phase boundary. As in Chapter 4, the objective is to compare standard and new forms of the level set equation, but in difference with Chapter 4, the velocity-field is not presumed here but calculated by the Navier-Stokes equations. For numerical solution of the Navier-Stokes equations we use the algorithm presented in Chapter 2: WENO-Z scheme is used for convective terms, 4^{th}-order central difference scheme for viscous terms and 3^{rd}-order TVD Runge-Kutta method for time integration. For discretization of the level set equation, we use WENO-Z scheme in conservative form. Re-initialization equation is discretized with 5^{th}-order Hamilton-Jacobi WENO scheme. Convergence criterion, given by Equation (4.5), is used for control the number of iterations. When the modified level set equation is used, we applied the zero-order local approximation (3.19) with the re-initialization procedure.

5.1 Capillary wave

This test, referred here to as Prosperetti 1981 test, represents the viscous decay of a two-dimensional standing wave on the interface between two fluids of different densities, ρ_1 and ρ_2. For simplicity, two fluids are set to be of the same kinematic viscosity, and the surface tension coefficient σ is assumed to be constant. Prosperetti (1981) developed the linear theory of such interaction between surface tension forces and viscous effects. His solution for evolution of wave amplitude with time, scaled by inviscid oscillation frequency $\sqrt{\frac{\sigma}{\rho_1+\rho_2}}$, represents the reference solution. Earlier, this reference solution was used in validation of the refined resolution in the interface vicinity, namely by spectrally refined interface method of Desjardins & Pitsch (2008), and by balanced force refined level set grid method of Herrmann (2008). In our work, we have undertaken similar demarche: using the same parameters and the same initial form of the

interface, as in Desjardins & Pitsch (2008) and Herrmann (2008), we compared new and standard approaches with the solution of Prosperetti (1981). The simulations are performed in a $[0,2\pi]\times[0,2\pi]$ domain, with $\rho_1=1$, $\lambda=\rho_2/\rho_1=1000$, $\sigma=2$, $\mu_1/\rho_1=0.0064720863$, and using different mesh resolution: 16×16, 32×32, 64×64. The initial form of the interface has been taken in the following form:

$$\varphi_0(x,y)=\pi-y+B_0\cos(2\pi x/B_1) \tag{5.1}$$

where B_1 is set to 2π and B_0 is set to $0.01B_1$. In the x-direction, periodic boundary conditions are employed, while the y-direction assumes top and bottom symmetry. In Fig.5.1, the variation of non-dimensional amplitude (amplitude divided by 2π) with non-dimensional time is shown in comparison with the Prosperetti solution using different grids, and new (on the left) and standard approaches. No visible difference is seen between two approaches; more refined mesh gives better solution, and 32×32, 64×64 grids predict very good the reference solution. These results are very similar to those obtained by Desjardins & Pitsch (2008) and Herrmann (2008). However, if the iterations number in the re-initialization procedure is fixed, and if to compare standard approach with the new one when the zero-local approximation is used, then mean deviation from $|\nabla\varphi|=1$ becomes a pronounced value. This is seen in Fig.5.2. However in general in this "smooth" test case, the difference between results from the both approaches is negligible.

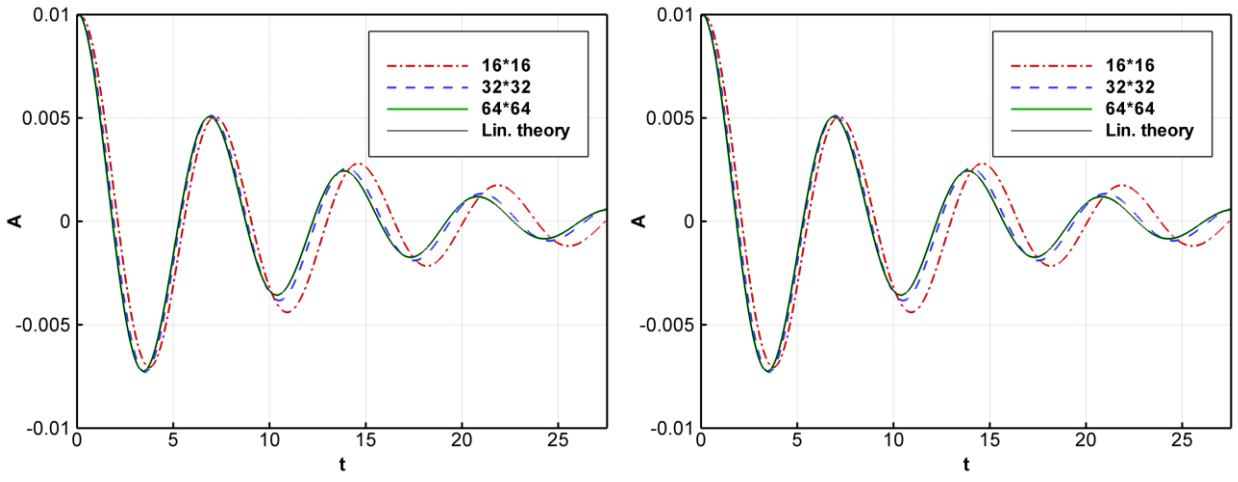


Figure 5.1: Time evolution of the normalized wave amplitude for different grid resolutions; on the left – standard approach; on the right – new approach.

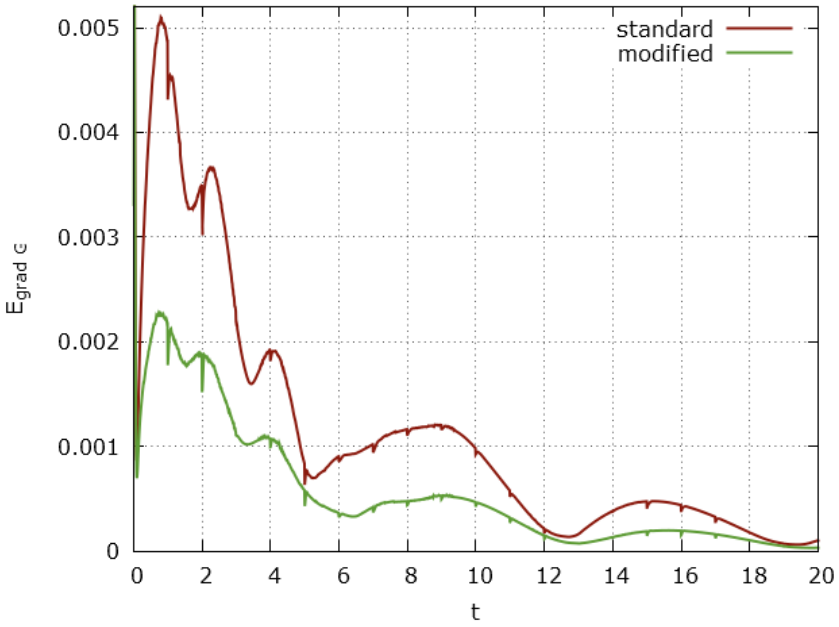


Figure 5.2: Mean deviation from $|\nabla\varphi|=1$ for the case of prescribed number of iterations, $N_{iter}=3$. Grid resolution 32×32 points.

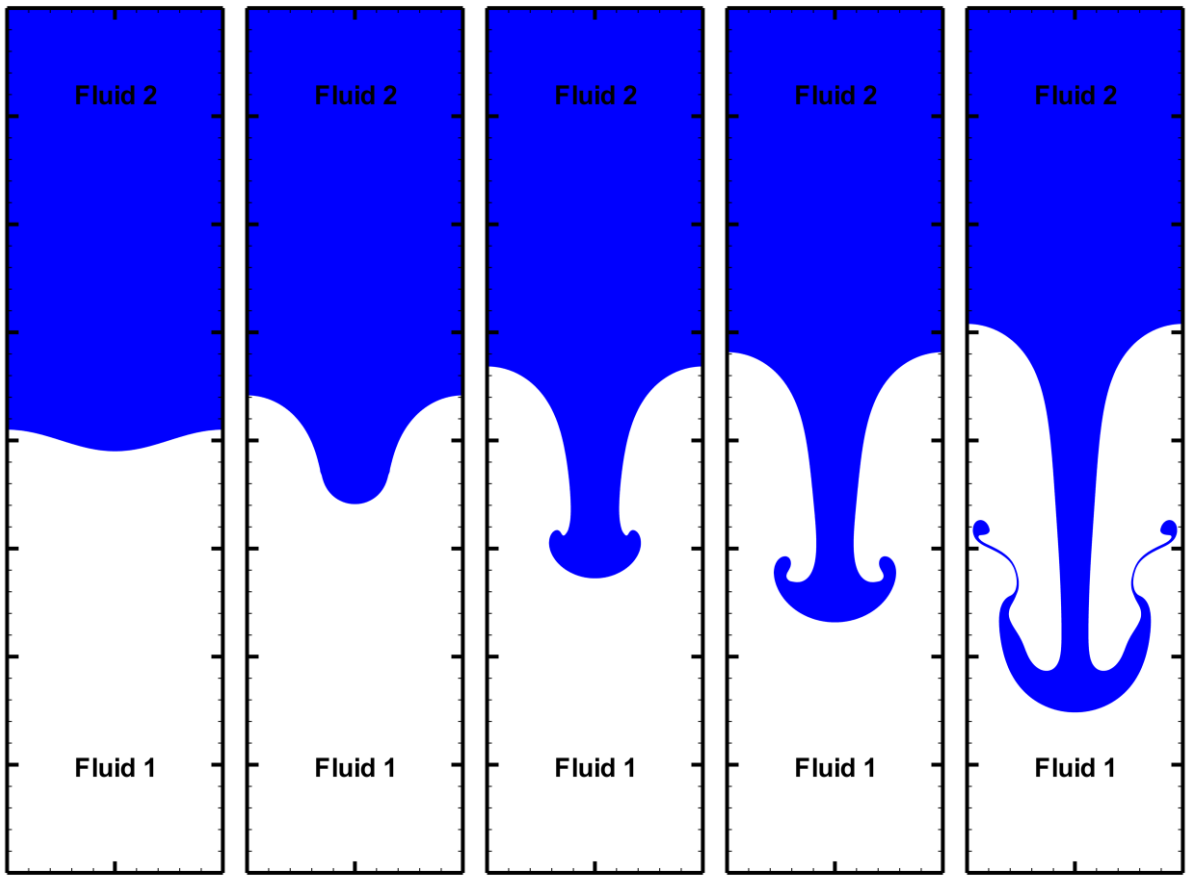
5.2 Rayleigh-Taylor instability

Numerous studies have used this test to characterize the quality of the interface transport method. In a $[0,1]\times[0,4]$ domain, two fluids are initially separated by an interface defined by the zero iso-contour

