이학박사 학위논문

Research on high order Lagrangian interface tracking problems

(고차 정확도 라그랑지안 표면 추적 문제에 대한 연구)

2018년 2월

서울대학교 대학원

수리과학부

김현욱

Research on high order Lagrangian interface tracking problems

(고차 정확도 라그랑지안 표면 추적 문제에 대한 연구)

지도교수 강 명 주

이 논문을 이학박사 학위논문으로 제출함

2017년 10월

서울대학교 대학원

수리과학부

김현욱

김현욱의 이학박사 학위논문을 인준함

2017년 12월

위 원 장 _____ (인)

부 위 원 장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Research on high order Lagrangian interface tracking problems

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Hyeonuk Kim

Dissertation Director : Professor Myungjoo Kang

Department of Mathematical Sciences
Seoul National University

February 2018

Abstract

In this thesis, we proposed the method for tracking the interface with higher order accuracy, when given the normal velocity. Our proposed model combines the well-known Lagrangian surface tracking method with the high order interpolation method for discontinuity capturing. The proposed method not only tracks the surface with higher accuracy than the conventional method, but also depends little on the geometric parameters. Furthermore, the method accurately detects the local shapes of the surface, which is an essential part for a stable interface tracking method. The model developed on the two-dimensional interface can be extended naturally on the three-dimensional interface using the high order interpolation method on triangular meshes.

Key words: Lagrangian interface tracking, wavefrontal motion, local shape detection, face offsetting method, WENO reconstruction, FOM-WENO scheme
Student Number: 2012-20248

Contents

Chapter 1

Introduction

Accurate interface tracking and interface capturing among distinct substances are important in many physical processes such as solidification, multiphase flow, and have a plenty of engineering applications. Generally, interface representation methods are classified into two categories, interface tracking and interface capturing. Explicit methods(or Lagrangian methods) usually track the interface, and implicit methods(or Eulerian methods) usually capture the interface.

The level set method [1] is a famous and typical Eulerian method for propagating interfaces. It was introduced by S. Osher and J. A. Sethian [2] in 1988, and many follow-up papers were published in last three decades [3, 4, 5, 6, 7]. The level set method is able to capture the presence of boundary singularities and topological changes. Its basic idea is based on implicit function ϕ, whose 0-level set represents the interface one want to track for. Then the evolution of the interface can be done by solving simple Hamiton-Jacobi equation. The method also has the advantage in that one can express many useful geometric information, e.g. normal vector, mean curvature using function ϕ. Above all, the most important advantage is an automatical treatment for topological change of the interface, which do not require special mesh surgery process. However, traditional level set method has relatively large computational cost, since its computation is done not in the interface neighborhood but in the whole domain for the implicit function ϕ. Moreover, the accuracy of the method at the singular boundary point is not good. To overcome such a weakness, local level set method [3] and particle level set method [5] were introduced to overcome computation cost problem and

smearing out problem, respectively.

Lagrangian methods also have been studied extensively. These are more accurate and computationally cheaper than Eulerian method with similar resolution, thus enabling efficient interface tracking. Papers on front-tracking methods in multiphase flow [8, 9, 10, 11] have been read by many researchers. Shin and Juric [12] enforced mass conservation and enabled to handle topological change in film rupture or filamentary breakup simulations. However, almost all Lagrangian methods use ad hoc techniques in order to handle a topology change. Although several researchers have devised a solution [13, 14, 15], but it is still a difficult problem under Lagrangian framework. Nevertheless, the Lagrangian method is often used because of its less diffusivity and high accuracy [16, 17].

One method of Lagrangian methods, the face-offsetting method(FOM) was introduced by X. Jiao [18]. It requires only an explicit surface mesh, but not a volume mesh. FOM scheme moves the face with a given speed and then reconstructs the vertices by an eigenvalue analysis. The local geometry near each vertex is determined in the process of eigenvalue analysis, and each vertex movement is restricted depending on the type of local geometry determined. In wavefrontal motion, the entropy-satisfying Huygens' principle must be satisfied and the way vertex moves is different according to whether it is contracting or expanding. To implement this, the FOM scheme checks whether the neighbor face is expanding or contracting at each point, and adjusts the position of the reconstructed vertex. Finally, to maintain mesh quality during propagation, the vertex is redistributed along a tangential direction. This process is called null-space smoothing. The FOM scheme is very effective method when moving the interface in the normal direction. It can effectively round the expanding corner according to the Huygens' principle while moving the boundary more precisely than the Eulerian methods.

The WENO scheme [19, 20] is a high order reconstruction method first introduced in [21], which is an evolution of the ENO scheme [22]. Both methods are based on high order interpolation and are improved to avoid numerical oscillation near discontinuity. In the ENO scheme, only the smoothest stencil is found and used for interpolation. In the WENO scheme, however, it utilize a smooth indicator to calculate smoothness and then determine appropriate weights according to smooth indicator to combine all possible stencils. The WENO scheme thus developed prevents the Gibbs phenomenon from occurring near discontinuity while ensuring high order in the smooth region. WENO schemes on unstructured meshes have also been developed in [23, 24]. Recently, various variants of WENO schemes, such as [25, 26, 27], have been actively researched as well.