$$\varphi_0(x,y)=y-y_0+\varepsilon\cos(2\pi x), \tag{5.2}$$

where the disturbance amplitude ε is taken to be 0.05. The top fluid has density ρ_2 = 1.225; the density of the bottom fluid is set to $\rho_1=0.1694$. The both fluids have the same dynamic viscosity, $\mu_1=\mu_2=3.13\times10^{-3}$. The surface tension effect is neglected here. Initial velocity field and the pressure are set to zero. The periodical boundary conditions for velocity are used in x-direction and the no-slip condition is taken in y-direction. For the pressure we used the Neumann condition in both directions. The reference solution was obtained by standard approach with the re-initialization procedure on a fine grid with resolution of 256×1024 points. At different times, it is depicted in Figure 5.3 in form of development of a mushroom shape with thin structures. In this test case, until time $t=0.7$, all numerical approaches (VOF by Zaleski, 2000; Level Set by Tanguy, 2005; Interface tracking by Popinet, 1999) give very similar results. However at later

times, the obtained in the literature solutions differ depending on the choice of the numerical algorithm. In Figure 5.4, the results of calculation from standard (left-hand side of each couple) and new approaches (right-hand side of each couple), the both with re-initializations, are compared for 64×256 and 128×512 grids. This figure shows snapshots of the density field and isolines of the level set function at time $t=0.9$.



(a) $t=0.0$ (b) $t=0.4$ (c) $t=0.7$ (d) $t=0.8$ (e) $t=0.9$

Figure 5.3: The reference density-field at different times in the Rayleigh-Taylor instability; solution is obtained on 256×1024 grid, by standard approach with re-initializations of level sets

Observing the interface topology in comparison with the reference solution, it is seen that on the grid containing 64×256 points, the both approaches flawed to preserve the stretched filament from its "numerical" disintegration. However on the grid containing 128×512 points, one may favor the results from new approach, which slightly better preserves filaments from their disintegration. The change of the total mass with time, shown in Figure 5.5, favor more

significantly the use of the modified level set equation, even in its simplest form, i.e. with the the zero-order local approximation (3.19) . This advantage is also explicit when the number of iterations in re-initialization procedure is compared. Figure 5.6 shows significantly less of those iterations required when new approach is used. This is also seen in Figure 5.7 where the mean deviation from $|\nabla\varphi|=1$ for the case of prescribed number of iterations, $N_{iter}=3$, is given.

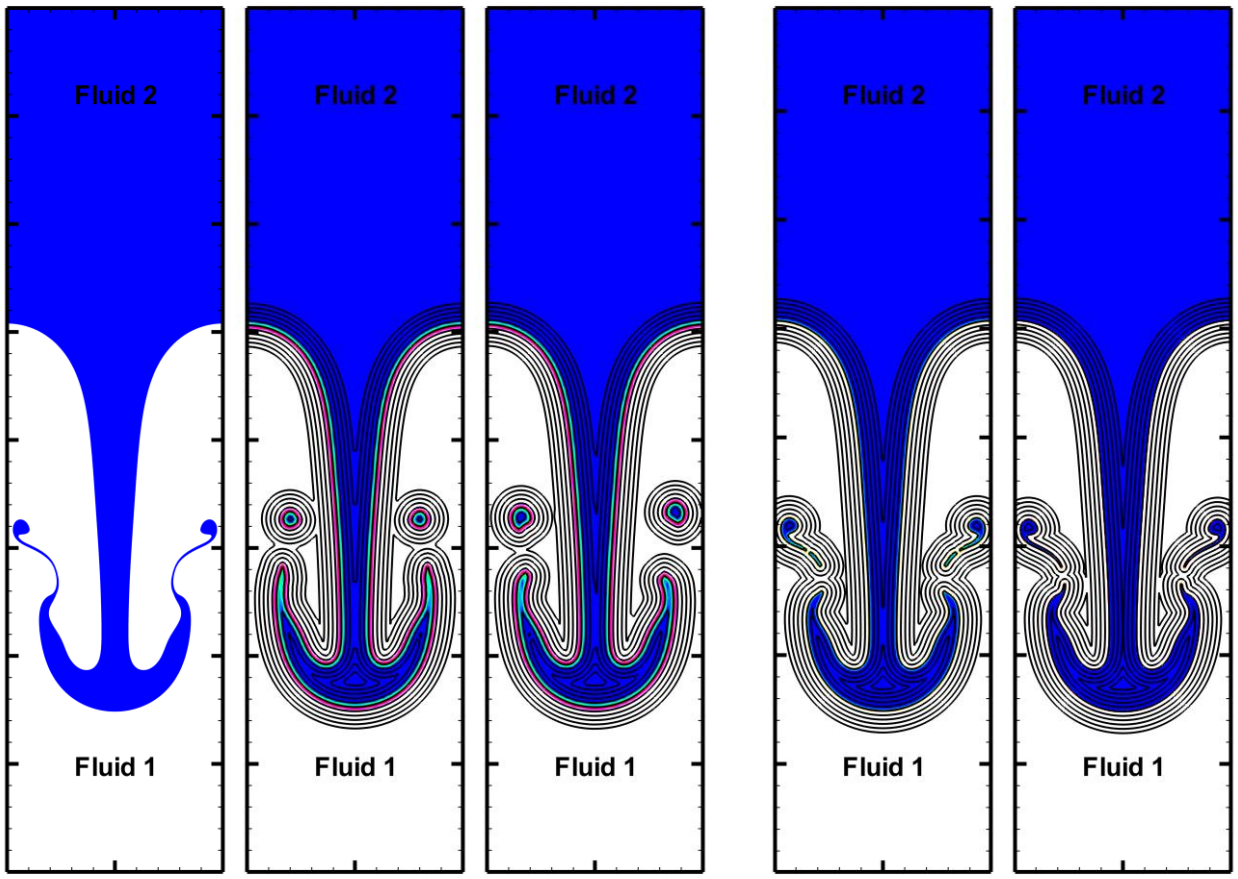


Figure 5.4: Rayleigh-Taylor instability. Density field and 11 isolines of the level set function $\varphi=\{-0.1,...,0.1\}$ for the standard (left-hand side of each couple of snapshots) and new (right-hand side of each couple of snapshots) approaches on two grids. On the left: reference solution.

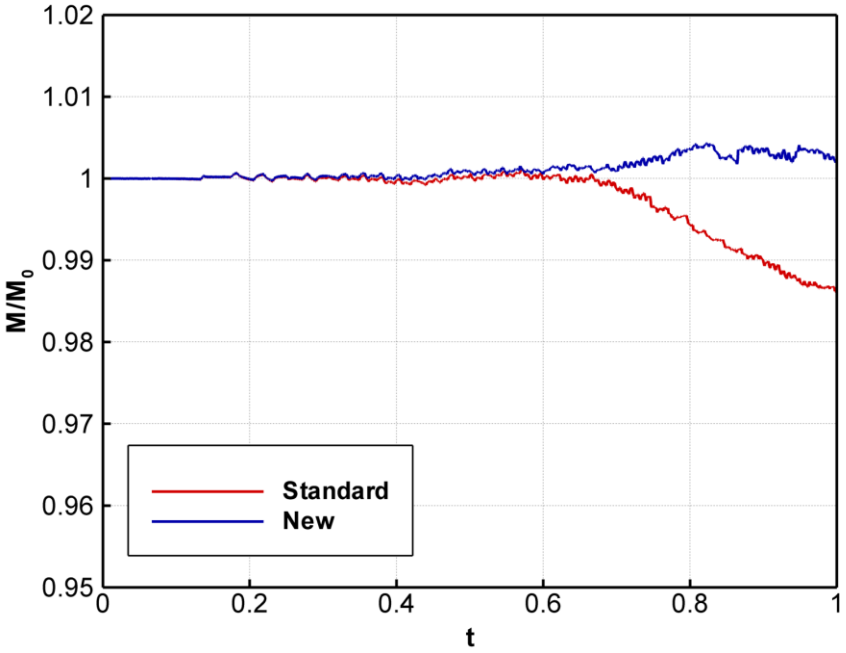


Figure 5.5: Change of normalized mass of liquid (fluid "2") with time in Rayleigh-Taylor instability. Grid 64×256 points.