In this thesis, we have combined the conventional FOM scheme and the WENO scheme described above to develop a new high order interface tracking method. Determining the offset direction using the normals of the neighboring face in the FOM scheme is very similar to determining the numerical flux at the cell boundary by solving the Riemann problem from the fluxes of the left and right cells in many CFD problems (details are in [28]). The WENO scheme is usually used to compute the numerical flux of the interface with high order in the left and right cells, and the numerical flux at the interface can also be approximated with a high order. In FOM-WENO method, the normal vectors at the vertex are approximated with high order and then use them to calculate the offset direction more accurately, to develop a higher order FOM scheme.

Since we utilize the WENO scheme, the FOM-WENO scheme can effectively find normal vectors without oscillations near sharp features such as ridges and corners. We experimentally show that our methodology can track each point more accurately than the conventional FOM scheme and that the resulting volume loss is much less than that of the conventional FOM scheme. In addition, the high order approximation of the normal vector will make our methodology less sensitive to special geometric parameters used in conventional FOM scheme, so we will be able to find local geometry more reliably than the conventional FOM scheme. We will compare the FOM-WENO with conventional FOM scheme in various surface propagation problems. We will also show natural extensibility of the FOM-WENO scheme over a triangular surface mesh.

After the introduction part in Chapter 1, the thesis consists of the following: Chapter 2 describes the basic numerical preliminaries, such as the level set method, the face offsetting method, the WENO scheme on 1D interval and on 2D triangular meshes, and the third order TVD Runge-Kutta method [29, 30] used for time stepping. In Chapter 3, we propose our FOM-WENO scheme and analyze it mathematically. Numerical results on various surfaces are then presented and compared with the existing FOM scheme. Finally, we will finish our thesis with concluding remarks in Chapter 4.

Chapter 2

Previous works

2.1 Level set method

Before introducing the face offsetting method, one of the famous Lagrangian interface tracking method, we first review the Eulerian counterpart, level set method.

2.1.1 Basic equations

Level set method is first introduced by Osher and Sethian in [2]. The main idea is to take implicit function ϕ for expressing and moving the interface. The interface can be described in the following manner.

Let Γ be the common boundary between two regions Ω^+ and Ω^-. A smooth function $\phi : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}$ is then defined by:

$$\begin{aligned} \phi(x,t) > 0 \quad &\text{if} \quad x \in \Omega^+, \\ \phi(x,t) = 0 \quad &\text{if} \quad x \in \Gamma, \\ \phi(x,t) < 0 \quad &\text{if} \quad x \in \Omega^-. \end{aligned}$$

Note that the 0-level set of ϕ is exactly Γ, which corresponds to the interface we want to track. This is where the name level set method originates from. Now, implicit function ϕ can be evolved with the following simple Hamilton-Jacobi equation for a given velocity field $\mathbf{V}$:

$$\phi_t + \mathbf{V} \cdot \nabla\phi = 0.$$

This is the Eulerian formulation that expresses the motion of the interface under the velocity field $\mathbf{V}$ using the implicit function ϕ. We call it the level set equation.

In this thesis, we especially treat the case when the normal velocity is given. In particular, when the interface moves with a constant speed a in the normal direction, the velocity field can be written as $\mathbf{V} = a\mathbf{N}$. Here, $\mathbf{N}$ denotes the outward normal of the interface given. The corresponding level set equation is

$$\phi_t + a|\nabla\phi| = 0.$$

2.1.2 Numerical discretization

In the previous section, we briefly summarized the level set equation for evolving the interface in the normal direction. To discretize level set equation using the simple upwind scheme, it is necessary to express the equation in the form of advection equation. The key observation to achieve this is that the velocity field is $\mathbf{V} = a\mathbf{N} = a\dfrac{\nabla\phi}{|\nabla\phi|}$. We now rewrite the above equation in the following form.

$$\phi_t + \left(\frac{a\phi_x}{|\nabla\phi|}, \frac{a\phi_y}{|\nabla\phi|}, \frac{a\phi_z}{|\nabla\phi|}\right) \cdot \nabla\phi = 0. \tag{2.1.1}$$

This equation corresponds to advection equation with the velocity field $a|\nabla\phi|^{-1}\nabla\phi$. For brevity, we only consider the x velocity term $a\phi_x|\nabla\phi|^{-1}$. To discretize this term with an upwind scheme, we should consider the sign of the term. Since $|\nabla\phi|$ is always positive, the sign of $a\phi_x$ determines whether ϕ_x^- or ϕ_x^+ is used to discretize ϕ_x. Especially, when the signs of ϕ_x^- and ϕ_x^+ are same, the sign of $a\phi_x$ is also automatically determined. For example, if both ϕ_x^- and ϕ_x^+ are positive, then $a\phi_x$ is also positive (assume $a > 0$ for convenience), so we can set $\phi_x = \phi_x^-$. However, when the signs of ϕ_x^- and ϕ_x^+ are different, the sign of $a\phi_x$ cannot be determined directly and we need more procedure to determine the approximation for ϕ_x.