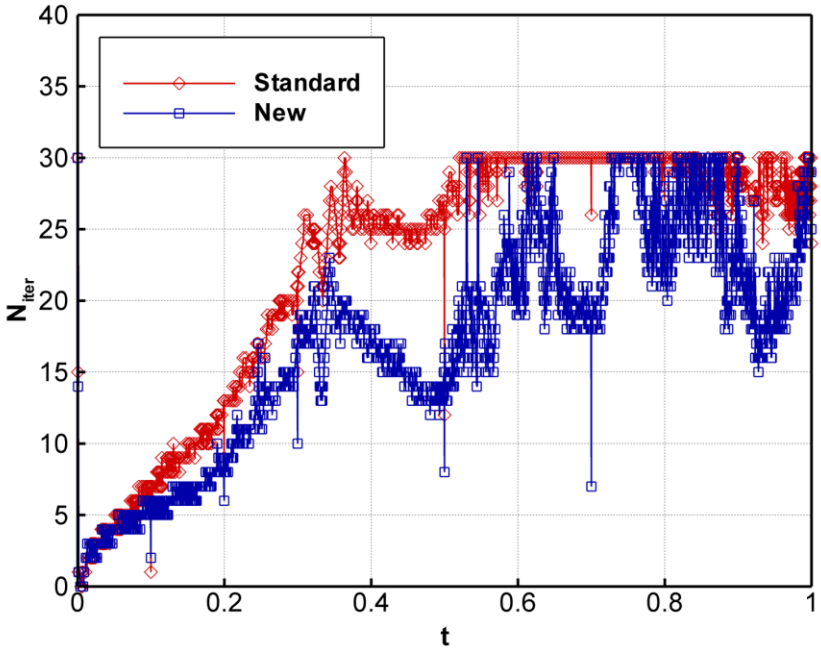


Figure 5.6: Number of iterations in re-initialization procedure in Rayleigh-Taylor instability. Grid 64×256 points.

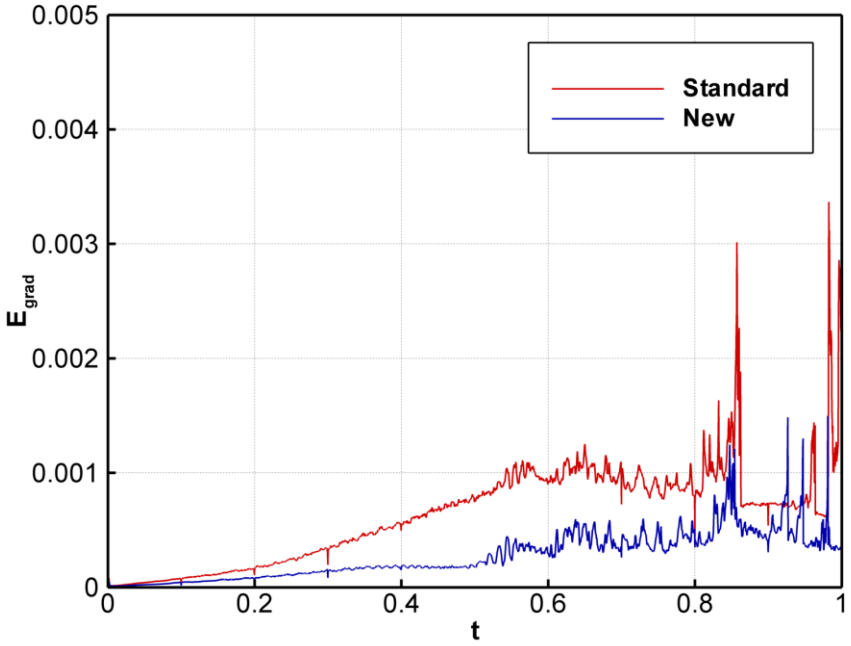


Figure 5.7: Rayleigh-Taylor instability. Mean deviation from $|\nabla\varphi|=1$ for the case of prescribed number of iterations, $N_{iter}=3$. Grid resolution 64×256 points.

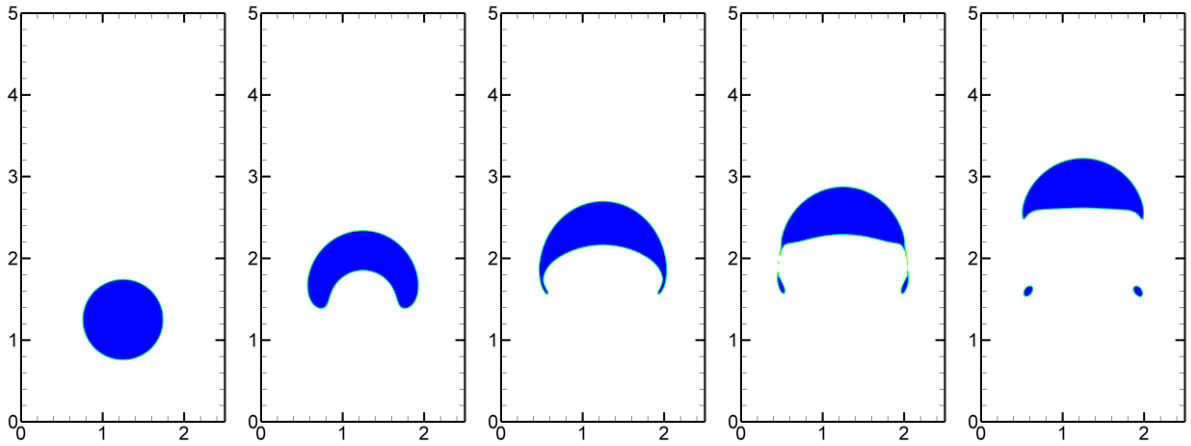
5.3 Rising bubble

In this case, we consider the temporal evolution of a two-dimensional bubble in a fully filled container. It is another challenging test for interface tracking methods because of complex interface changes. This test problem was also considered by many authors; in our study we used the parameters proposed in Gaudlitz & Adams (2008). Initially, the bubble is a circle with radius $R=0.5$ centered at $(x,y)=(1.25,1.25)$ in computational domain $\Omega:[0,2.5]\times[0,5]$. The corresponding initial data for the level set function is

$$\varphi_0(x,y)=\sqrt{(x-1.25)^2+(y-1.25)^2}-R. \tag{5.3}$$

The physical parameters are the following: $Re=58$, $We=104$, $Fr=1$, the density ratio $\lambda=0.025$ and the viscosity ratio $\eta=0.012$. The initial velocity field is zero, no-slip boundary conditions for velocity used in the both directions. Computation is done up to $t=5.0$ when bubble attains a distance of about half the domain height. During its ascent the bubble undergoes strong deformation, and two smaller bubbles are pinched off in outer regions of the stretched bubble. The reference solution was obtained by standard approach with 160×320 grid points. This solution is presented in Figure 5.8. Here, time sequences of the deformed bubble shape are

shown. Starting from the initial spherical shape, the bubble is stretched to its oblate-ellipsoidal cap shape, as shown in the figure. During this ascension, a very thin bubble skirt is formed behind the bubble, and finally, two small bubbles are pinched off from the "mother" bulk.



(a) $t = 0.0$ (b) $t = 2.0$ (c) $t = 3.0$ (d) $t = 3.5$ (e) $t = 4.5$

Figure 5.8: Computed shapes of a gas bubble rising in a quiescent liquid in the rising bubble test. Grid 160×320 points.

Comparing standard and new approaches, both with the re-initialization procedure and new one with the zero-order local approximation (3.19), Figure 5.9 shows the results on coarse grid with 40×80 points and on fine grid with 160×320 points. Here a smoothed density field and isolines of the level set function are presented. Compared to the reference solution on coarse grid with 40×80 points, it is seen that new approach predicts pinch-off of "daughter" bubbles, which is not the case when standard approach is applied. On fine grid with 160×320, the both methods give very similar distributions with secondary bubbles. Figure 5.10 shows the time evolution of the normalized bubble mass. Using standard approach, our results reproduced the results presented in Gaudlitz & Adams (2008). However using new approach, it is seen that the normalized bubble mass is conserved fairly better than in the case of standard approach. This advantage is confirmed by comparison of the mean deviation of φ from signed distance function near the interface. This is given in Figure 5.9.

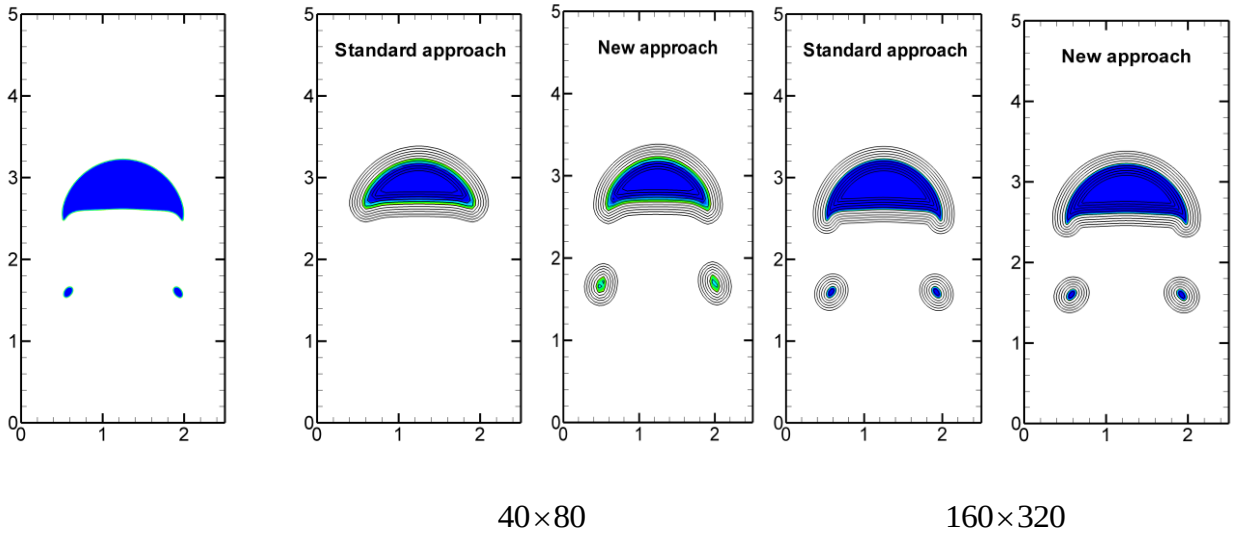


Figure 5.9: Density field and 11 isolines of the level set function $\varphi = \{-0.15, \ldots, 0.15\}$ at time $t = 4.5$ in the rising bubble test for the standard and new approach using coarse and fine grids. On the left: reference solution from Fig. 5.8 e)

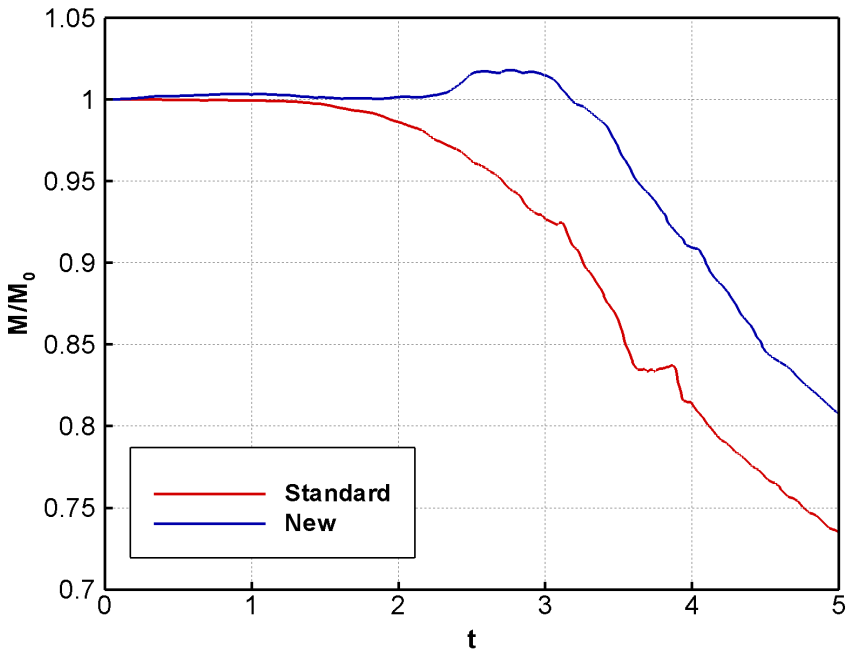


Figure 5.8: Time evolution of normalized mass of bubble in the rising bubble test. Grid: 40×80

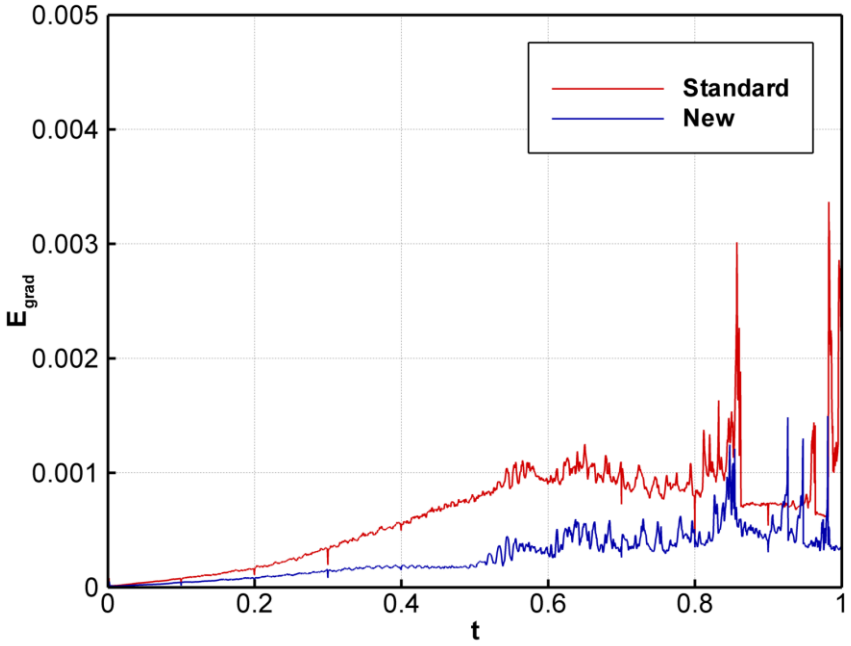


Figure 5.9: Mean deviation from $|\nabla\varphi|=1$ for the case of prescribed number of iterations, $N_{iter}=3$ in the rising bubble test. Grid: 40×80.

5.4 Conclusion

Validations, considered in this Chapter, concern Capillary Wave, Rayleigh-Taylor instability and Rising Bubble test cases. These assessments performed are preliminary and more refined estimation and more complex configurations are needed to be investigated. This is envisaged for our future work. On the stage of the presented analysis, it is clear that both approaches, standard and new with the zero-order local approximation, the both with re-initializations of level sets, may predict very similar configuration of interface. Filaments slightly better preserved from numerical disintegration in the case of Rayleigh-Taylor instability, secondary bubbles predicted more closely to reference solution on coarse grid in the case of Rising Bubble, all these results, obtained by the new approach, does not favor explicitly this approach compared to the standard one. However we observed that the new approach provides the result with better preservation of the total mass, and by the number of iterations fairly less than the standard one does. Consequently, new approach may significantly reduce the computational cost.

General conclusions

In a variety of physical processes the discontinuity in physical properties is mimicked by evolution of a fluid-interface. Examples include immiscible gas-liquid flows, premixed flames, solidification and melting phenomena, etc. For computing of interface evolution, the level set methods are often used. Our work is done in same lines: we were interested in simulation of two-phase flows with free interface, which represents a new subject in LMFA, with target in the future on atomization process at a high-Reynolds number. After many flawed attempts to get an existing already numerical code, we started from the very beginning: we constructed our own code. To this end in **Chapter 2**, the numerical method for solving two-dimensional incompressible Navier-Stokes equations is presented. Our chosen numerical algorithm is similar, in part, to that in Ph.D. theses of Tanguy (2004) and Couderc (2007). It is related to the scheme of Zhang & Jackson (2009), which is a high-order incompressible flow algorithm based on the projection method and Weighted Essentially Non-Oscillatory (WENO) finite differences. Our contribution is as follows: in order to increase the accuracy, we introduced two modifications in the method of Zhang & Jackson (2009). First, we implemented low dissipative WENO-Z scheme of Borges *et al.* (2008) and bandwidth optimized WENO-SYM scheme of Martin *et al.* (2006), instead of classical WENO scheme of Shu & Jiang (1996). Second, we used the high-order interpolation scheme in order to determine velocity on the staggered grid, instead of standard linear interpolation. Our algorithm, which is close to 5^{th}-order accurate for velocity variables and 3^{rd}-order in time, was validated on different test problems with the reference solution, either exact or numerical. The test cases included the Taylor-Green vortex, the two-dimensional shear layer, the driven cavity flow and the two-dimensional decaying turbulence. We demonstrated that results of WENO-Z scheme for convective terms are fairly better than those obtained for the standard WENO scheme. From other side, the computational cost of the modified scheme remains the same as for standard WENO scheme. As to WENO-SYM scheme, it shows also improvement in numerical results, but not so significantly as it does by WENO-Z. Thus the WENO-Z scheme is applied for convective terms in straight-forward way without artificial compressibility. Our next step concerned the level set approach itself.