In general, the above equation is discretized using the Godunov's method. For brevity, the details are omitted and only the results are shown here. If $a\phi_x^+$ and $a\phi_x^-$ are both positive, than use $\phi_x = \phi_x^-$. If $a\phi_x^+$ and $a\phi_x^-$ are both negative, than use $\phi_x = \phi_x^+$. If $a\phi_x^+ \leqslant 0$ and $a\phi_x^- \geqslant 0$, the situation corresponds to rarefaction, and we set $\phi_x = 0$. Finally, if $a\phi_x^+ \geqslant 0$ and $a\phi_x^- \leqslant 0$, the shock occurs, then we compare the absolute values of $a\phi_x^+$ and $a\phi_x^-$, and

use larger one to discretize ϕ_x. We can express the results in the following compact form.

When $a > 0$,

$$\phi_x^2 \approx \max\left(\max(\phi_x^-, 0)^2, \min(\phi_x^+, 0)^2\right).$$

When $a < 0$,

$$\phi_x^2 \approx \max\left(\min(\phi_x^-, 0)^2, \max(\phi_x^+, 0)^2\right).$$

In order to finalize the spatial discretization, we should approximate ϕ_x^+ and ϕ_x^-. There are various methods for this approximation, and one of the famous method is 5-th order WENO method, which will be introduced in later section. We also use the Runge-Kutta method in order to achieve higher order in time variable, and we will also introduce it in a subsequent section.

2.1.3 Reinitialization

Equation (2.1.1) reduces to a simple ODE $\phi_t = -a$ when ϕ is a signed distance function. From this equation, we see that ϕ becomes larger or smaller according to the sign of a. In particular, if a is a constant, the equation has $\phi(t) = \phi_0 – at$ as the exact solution. In general, if ϕ is a signed distance function, then the level set method becomes very simple. Therefore, there is a huge advantage for maintaining ϕ to be closer to the signed distance function.

Of course the level set equation works well even if ϕ is not a signed distance function. However, as ϕ moves along the level set equation, the noise gradually increases and the derivative also becomes steeper, which inevitably damages the finite difference scheme. On the other hand, if the value of ϕ is kept close to the signed distance function, the magnitude of the gradient of ϕ is kept at 1 and ϕ stays sufficiently smooth function, so that it can be approximated to the desired accuracy for discretization. Note that even if the initial ϕ value is set to be a signed distance function, ϕ is not necessarily a signed distance function since ϕ moves away from the signed distance as the interface moves. Therefore, we need an additional reinitialization technique periodically to maintain ϕ as a signed distance function.

When reinitializing level set, we require that ϕ to be a signed distance function with the $\phi = 0$ on the interface we are tracking. Sussman, Smereka, and Osher [31] devised the following reinitialization equation.

$$\phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0. \tag{2.1.2}$$

In equation (2.1.2), τ is a virtual time step for reinitializing ϕ, and $S(\phi_0)$ is the sign function which is 1 in Ω^+, -1 in Ω^-, and 0 on the interface Γ. In practice, we can get better results by using mollification of $S(\phi_0)$ near the interface. For example, (A) used the following instead of original sign function.

$$S(\phi_0) = \frac{\phi_0}{\phi_0^2 + (\Delta x)^2}.$$

Peng, Merriman, Osher, Zhao, and Kang [3] suggested the following molifier

$$S(\phi_0) = \frac{\phi_0}{\phi_0^2 + |\nabla\phi_0|^2(\Delta x)^2}.$$

It seems that artificially inserting numerical smearing in the sign function gives better results because it slows down the information transfer rate near the interface, preventing the characteristic from skipping over the interface. As before, $S(\phi_0)|\nabla\phi|$ is generally discretized using the Godunov's method, and the Runge-Kutta method is used for time variable.

2.2 Face offsetting method

The face-offsetting method(FOM) is first introduced by X. Jiao [18]. It is a Lagrangian interface tracking method with vertices reconstruction using an eigenvalue analysis. Let Γ^0 be an orientable surface, which is constructed with 0-dimensional vertices, 1-dimensional edges, and 2-dimensional faces. A normal speed $f(\mathbf{x}, t) : \Gamma \times \mathbb{R}^+ \to \mathbb{R}$ is given for each face and the interface is propagated under the following equation :

$$\frac{\partial \mathbf{x}}{\partial t} = f\mathbf{n}(\mathbf{x}, t),$$

where $\mathbf{n}$ is the outward unit normal vector. Since the normal vector is computed from the local geometry, usually this problem is much difficult than propagating interfaces under given velocity field.