The well-known problem addressed to the level set equation is this: if the flow velocity is not constant, the level set function may become strongly distorted. Thus, the numerical integration of the level set equation may suffer from loss of accuracy in prediction of interface. In level set methods, this problem is remedied by the reinitialization procedure, *i.e.* by reconstruction of the level set function in a way to satisfy the eikonal equation. However from

numerical experience, it has been observed that after several iterations by the re-initialization procedure, the zero level set may move towards nearest grid points which will not lie directly on the interface. This may incur errors into solution to the level set equation. This motivated us to present in **Chapter 3** the new form of the level set equation by embedding a source term. The exact expression of this term is such that the eikonal equation is automatically satisfied. Furthermore on the interface, this term is equal to zero. Thereby integrating this new form of the level set equation, there is no more necessity in reinitialization of level sets, similar to the extension velocity method. In the meantime, the advantage of new approach proposed is this: the exact expression of the source term allows for the possibility of derivation of its local approximate forms, of zero-, first- and higher-order accuracy. Then compared to the extension velocity approach, the new approach will open simplifications in realization of level set methods. Compared to the standard approach with the reinitialization procedure, this approach gives the economies in the number of level set re-initializations, and also, due to reduced number of re-initializations, it allows the improvement in resolution of the interface. Let us remind here the modified level set equation and its local approximations. The exact form of the modified level set equation is

$$\frac{\partial \varphi}{\partial t} + u_k \frac{\partial \varphi}{\partial x_k} = \left[u_k - \left(u_k \right)\big|_{\varphi=0} \right] \frac{\partial \varphi}{\partial x_k}$$

The zero-order and the first-order local approximations have the following form, respectively:

$$\frac{\partial \varphi}{\partial t} + u_k \frac{\partial \varphi}{\partial x_k} = \frac{\partial \varphi}{\partial x_j} \left(\frac{\partial u_k}{\partial x_j} \right) \frac{\partial \varphi}{\partial x_k} \varphi$$

$$\frac{\partial \varphi}{\partial t} + u_k \frac{\partial \varphi}{\partial x_k} = \left(\frac{\partial u_k}{\partial x_j} - \frac{1}{2} \varphi \frac{\partial \varphi}{\partial x_l} \frac{\partial^2 u_k}{\partial x_l \partial x_j} \right) \frac{\partial \varphi}{\partial x_j} \frac{\partial \varphi}{\partial x_k} \varphi$$

Chapter 4 is devoted to numerical assessment of modified level set equation. This equation was discretized in time using the 3-stage third-order TVD Runge-Kutta scheme (Shu & Osher, 1988). For spatial discretization of the convective term, the fifth-order WENO scheme (Jiang & Shu, 1996) was used. Concerning discretization of the source term, as local approximation, we used the fifth-order WENO scheme for Hamilton–Jacobi equations, which was proposed in Jiang & Peng (2000). In this scheme, the Hamiltonian $H = \alpha_{ij} \partial_i \varphi \partial_j \varphi, \;\; \alpha_{ij} = \varphi \partial_i u_j$ is approximated by global Lax-Friedrichs flux function. Our experience showed that such an approximation to the source term eliminates the high-frequency numerical oscillations, which may be developed if the central-difference scheme is used. As to the exact form of the source term in the level set equation, the main difficulty was to embed $\vec{u}\big|_{\varphi=0}$ into computation of $\varphi(\vec{x},t)$, *i.e.* to determine

the interface propagation velocity in the direction of the characteristics indexed by the position of interest $\vec{x}$. Similar to Gomes & Faugeras (2000), our calculation of $\vec{u}\big|_{\varphi=0}$ was also based on computation of $\vec{x}\big|_{\varphi=0} = \vec{x} - \varphi\nabla\varphi$ first, and then of $\vec{u}\big|_{\varphi=0}$, corresponding to $\vec{x}\big|_{\varphi=0}$. However, the direct computation of $\vec{x}\big|_{\varphi=0} = \vec{x} - \varphi\nabla\varphi$ reduced the robustness of the algorithm: the errors in $\nabla\varphi$ introduced errors in $\vec{x}\big|_{\varphi=0}$. This motivated us to use the iterative gradient descent procedure, described in Ph.D of Herrmann (2001), for example.

The following test cases were chosen: (i) the one and two-dimensional flows produced by homogeneous strain, since these flows have analytical solutions; (ii) the interface stretching by a single vortex, with the reference solution preliminary obtained on the grid 1024×1024; (iii) the oscillating circle test proposed by Hartmann *et al*. (2008); (iv) the premixed flame propagation; (v) the solid body rotation of disk with a thin slot proposed by Zalesak (1979).

We obtained numerical results from the modified level set equation with exact form of the source term and without re-initializations; or with local approximation of the source term and with re-initializations. These results were compared with the reference solutions (either analytical or numerical) and with standard approach completed by the re-initialization procedure. For all assessed test cases, we observed that the use of derived approximations to the source term in the level set equation can significantly reduce the number of iterations. We showed that modifications proposed for the level set equation provide the interface being more precisely predicted, and the computational cost being lower in the comparison to the standard approach. When the external velocity field depends on the derivatives of the level set function, this advantage of the new approach is fairly noticeable. "Oscillating circle test", flow with one-dimensional strain, and the test example with premixed combustion exhibited clearly this. However, when the external velocity field is independent of the level set field, and if the interface is excessively deformed, both approaches, require a relatively large number of iterations to converge re-initializations. Although the computation of two-dimensional flow, produced by the homogeneous strain, showed that the area enclosed by the zero level function is preserved better by the new approach, the accuracy in prediction of the analytical solution is of the same order in the new approach and in the standard one with re-initializations. In this Chapter we also examined the ability of different high-resolution schemes applied to the level set transport. We showed an explicit advantage of the WENO-SYM and WENO-Z schemes versus the standard 5^{th}-order WENO-JS scheme.

In **Chapter 5**, we considered three examples of flows with interface: Capillary Wave, Rayleigh-Taylor instability and Rising Bubble. As in Chapter 4, the objective was to compare standard and new forms of the level set equation, but in difference with Chapter 4, the velocity-field was not presumed but calculated by the Navier-Stokes equations. We observed that for the both approaches, standard one and new one with the zero-order local approximation (the both

with re-initializations of level sets), the predictions of interface configuration are very similar. Filaments slightly better preserved from numerical disintegration in the case of Rayleigh-Taylor instability, the secondary bubbles predicted more closely to the reference solution on coarse grid in the case of Rising Bubble, all these positive results, obtained by the new approach, does not favor explicitly this approach compared to the standard one. However we observed that the new approach provides the result with better preservation of the total mass, and by the number of iterations fairly less than the standard one does. Consequently, new approach may significantly reduce the computational cost. It is worthwhile to note that these assessments performed are preliminary and more refined estimation, with more complex flow configurations, are needed to be investigated. This is envisaged for our future work.

The manuscript is ended up with **Appendix A, B, C. Appendix A** contains demonstration of how the re-initialization procedure works. A simple flow produced by one-dimensional strain is selected to illustrate clearly the key point of this procedure. Namely, it consists in the use of two *G*-fields at successive time steps: (i) the first field, with $\left|\nabla\tilde{\tilde{G}}\right|>1$, is used to find the position of zero level set at current time; (ii) the second field, with $\left|\nabla G_{rein}\right|=1$, is constructed from the knowledge of this position. **Appendix B** gives derivation of local approximations from the exact equation, and **Appendix C** concerns WENO interpolation

Appendix A

Simple illustration of the re-initialization procedure

Consider the case of non-rotational time-independent strain $u = -kx,\ \ k \equiv const > 0$. The one-dimensional level set equation is

$$\frac{\partial G}{\partial t} - kx\frac{\partial G}{\partial x} = 0,\ \ G(x, t=0) = -x + x_0. \tag{A1}$$

Its exact solution may readily be verified:

$$G(x,t) = -\exp(kt)x + x_0. \tag{A2}$$

It is seen that the norm of gradient of this solution grows exponentially:

$$\left|\nabla G(x,t)\right| = \left|\frac{\partial G}{\partial x}\right| = \exp(kt) \underset{t\to+\infty}{\to} +\infty \tag{A3}$$

and the position $x_f(t)$ of the zero level , $G(x_f(t), t) = 0$, changes with time according to:

$$x_f(t) = x_0 \exp(-kt). \tag{A4}$$

The exponential growth of level set gradients involves progressively the approximation error in $G(x,t)$; it degrades the numerical accuracy for the zero level set.

Our simple illustration of the re-initialization procedure deals with the explicit Euler scheme. In integration of (A1), the advancement in time after the first time step gives

$$G(\Delta t) = G\big|_{t=0} + kx\left.\frac{\partial G}{\partial x}\right|_{t=0}\Delta t = -x(1 + k\Delta t) + x_0, \tag{A5}$$

$$\left|\frac{\partial G(\Delta t)}{\partial x}\right| = (1 + k\Delta t) > 1. \tag{A6}$$

Thereby according to (A5), the position of the zero level $G(x_f(\Delta t)) = 0$ moves from x_0 to

$$x_f(\Delta t) = \frac{x_0}{(1+k\Delta t)}. \tag{A7}$$

After the second step, we have

$$G(2\Delta t) = G(\Delta t) + kx\frac{\partial G}{\partial x}\bigg|_{t=\Delta t}\Delta t = -x(1+k\Delta t) + x_0 - kx(1+k\Delta t)\Delta t = -x(1+k\Delta t)^2 + x_0 \tag{A8}$$

and the new position of zero level

$$x_f(2\Delta t) = \frac{x_f(\Delta t)}{(1+k\Delta t)} = \frac{x_0}{(1+k\Delta t)^2}. \tag{A9}$$

After $N = \frac{t}{\Delta t}$ steps, the recursion is

$$G(t = N\Delta t) = -x(1+kN\Delta t)^N + x_0 = -x(1+k\frac{t}{N})^N + x_0 \tag{A10}$$

and discrete positions of the zero level set, $G(x_f(t)) = 0$ are determined by:

$$x_f(t = N\Delta t) = \frac{x_0}{(1+kN\Delta t)^N} = x_0(1+k\frac{t}{N})^{-N}. \tag{A11}$$

Obviously, if $N \to +\infty$ (A10), (A11) reduce to (A2), (A4), respectively.

Now after each single-step, let us re-initialize the level set. After the first step, $G(\Delta t)$ field, given by solution (A5), should be replaced by the new one, say $G_{rein}(\Delta t)$ field, in which the zero level is the same as in original field, *i.e.* it is determined by $x_f(\Delta t)$ from (A7), and the norm of gradient of this new field should be equal to unity. To this end, $G_{rein}(\Delta t)$ is simply represented by translation of initial field $G(x, t=0) = -x + x_0$ to

$$G_{rein}(\Delta t) = -x + x_f(\Delta t), \tag{A12}$$

$$\left|\frac{\partial G_{rein}(\Delta t)}{\partial x}\right| = 1 \tag{A13}$$

Then the second step will start from (A12), and not from (A5). Instead of (A8), we have the level set equation:

$$\tilde{G}(2\Delta t) = G_{rein}(\Delta t) + kx\frac{\partial G_{rein}}{\partial x}\bigg|_{t=\Delta t}\Delta t = -x(1+k\Delta t) + x_f(\Delta t) \tag{A14}$$

$$\left|\frac{\partial\tilde{G}(2\Delta t)}{\partial x}\right| = (1+k\Delta t) > 1. \tag{A15}$$

Note that the zero level in (B14), $-x_f(2\Delta t)(1+k\Delta t)+x_f(\Delta t)=0$, has the same new position $x_f(2\Delta t)$, as the zero level in original field $G(2\Delta t)$, and is determined by (A9). The re-initialization of $\tilde{G}(2\Delta t)$ is similar to re-initialization of $G(\Delta t)$: it corresponds to translation of $G_{rein}(\Delta t)$ from (A12) to

$$G_{rein}(2\Delta t) = -x + x_f(2\Delta t), \quad \left|\frac{\partial G_{rein}(2\Delta t)}{\partial x}\right| = 1. \tag{A16}$$

The third step starts from (A16):

$$\tilde{G}(3\Delta t) = G_{rein}(2\Delta t) + kx\left.\frac{\partial G_{rein}}{\partial x}\right|_{t=2\Delta t}\Delta t = -x(1+k\Delta t) + x_f(2\Delta t), \tag{A17}$$

$$\left|\frac{\partial G_*(3\Delta t)}{\partial x}\right| = (1+k\Delta t) > 1. \tag{A18}$$

Here again the zero level position at $t=3\Delta t$ coincides with the zero level position $x_f(3\Delta t)$ in original field $G\left(x_f(3\Delta t)\right)=0$, and re-initialization of $\tilde{G}(3\Delta t)$ consists in translation of $G_{rein}(2\Delta t)$ to

$$G_{rein}(3\Delta t) = -x + x_f(3\Delta t), \quad \left|\frac{\partial G_{rein}(3\Delta t)}{\partial x}\right| = 1 \tag{A19}$$

Performing the reinitialization at each consecutive time step, we obtain after N time steps

$$G_{rein}((N-1)\Delta t) = -x + x_f((N-1)\Delta t), \quad \left|\frac{\partial G_{rein}((N-1)\Delta t)}{\partial x}\right| = 1, \tag{A20}$$

which represents the intermediate level set function in interval $\left[(N-1)\Delta t, N\Delta t\right]$. Then at $t = N\Delta t$, the level sets are updated by the level set equation to

$$\tilde{G}(N\Delta t) = -x(1+k\Delta t) + x_f((N-1)\Delta t), \quad \left|\frac{\partial\tilde{G}(N\Delta t)}{\partial x}\right| = (1+k\Delta t) > 1. \tag{A21}$$

Thus at each time the re-initialization procedure is carried out by two stages: preliminary level set translation $G_{ren}(x,t) = -x + x_f(t)$, $\left|\frac{\partial G_{rein}}{\partial x}\right| = 1$, and then computation of updated level set by the level set equation.

Appendix B

Derivation of local approximations from the exact equation

In (3.9), the following expansions in a Taylor series are used:

$$A(\vec{x},t) = A_0(\vec{x}_{\varphi=0},t) + A_1(\vec{x}_{\varphi=0},t)\varphi + \ldots \tag{B1}$$

$$\frac{\partial u_k}{\partial x_i} = \left(\frac{\partial u_k}{\partial x_i}\right)\bigg|_{n=0} + \left(\frac{\partial^2 u_k}{\partial x_i \partial x_l}\right)\bigg|_{n=0} dx_l + \ldots \tag{B2}$$

where $dx_l = \dfrac{\partial \varphi}{\partial x_l}\varphi.$ Inserting these expansions into (3.9) yields

$$A_0 + A_1\varphi = -\varphi\frac{\partial \varphi}{\partial x_i}\frac{\partial}{\partial x_i}\left(A_0 + A_1\varphi\right) + \frac{\partial \varphi}{\partial x_i}\left[\frac{\partial u_k}{\partial x_i}\bigg|_{n=0} + \frac{\partial^2 u_k}{\partial x_i \partial x_l}\bigg|_{n=0}\frac{\partial \varphi}{\partial x_l}\varphi\right]\frac{\partial \varphi}{\partial x_k} \tag{B3}$$

By equating coefficients at coincident powers of φ, we write

$$A_0(\vec{x}_{\varphi=0},t) = \frac{\partial \varphi}{\partial x_i}\left(\frac{\partial u_k}{\partial x_i}\right)\bigg|_{n=0}\frac{\partial \varphi}{\partial x_k} = \left(\frac{\partial u_k n_k}{\partial n}\right)\bigg|_{n=0} \tag{B4}$$

$$A_1(\vec{x}_{\varphi=0},t) = -A_1(\vec{x}_{\varphi=0},t)\frac{\partial \varphi}{\partial x_i}\frac{\partial \varphi}{\partial x_i} + \frac{\partial \varphi}{\partial x_i}\left(\frac{\partial^2 u_i}{\partial x_k \partial x_l}\right)\bigg|_{n=0}\frac{\partial \varphi}{\partial x_k}\frac{\partial \varphi}{\partial x_l} \tag{B5}$$

From (C5), taking into account that $\dfrac{\partial \varphi}{\partial x_i}\dfrac{\partial \varphi}{\partial x_i} = \left|\nabla \varphi\right|^2 = 1,$ we obtain

$$A_1(\vec{x}_{\varphi=0},t) = \frac{1}{2}\frac{\partial \varphi}{\partial x_i}\frac{\partial \varphi}{\partial x_l}\left(\frac{\partial^2 u_i}{\partial x_k \partial x_l}\right)\bigg|_{n=0}\frac{\partial \varphi}{\partial x_k} = \frac{1}{2}\left(\frac{\partial^2 u_k\, n_k}{\partial n^2}\right)\bigg|_{n=0} \tag{B6}$$

It is seen that (B4) and (B6) are coincident with (3.20). The higher-order coefficients can be obtained by the same proceeding.

Appendix C

WENO interpolation

Consider the 1D interpolation problem (Fig. C1): given the six points x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, and x_{i+3}, corresponding data f_{i-2}, f_{i-1}, f_i, f_{i+1}, f_{i+2}, f_{i+3}, we want to estimate a value $f(x)$ for point $x \in [x_i, x_{i+1})$.

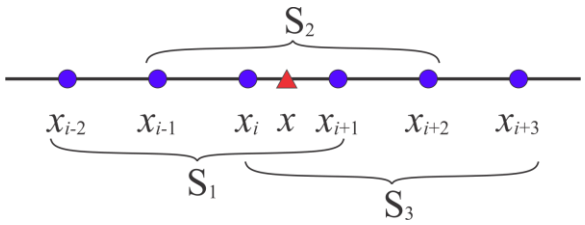


Figure C1: One-dimensional WENO interpolation: candidate stencils and interpolation point.

We begin with three candidate interpolants:

$$p_1(x) = f_{i-2} + \frac{f_{i-1} - f_{i-2}}{\Delta x}(x - x_{i-2}) + \frac{f_i - 2f_{i-1} + f_{i-2}}{2\Delta x^2}(x - x_{i-2})(x - x_{i-1}) + \\ + \frac{f_{i+1} - 3f_i + 3f_{i-1} - f_{i-2}}{6\Delta x^3}(x - x_{i-2})(x - x_{i-1})(x - x_i), \tag{C1}$$

$$p_2(x) = f_{i-1} + \frac{f_i - f_{i-1}}{\Delta x}(x - x_{i-1}) + \frac{f_{i+1} - 2f_i + f_{i-1}}{2\Delta x^2}(x - x_{i-1})(x - x_i) + \\ + \frac{f_{i+2} - 3f_{i+1} + 3f_i - f_{i+1}}{6\Delta x^3}(x - x_{i-1})(x - x_i)(x - x_{i+1}), \tag{C2}$$

$$p_3(x) = f_i + \frac{f_{i+1} - f_i}{\Delta x}(x - x_i) + \frac{f_{i+2} - 2f_{i+1} + f_i}{2\Delta x^2}(x - x_i)(x - x_{i+1}) + \\ + \frac{f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i}{6\Delta x^3}(x - x_i)(x - x_{i+1})(x - x_{i+2}), \tag{C3}$$

where each interpolant corresponds to the cubic polynomial fit to the data given on one of the three candidate stencils $S_1 = \{x_{i-2}, \ldots, x_{i+1}\}$, $S_2 = \{x_{i-1}, \ldots, x_{i+2}\}$, and $S_3 = \{x_i, \ldots, x_{i+3}\}$ (see Fig.C1). Convex combination of these interpolants gives the WENO interpolant

$$I_{\text{WENO6}}(x) = \omega_1(x)\, p_1(x) + \omega_2(x)\, p_2(x) + \omega_3(x)\, p_3(x), \tag{C4}$$

where $\omega_k(x),\ k = 1,2,3$ are the required weights. In a smooth problem, all the point data should be used to obtain an interpolation which is as high order as possible, i.e., that agrees with the degree five interpolating polynomial through all six points. These "ideal" weights $d_k(x),\ k = 1,2,3$ are given by

$$\begin{aligned} d_1(x) &= \frac{(x_{i+2} - x)(x_{i+3} - x)}{20\Delta x^2}, \\ d_2(x) &= \frac{(x_{i+3} - x)(x - x_{i-2})}{10\Delta x^2}, \\ d_3(x) &= \frac{(x - x_{i-2})(x - x_{i-1})}{20\Delta x^2}. \end{aligned} \tag{C5}$$

Note that unlike WENO for hyperbolic conservation laws presented in Chapter 2, here the interpolation point x is not fixed and the values of the ideal weights depend on x. Still, these weights $d_k(x)$ are completely analogous to the well-known "1/10, 6/10, 3/10" weights when $x = x_{i+1/2}$.