Now, we briefly introduce the core of FOM scheme. We first propagate each face under given normal speed. Then using eigenvalue analysis, the local shape near each vertex is determined so that the new vertices are located on the propagated face. The normalized vertex displacement at this time is also called the offset direction. Finally, the scheme corrects the vertices according to whether the type of motion is expanding or contracting. Each time one move the vertices, one will redistribute vertices in a tangential direction to preserve mesh quality during propagation.

2.2.1 Advection type

In FOM scheme, neighboring faces for given vertex are moved according to their normal velocity first, then the intersection of moved faces is calculated as least square sense, and the vertex is moved to the solution of least square problem. However, if the vertices are in a smooth region of the interface, such intersection of such faces may occur at odd locations that are far from the original vertex position, which can lead to severe degradation of mesh quality or even break the mesh altogether. The conventional FOM scheme uses the concept of primary space to overcome this problem. The primary space is defined by the eigenvalue analysis, which defines the space in which the vertex should move according to its local geometry. By projecting the previously computed intersection point to the primary space, we can reliably calculate the vertex displacement.

Consider a vertex v in 3-dimensional space. A plane containing point $\mathbf{p}$ with normal vector $\mathbf{n}$ can be represented by $\mathbf{n}^T\mathbf{x} = \mathbf{n}^T\mathbf{p}$. Suppose v has m neighborhood faces. We refer to the new location of all the points of a face

as its face offset. Then, provided that there is no topological change, new position of a vertex v has to be contained in the offsets of m neighborhood faces. It is the solution of the $m \times 3$ linear system

$$\mathbf{Nx} = \mathbf{a},$$

of which each row corresponds to each neighborhood face of v. In general, $m \geqslant 3$ and we need to solve Eq.(2) in a weighted least square sense. Then we have linear system

$$\mathbf{Ax} = \mathbf{b},$$

where $\mathbf{A} = \mathbf{N^T W N}$, $\mathbf{b} = \mathbf{N^T W a}$, and $\mathbf{W} = (\mathbf{W}_{ij})$ is an $m \times m$ diagonal matrix with $\mathbf{W}_{ii}$ is the weight of the i-th neighborhood face of v. Here, weights $\mathbf{W}_{ii}$ are generally taken as the face area of corresponding face.

As mentioned before, we carry out an eigenvalue analysis for matrix $\mathbf{A}$, which is symmetric and positive semi-definite. Let λ_i be real and non-negative eigenvalues and e_i be corresponding eigenvectors. Assume $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$. We can obtain geometric information from these eigenvalues and eigenvectors. The fact that the eigenvalues corresponding to the eigenvectors are large means that the normal directions of the neighborhood faces are relatively directed toward the direction of the eigenvector. We classify local geometry near vertices into three categories. They are summarized in table 2.1(Interface in 2D can be regarded as consisting of the smooth region and the ridge only). In each case, the characteristics of eigenvalues are listed.

Smooth region	A relatively large λ_1 and small λ_2, λ_3 $\lambda_1 \gg \lambda_2 \geqslant \lambda_3$
Ridge	Relatively large λ_1, λ_2 and a small λ_3, $\lambda_1 \geqslant \lambda_2 \gg \lambda_3$
Sharp corner	Three eigenvalues are of similar magnitude, $\lambda_1 \approx \lambda_2 \approx \lambda_3$

Table 2.1: Classification of local geometries and their characteristics in eigenvalue

Therefore, we can classify local flatness of surface by comparing the magnitude of eigenvalues. If we refer to the vector space spanned by eigenvectors corresponding to relatively large(whose meaning will be clear later on) eigenvalues of $\mathbf{A}$ as its primary space, the vertex displacement within the primary

space of dimension k is computed by

$$\mathbf{d}_{\text{adv}} = \sum_{i=1}^{k} \mathbf{e}_i^t \mathbf{be_i} / \lambda_i.$$

The following figure 2.1 shows the primary space for three types of local geometry introduced above. For a point lying on the smooth region, the primary space is a one dimensional vector space spanned by a normal vector at the point. For a point lying on the ridge, the primary space bisects the two surfaces two surfaces constituting the ridge. Finally, for a point lying at corner, the primary space is $\mathbb{R}^3$.

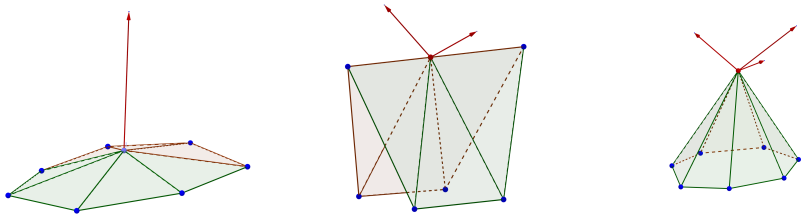

Figure 2.1: The eigenvectors corresponding to relatively large eigenvalues and the primary space for (a) Smooth. (b) Ridge. (c) Corner

The dimension k is determined from the relative magnitude of three eigenvalues, for example, one can compute λ_2/λ_1 and λ_3/λ_1. To set some thresholds to classify a ridge and a corner based on geometry, we also consider the following various geometric quantity. First, one can consider the sum of the angles at the vertex in its incident faces, denoted by θ_s, which can be used to detect corner. $\theta_s \gg 2\pi$ means the vertex is located at a corner. Let $\theta_a = \theta_s - 2\pi$. θ_a alone is not enough to distinguish whether the vertex lies on a ridge. To classify the vertex more rigorously, let $\mathbf{e}_1$be the first eigenvector with $\mathbf{e}_1^T \mathbf{b} > 0$. Then the following criterion may be useful.

1. if $\lambda_3/\lambda_1 \geqslant \chi_c$ or $|\theta_a| \geqslant \pi/2$, the v is a corner;
2. if $\lambda_2/\lambda_1 \geqslant \chi_r$ or $\mathbf{e}_1^T \mathbf{n} \leqslant 0$ in incident face, then v is on a ridge.

Depending on the size of χ_c, there is a possibility for all vertices to be classified as corners, or no vertices are classified as corners. The same phenomena can happen when classifying ridges. Furthermore, classifying results react very sensitive to these parameters in some geometry. Reasonable choice of χ_r and χ_c is related to the dihedral angle and open angle of a cone. If

a vertex v on a ridge has almost uniform weights, the dihedral angle θ_d is less than $\pi/2$ and the eigenvalues satisfy $\lambda_2/\lambda_1 \approx \tan^2(\theta_d/2)$, $\lambda_3 \approx 0$. If a vertex v is located at a corner with an open angle $\pi - \theta_o$, the eigenvalues satisfy $\lambda_3/\lambda_1 \approx \lambda_2/\lambda_1 \approx \tan^2(\theta_o/2)/2$. From this observation, if we set a dihedral angle thresholds ϕ_r and open angle thresholds ϕ_c, we can define $\chi_r = \tan^2(\phi_r/2)$ and $\chi_c = \tan^2(\phi_c/2)/2$. The values of ϕ_r and ϕ_c are determined by how the initial geometry is given. For example, X.Jiao uses $\phi_r = 15^\circ$ and $\phi_c = 45^\circ$ in [32, 18], then $\chi_r \approx 0.0173$ and $\chi_c \approx 0.0858$. In this case, interface in 2D can be regarded as having only χ_r. Experimentally, it is known that it is more sensitive to determine the value of χ_c than the value of χ_r.

2.2.2 Wavefrontal type

Before we discuss wavefrontal type propagations, it is essential to distinguish contracting and expanding motions, which are illustrated in figure 2.2. In case of contracting motions, the vertex displacement $\mathbf{d}_{\text{wav}}$ is identical to aforementioned advection type, which is determined by the intersection of the moved faces. Therefore, total displacement of each vertex at each step is equal to $\mathbf{d}_{\text{adv}}$. However, for expanding motion case, a corner point tends to be rounded during propagation, and the vertex displacement computed as if it were contracting motion is too long. In other words, to implement wavefrontal motion, we must modify the previously computed vertex displacement. Experimentally, $\mathbf{d}_{\text{adv}}$ is correctly identifies the direction in which the vertex moves. Therefore, it is enough for adjusting length only according to the type of motions. We first determine whether the i-th neighborhood face is expanding or contracting with respect to the vertex from the following simple formula. If $\mathbf{s}_i$ is a vector that connects up to the opposite edge of the vertex for the face moved from the vertex moved along $\mathbf{d}_{\text{adv}}$, the face is contracting with respect to the vertex if $\mathbf{d}_{\text{adv}}^T \mathbf{s}_i \geqslant 0$, and expanding if $\mathbf{d}_{\text{adv}}^T \mathbf{s}_i < 0$.