In non-smooth regions, at least one of the interpolations $p_k(x),\ k = 1,2,3$ will be superior to an interpolation with the "ideal" values because of the problems associated with fitting high-order polynomials to non-smooth data – namely highly oscillatory results. To decide which stencils to use, we compute a smoothness indicator for each interpolant. We take the smoothness indicator IS_k for interpolant p_k as a sum of squares of scaled L_2 norms of all the derivatives of the interpolant p_k over the interval of interpolation. Specifically,

$$IS_k(x) = \sum_{j=1}^{3} \int_{x_k}^{x_{k+1}} (\Delta x)^{2j-1} \left(\frac{d^j p_k(x)}{dx^j} \right)^2 dx. \tag{C6}$$

If a particular interpolant exhibits rapid change on the interval (x_k, x_{k+1}) compared to the other two interpolants, then it will have larger-in-magnitude derivatives on that interval, which in turn increases the corresponding smoothness indicator (C6). Desirable smooth interpolants exhibit less drastic changes in their derivatives and thus minimize (C6). If all three candidate

interpolants are smooth, then all three smoothness indicators will have similar small values. For completeness, (C6) can be worked out as

$$IS_1(x) = (814 f_{i+1}^2 + 4326 f_i^2 + 2976 f_{i-1}^2 + 244 f_{i-2}^2 - 3579 f_i f_{i+1} - 6927 f_i f_{i-1} + \\ + 1854 f_i f_{i-2} + 2634 f_{i+1} f_{i-1} - 683 f_{i+1} f_{i-2} - 1659 f_{i-1} f_{i-2}) / 180, \tag{C7}$$

$$IS_2(x) = (1986 f_{i+1}^2 + 1986 f_i^2 + 244 f_{i-1}^2 + 244 f_{i+2}^2 + 1074 f_i f_{i+2} - 3777 f_i f_{i+1} + \\ - 1296 f_i f_{i-1} + 1074 f_{i+1} f_{i-1} - 1269 f_{i+2} f_{i+1} - 293 f_{i+2} f_{i-1}) / 180, \tag{C8}$$

$$IS_3(x) = (814 f_i^2 + 4326 f_{i+1}^2 + 2976 f_{i+2}^2 + 244 f_{i+3}^2 - 683 f_i f_{i+3} + 2634 f_i f_{i+2} \\ - 3579 f_i f_{i+1} - 6927 f_{i+1} f_{i+2} + 1854 f_{i+1} f_{i+3} - 1659 f_{i+2} f_{i+3}) / 180. \tag{C9}$$

We note as expected that the smoothness indicators do not depend on the particular point of interpolation x because they measure a property of the interpolant candidates themselves. The computation of the weights is carried out using the smoothness indicators as in the standard WENO procedure by first calculating

$$\alpha_k(x) = \frac{d_k(x)}{(\varepsilon + IS_k)^2}, \quad k = 1, 2, 3, \tag{C10}$$

where ε is a small parameter to prevent division-by-zero in the case when all IS_k; we use $\varepsilon = 10^{-6}$ in all our calculations. Finally, the weights are

$$\omega_k(x) = \frac{\alpha_k(x)}{\alpha_1(x) + \alpha_2(x) + \alpha_3(x)}, \quad k = 1, 2, 3. \tag{C11}$$

Two-dimensional WENO interpolation is built in the standard fashion from one-dimensional interpolations. First, we perform six one-dimensional interpolations in x-direction and we obtain six x-coordinate values that agree with the interpolation point. Then one-dimensional interpolation is carried out on these six points to get the desired interpolated value. See Fig.C2 for illustration. Three-dimensional case is treated in a similar dimension-by-dimension manner.

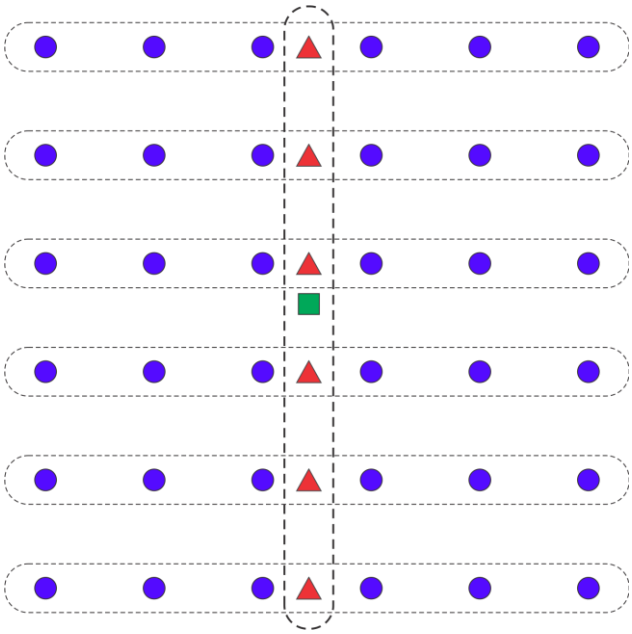


Figure C2: Two-dimensional WENO interpolation.

Bibliography

[1] Adalsteinsson, D. & Sethian, J.A. (1995). A fast level set method for propagating interfaces. *J. Comput. Phys*. **118**: 269-277.

[2] Adalsteinsson, D. & Sethian, J.A. (1999). The fast construction of extension velocities in level set methods. *J. Comput. Phys*. **148**: 2-22.

[3] Anderson, D.M., McFadden, G.B. & Wheeler, A.A. (1998). Diffuse-interface methods in fluid mechanics. *Ann. Rev. Fluid Mech.* **30**: 139-165.

[4] Arnold, V.I. (1983). *Geometrical Methods in the Theory of Ordinary Differential Equations*. Springer-Verlag, New York.

[5] Bell, J.B., Colella, P., Glaz, H.M. (1989). An efficient second-order projection method for the incompressible Navier-Stokes equation. *J. Comput. Phys.* **85**: 257-283.

[6] Chen, Y.-N., Yang, S.-C & Yang, J.-Y. (1999). Implicit weighted essentially non-oscillatory schemes for the incompressible Navier–Stokes equations. *Int. J. Numer. Meth. Fluids* **31**: 747-765.

[7] Cheng, L.-T. & Tsai, Y.-H. (2008). Redistancing by flow of time-dependent Eikonal equation. *J. Comput. Phys*. **227**: 4002-4017.

[8] Chopp, D.L. (1993). Computing minimal surfaces via level set curvature flow. *J. Comput. Phys.* **106**: 77-91.

[9] Chopp, D.L. (2009). Another look at velocity extensions in the level set method. *SIAM J. Sci. Comput*. **31**: 3255-3273.

[10] Chorin A.J. (1967) A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.* **2**: 12-26.

[11] Chorin, A.J. (1968). Numerical solution of Navier-Stokes equations. *Mathematics of Computation* **22**: 745-762.

[12] Couderc, F. (2007). *Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles. Application à la désintégration assistée d'un jet*

liquide par un courant gazeux. Ph.D. thesis, Ecole nationale supérieure de l'aéronautique et de l'espace.

[13] Cummins, S.J., Francois, M.M. & Kothe, D.B. (2005). Estimating curvature from volume fractions. *Computers and Structures* **83**: 425-434.

[14] Daly, B.J. (1967). Numerical study of two fluid Rayleigh-Taylor instability. *Phys. Fluids* **10**: 297-307.

[15] Daly, B.J. & Pracht, W.E. (1968). Numerical study of density current surges. *Phys. Fluids* **11**: 15-30.

[16] Deng, X. & Zhang, H. (2000). Developing high-order weighted compact nonlinear schemes. *J. Comput. Phys*. **165**: 22-44.

[17] Desjardins, O. (2008). *Numerical methods for liquid atomization and application in detailed simulations of a diesel jet*. Ph.D. thesis, Stanford University.

[18] Desjardins, O., Moureau, V. & Pitsch, H. (2008). An accurate conservative level set/ ghost fluid method for simulating turbulent atomization. *J. Comput. Phys*. **227**: 8395-8416.

[19] Desjardins, O. & Pitsch, H. (2009). A spectrally refined interface approach for simulating multiphase flows. *J. Comput. Phys.* **228:** 1658-1677.