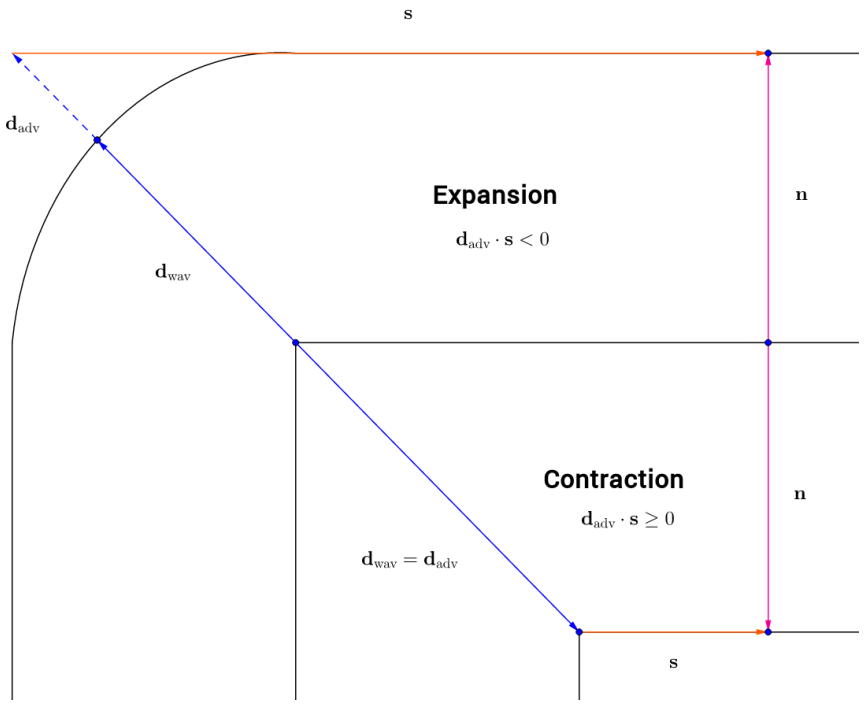

Figure 2.2: Expanding and contracting

X.Jiao suggested the following simple formula obtaining length l incorporating both motions

$$l = \left(\sum_{i=1}^{m} \mu_i l_i \right) / \left(\sum_{i=1}^{m} \mu_i \right),$$

where

$$\mu_i = \begin{cases} a_i & \text{if expanding} \\ a_i \cos^2(\theta_i) & \text{if contracting} \end{cases}, \; l_i = \begin{cases} c_i & \text{if expanding} \\ c_i / |\cos(\theta_i)| & \text{if contracting} \end{cases}.$$

Here, a_i is the area of the i-th face, c_i is the length of i-th face moving under normal velocity, and θ_i is the angle between $\mathbf{d}_{\text{adv}}$ and $\mathbf{n}_i$. The final vertex displacement $\mathbf{d}_{\text{wav}}$ can be calculated as follows.

$$\mathbf{d}_{\text{wav}} = l \frac{\mathbf{d}_{\text{adv}}}{|\mathbf{d}_{\text{adv}}|}.$$

2.2.3 Null-space smoothing

Vertex redistribution or smoothing is used to maintain mesh quality during propagation. Note that this redistribution procedure should be done on the complementary space of the primary space(that is, smoothing movement of vertex should be tangential to moving direction), which we call null space. Therefore, vertex movement is a sum of vertex displacement $\mathbf{d}$ (which can be either $\mathbf{d}_{\text{adv}}$ or $\mathbf{d}_{\text{wav}}$), and tangential displacement $\mathbf{t}$, i.e., $\mathbf{d} + \mathbf{t}$. Here, $\mathbf{t}$ is given by

$$\mathbf{t} = \mathbf{T}\mathbf{T}^T \left(\sum_{i=1}^{m} w_i \mathbf{c}_i \right) / \left(\sum_{i=1}^{m} w_i \right),$$

where $\mathbf{T}$ is a $3 \times (3-k)$ matrix whose column vectors consists of orthonormal bases of the null space of $\mathbf{A}$, $\mathbf{c}_i$ is a vector connecting the vertices from the vertex to the center of the i-th neighboring face, and w_i is the weight for i-th face. Here, one can use same weights as in determining local geometry previously.

Note that the linear map $\mathbf{T}\mathbf{T}^T$ projects each vector in $\mathbb{R}^3$ to a null space of $\mathbf{A}$. Therefore, this method can be seen as a projection of the weighted Laplacian smoothing to the null space. In the smooth region, the two vectors which spans the null space are also a basis of the tangent plane at the vertex and thus the smoothing process is done in the tangent plane. Likewise, the vertex on ridges can be only smoothed along the direction of ridges, and smoothing does not occur at the corner vertex. With this simple smoothing scheme, we can prevent sharp features from being corrupted during the smoothing process. We call such smoothing procedure null-space smoothing.

2.3 Weighted essentially non-oscillatory scheme

In this section we are going to introduce a weighted essentially non-oscillatory scheme(WENO), a high order reconstruction scheme that is essential for developing FOM-WENO scheme. In particular, we focus on the WENO scheme in one dimensional space, which is a building block for developing two dimensional FOM-WENO scheme. For one dimensional mesh, we can arrange cells and points in order. We conveniently arrange the cells in order of $I_1, I_2, \cdots I_N$. The point shared by I_i and I_{i+1} is denoted by $x_{i+1/2}$.

2.3.1 Polynomial reconstruction

The most important component for the WENO scheme is how to define a stencil containing each cell I_i, and how to interpolate the value of a given point using averages of cell which is in the same stencil with a given point. Here, the cell average for a function v in j'th cell I_j is given by

$$\bar{v}_j \equiv \frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(x)dx, \quad j = 1, 2, \cdots, N.$$

Now, let us have all cell averages. For a given cell I_i, we consider the interpolation problem for v at $x_{i+1/2}$(which we simply write $v_{i+1/2}$) using cell averages nearby I_i. The most simple, and natural first order approximation is $\bar{v}_i$. For high order reconstruction, we need to make use of cell averages in a neighborhood of I_i as well. Thus, we consider the set S containing I_i as well as the neighbor cells.

$$S = \{I_{i-r}, I_{i-r+1}, \cdots, I_i, \cdots, I_{i+s}\}, \quad r, s \geqslant 0.$$

We call this S the stencil of I_i. For $k = r + s + 1$, there is an unique polynomial of order $(k-1)$ such that the cell averages of $p(x)$ in the stencil are equal to those of v.

$$\frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p(x)dx = \bar{v}_j, \quad j = i - r, \cdots, i + s.$$

If a function v is smooth in the stencil, then $p(x)$ is an kth order approximating polynomial for v in the stencil. In particular, $p(x_{i+1/2})$ is a kth order approximation for $v_{i+1/2}$. Now, for a fixed k, we denote this ap-

proximation as $v^{(r)}_{i+1/2}$. This value can be easily shown to be linear with respect to cell averages $\bar{v}_i$ in the stencil. Therefore, one can define constants $c_{rj}, \quad j = 0, \cdots k-1$ which are independent of a function v to find the following expression for $v^{(r)}_{i+1/2}$.

$$v^{(r)}_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj}\bar{v}_{i-r+j}. \tag{2.3.1}$$

More precisely, using k and $\Delta x_i = x_{i+1/2} - x_{i-1/2}$, one can derive the following expression for constants c_{rj}. Detailed calculations are omitted here.

$$c_{rj} = \Delta x_{i-r+j} \sum_{m=j+1}^{k} \frac{\sum_{l=0,l\neq m}^{k} \prod_{q=0,q\neq m,l}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{l=0,l\neq m}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)}. \tag{2.3.2}$$

Note that one can also handle the interpolation for $v_{i-1/2}$ in the same manner.

2.3.2 ENO reconstruction

Before introducing the WENO scheme, we breifly introduce an essentially non-oscillatory scheme(ENO) [22], which has become the motivation of the WENO scheme. We defined a stencil at cell I_i consisting of k cells containing I_i in the previous section, and if a function v is smooth in this stencil, then the value $v_{i+1/2}$ can be approximated to kth order using cell average values of cells belonging to the stencil. Let us now consider the case where v is piecewise smooth and there is a discontinuity(shock) for v near the cell I_i. In this case, if discontinuity exists in the stencil, the order of accracy for interpolation is no longer guaranteed. In other words, the interpolating value for $v_{i+1/2}$ can be overshooted or undershooted, which results in an oscillation called Gibbs Phenomena. To overcome this, we consider the following k different stencil.

$$S_r = \{I_{i-r}, I_{i-r+1}, \cdots, I_i, \cdots, I_{i-r+k-1}\}, \quad r = 0, \cdots, k-1.$$

The core idea for ENO scheme is to use the smoothest stencil to avoid discontinuity. It follows that it is essential for us to have a suitable measure of smoothness. In the conventional ENO scheme, one usually consider the

primitive function $P_r(x)$ of the interpolating polynomial $p_r(x)$ computed from each S_r. Polynomial $P_r(x)$ interpolates the following function $V(x)$

$$V(x) \equiv \int_{-\infty}^{x} v(x')dx',$$

at $k+1$ points, $x_{i-r-1/2}, \cdots, x_{i-r+k-1/2}$. Now, the Newton divided difference of V is used as the measurement of the smoothness in the stencil. Starting from 1-cell stencil I_i, by repeatedly adding left or right cell which has smaller divided differences, we can construct the stencil consisting of k cells. After determining stencil for I_i, the equation (2.3.1) can be used to approximate $v_{i+1/2}$.

There are several problems with the ENO scheme. One of them is that although $(2k-1)$ cells in a neighborhood of I_i are likely to be used in selecting stencil of I_i, only k of them are used and the order of accuracy is also limited to k. Another problem is that the approximated value for $v_{i+1/2}$ is not smooth with repsect to cell average values, since the chosen stencil changes abruptly. Various methodologies have been proposed to overcome these problems, the most famous of which is the WENO scheme.