[20] Engquist, B. & Sjogreen, B. (1998). The convergence rate of finite difference schemes in the presence of shocks. *SIAM J. Numer. Anal*. **35**: 2464-2485.

[21] Enright, D., Fedkiw, R., Ferziger, J, Mitchell, I. (2002). A hybrid particle level-set method for improved interface capturing. *J. Comput. Phys*. **183**: 83-116.

[22] Ferziger, J. H. & Peric M. (2002). *Computational Methods for Fluid Dynamics*. Springer: New-York.

[23] Francois, M.M., Cummins, S.J., Dendy, E.D., Kothe, D.B. *et al*. (2006). A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys* **213**: 141-173.

[24] Ghia, U., Ghia, K. & Shin, C. (1982). High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, *J. Comput. Phys.* **48**: 387-411.

[25] Giga, Y. (2006). *Surface Evolution Equations. A Level Set Approach*. Birkhäuser-Verlag.

[26] Glimm, J., Grove, J.W., Li, X.L, Shyue, K.-M., Zeng, Y. & Zhang, Q. (1998). Three-dimensional front tracking. SIAM J. Sci. Comput. **19**: 703-727.

[27] Gottlieb, S., Shu, C.-W. (1998): Total variation diminishing Runge-Kutta schemes. *Math. Comput*. **67**, 73–85.

[28] Gomez, P., Hernandez, J. & Lopez, J. (2005). On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows. *Int. J. Numer. Meth. Eng*., **63**: 1478-1512.

[29] Gomes, J. & Faugeras, O. (2000). Reconciling distance functions and level sets. *J. Vis. Commun. and Image Repres.* **11**: 209-223.

[30] Harlow, F. & Welch, J. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces. *Phys. Fluids*, **8**: 2182-2189.

[31] Hartmann, D., Meinke, M. & Schroder W. (2008). Differential equation based constrained reinitialization for level set methods. *J. Comput. Phys*., **227**: 6821-6845.

[32] Hartmann, D., Meinke, M. & Schroder, W. (2010). The constrained reinitialization equation for level set methods. *J. Comput. Phys*., **229**: 1514-1535.

[33] Henrick, A.K., Aslam, T.D. & Powers, J.M. (2005). Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. *J. Comput. Phys.* **207**: 542-567.

[34] Herring, J.R., Orszag, S.A., Kraichnan R.H. & Fox, D.G. (1974). Decay of two-dimensional homogeneous turbulence. *J. Fluid Mech*. **66**: 417-444.

[35] Herrmann, M. (2008). A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comput. Phys*. **227**: 2674-2706.

[36] Hirt, CW, Amsden, A.A. & Cook, J.L. (1974) Arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys*. **14**: 227-253.

[37] Hirt, C.W. & Nichols, B.D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys*., **39**: 201-225.

[38] Ichikawa, Y., Otawara, Y., Kobavashi, H., Ogami, Y (2011). Flame structure and radiation characteristics of $CO/H_2/CO_2$/air turbulent premixed flames at high pressure. *Proceedings of the Combustion Institute,* **33**: 1543-1550.

[39] Jacqmin, D. (1999). Calculation of two-phase Navier-Stokes flows using phase-field modeling. *J. Comput. Phys*., **155**: 96-127.

[40] Jiang, G.-S. & Peng, D. (2000). Weighted ENO schemes for Hamilton-Jacobi equations, *SIAM J. Sci. Comput.*, **21**: 2126-2143.

[41] Jiang, G.-S. & Shu, C.-W. (1996). Efficient implementation of weighted ENO schemes, *J. Comput. Phys*, **126**: 202-228.

[42] Kang, M., Fedkiw, R. & Liu, X.D. (2000). A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput*., **15**: 323-360.

[43] Landau, L. D. & Lifshitz, E. M. (1978). Fluid Mechanics, Pergamon Press, NY.

[44] Lowengrub, J. & Truskinovsky, L. (1998). Quasi-incompressible Cahn–Hilliard fluids and topological transitions. *Proc. R. Soc. London Ser. A.* **454**: 2617-2654.

[45] Malladi, R. & Sethian, J.A. (1995). Image processing via level set curvature flow. Proc. Natl. Acad. Sci. 92: 7046-7050.

[46] Martin, M., Taylor, E., Wu, M. & Weirs, V. (2006). A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence, *J. Comput. Phys*. **220**: 270-289.

[47] Minion, M.L. & Brown, D.L. (1997). Performance of under-resolved two-dimensional incompressible flow simulations II. *J. Comput. Phys.* **138**: 734-765.

[48] Olsson, E. & Kreiss G. (2005). A conservative level set method for two phase flow. *J. Comput. Phys*., **210**: 225-246.

[49] Olsson, E., Kreiss, G. & Zahedi, S. (2007). A conservative level set method for two phase flow II. *J. Comput. Phys*., **225**: 785-807.

[50] Osher, S. & Sethian, J.A (1988). Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys*., **79**: 12-49.

[51] Osher, S. & Fedkiw, R. (2001). Level set methods: an overview and some recent results. *J. Comput. Phys*., **169**: 463-502.

[52] Osher, S. & Fedkiw, R. (2003). *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York.

[53] Ostapenko, V.V. (2010). On convergence of high order shock capturing difference schemes. *AIP Conf. Proc*. **1301**: 413-425.

[54] Peng, D.P., Merriman, B., Osher, S., Zhao, H.K. & Kang M.J. (1999). A PDE-based fast local level set method. *J. Comput. Phys*., **155**: 410-438.

[55] Peskin, C. (1977). Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **25**: 220.

[56] Pilliod, J.E. & Puckett E.G. (2004) Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys*. **199**: 465-502.

[57] Pirozzoli, S. (2002). Conservative hybrid compact-WENO schemes for shock-turbulence interaction. *J. Comput. Phys*. **178**: 81-117.

[58] Prosperetti, A. (1981). Motion of two superposed viscous fluids. *Phys. Fluids* **24**: 1217-1223.

[59] Quan, S., Lou J. & Schmidt D.P. (2009) Modeling merging and breakup in the moving mesh interface tracking method for multiphase flow simulations. *J. Comput. Phys.* **228**: 2660-2675.

[60] Rider, W.J. & Kothe, D.B. (1998). Reconstructing volume tracking. *J. Comput. Phys*. **141**: 112-152.

[61] Russo, G. & Smereka, P. (2000). A remark on computing distance functions. *J. Comput. Phys.* **163**: 51-67.

[62] Rutland, D. F. & Jameson, G. J. (1971). A non-linear effect in the capillary instability of liquid jets. *J. Fluid Mech*. **46**: 267-271.

[63] Sethian, J.A. (1996). *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Science*. Cambridge University Press, London.

[64] Sethian, J.A. & Smereka, P. (2003). Level set methods for fluid interfaces, *Annu. Rev. Fluid Mech.*, **35**: 341-372.

[65] Scardovelli, R & Zaleski, S. (1999). Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech*. **31**: 567–603.

[66] Shetty, D.A., Fisher, T.C., Chunekar, A.R., Frankel, S.H. (2010). High-order incompressible large-eddy simulation of fully inhomogeneous turbulent flows, *J. Comput. Phys*., **229**: 8802-8822.

[67] Shu, C.-W. & Osher, S.J. (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys*., **77**: 439-471.

[68] Sussman, M, Smereka, P & Osher, S. (1994). A level set approach to computing solutions to incompressible two-phase flow. *J. Comput. Phys*. **114**: 146-159.

[69] Sussman, M, Fatemi, E., Smereka, P. & Osher S. (1998). An improved level set method of incompressible two-fluid flows. *Comput. Fluids* **27**: 663–680.

[70] Sussman, M. & Fatemi, E. (1999). An efficient interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput*. **20**: 1165-1191.

[71] Sussman, M. & Puckett, E.G. (2000). A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys*. **162**: 301-337.

[72] Tan, L. & Zabaras, N. (2007). A level set simulation of dendritic solidification of multi-component alloys. *J. Comput. Phys*. **221**: 9-40.

[73] Tanguy, S. (2004). *Développement d'une méthode de suivi d'interface. Applications aux écoulements diphasiques*. Ph.D. thesis, Université de Rouen - CORIA.

[74] Temam, R. (1968). Une méthode d'approximation des solutions des équations Navier-Stokes. *Bull. Soc. Math. France* **98**: 115-152.

[75] Tryggvason, G., Bunner, B., Esmaeeli, A, Juric, D, Al-Rawahi, N, *et al.* (2001). A front-tracking method for the computations of multiphase flow. *J. Comput. Phys*. **169**: 708–759.

[76] Unverdi, S.O. & Tryggvason, G. (1992). A front-tracking method for viscous, incompressible, multifluid flows. *J. Comput. Phys*. **100**: 25-37.

[77] Wu, C.C. (2007). A high order WENO finite difference scheme for incompressible fluids and magnetohydrodynamics. *Geophys. Astrophys. Fluid Dynam*. **101**: 37-61

[78] Zalesak, S.T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comp. Phys.* **31**: 335–362.

[79] Zhang, Y. & Jackson, T.L. (2009). A high-order incompressible flow solver with WENO, *J. Comput. Phys*. **228**: 2426-2442.

[80] Zhao, H.K., Chan, T., Merriman, B. & Osher, S. (1996). A variational level set approach to multiphase motion. *J. Comput. Phys*. **127**: 179-195.