2.3.3 WENO reconstruction

Weighted ENO(WENO) is a method intrdouced by Liu, Osher, and Chan [21], where several problems of ENO scheme discussed above are overcomed by using a weighted sum wisely. Consider k different stencils S_r discussed above. The core idea of WENO is that the approximate value for $v_{i+1/2}$ is expressed by a weighted sum of approximate value $v_{i+1/2}^{(r)}$ which is computed from each stencil S_r.

$$v_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r v_{i+\frac{1}{2}}^{(r)}, \quad \omega_r \geqslant 0, \quad \sum_{r=0}^{k-1} \omega_r = 1. \tag{2.3.3}$$

Two conditions in equation (2.3.3) are necessary to satisfy stability and consistency for WENO scheme. The ENO scheme discussed above can be seen as a special case of WENO scheme, in that the weight ω_r is 1 in the smoothest stencil and 0 in the remaining stencil. Now we want to design weights ω_r to satisfy the following three conditions.

1. In smooth region, resulting order of accuracy for $v_{i+1/2}$ is $(2k-1)$,

2. For a stencil containing a shock, ω_r is nearly zero,

3. ω_r is a smooth function for the cell average value.

If we rewrite the first condition, we have to find constants d_r so that the following holds.

$$\sum_{r=0}^{k-1} d_r v_{i+\frac{1}{2}}^{(r)} = v_{i+\frac{1}{2}} + O(\Delta x^{2k-1}). \tag{2.3.4}$$

Furthermore, we do not want d_r to depend on the cell average values. We first find the expression which approximates $v_{i+\frac{1}{2}}$ by $(2k-1)$-th order using $(2k-1)$ cell averages.

$$v_{i+\frac{1}{2}} = \sum_{j=0}^{2k-2} \widetilde{c}_j \bar{v}_{i-k+1+j} + O(\Delta x^{2k-1}). \tag{2.3.5}$$

Then, we equate the expression (2.3.5) with the weighted sum of $v_{i+\frac{1}{2}}^{(r)}$ to find constants d_r. In other words, the following equation should be satisfied.

$$\sum_{j=0}^{2k-2} \widetilde{c}_j \bar{v}_{i-k+1+j} = \sum_{r=0}^{k-1} d_r v_{i+\frac{1}{2}}^{(r)} = \sum_{r=0}^{k-1} d_r \left(\sum_{j=0}^{k-1} c_{rj} \bar{v}_{i-r+j} \right). \tag{2.3.6}$$

We want this equation holds whatever the cell average value is, and this yields the linear equation. At first glance, it seems that the system is overdetermined since the number of d_r sought is k, whereas the number of equation equals $(2k-1)$, which is the number of cell averages. However, the condition $\sum_{r=0}^{k-1} d_r = 1$ enforces the equation hold for $v = 1, x, \cdots, x^{k-1}$, since both part of equation (2.3.6) correctly reconstruct all for polynomials of order less than k. Therefore, actual number of the equation reduces by $(k-1)$, then we have a linear system of k equations with k unknowns. d_r can be uniquely found by solving this linear system.

Now, to satisfy the second and third conditions, we use the following form of weights :

$$\omega_r = \frac{\widetilde{\omega}_r}{\sum_{s=0}^{k-1} \widetilde{\omega}_s}, \quad r = 0, \cdots, k-1, \tag{2.3.7}$$

where

$$\widetilde{\omega}_r = \frac{d_r}{(\epsilon + \beta_r)^2}. \tag{2.3.8}$$

β_r is a measure of the smoothness of stencil S_r, called smooth indicators, and $\epsilon > 0$ is a small constant to prevent the denominator from reaching 0. Using the designed weights, the smooth indicators are nearly zero in the smooth region, and this implies $\omega_r \approx d_r$. That is, from the equation (2.3.4), we can obtain $(2k-1)$ order approximation in the smooth region. If one has discontinuity in stencil r', then $\beta_{r'}$ is very large, which in turn implies $\alpha_{r'}$ is very small compared to other stencils. Thus, $\omega_{r'}$ is nearly zero for the stencil which has a discontinuity. Therefore, you can avoid using stencil with discontinuity as in ENO scheme. Finally, we can see that ω_r is smooth provided β_r is smooth with repsect to cell average values.

We used the Newton divided differences to measure the smoothness of the stencil. The following smooth indicators are generalized versions of total variations, which are known to be good measures of smoothness, to higher order variations. For the 3rd order and 5th order WENO schemes, the robustness of this smooth indicator has been verified through several numerical experiments.

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x_i^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial^l x} \right)^2 dx. \tag{2.3.9}$$

We can calculate ω_r using this smooth indicator, and finally the WENO reconstruction can be done as follows.

$$\begin{aligned} v_{i+\frac{1}{2}}^{(r)} &= \sum_{j=0}^{k-1} c_{rj} v_{i-r+j}, \quad j = 0. \cdots, k-1, \\ v_{i+\frac{1}{2}} &\approx \sum_{r=0}^{k-1} \omega_r v_{i+\frac{1}{2}}^{(r)}. \end{aligned}$$

Note that the value of $v_{i+1/2}$ can also be approximated in I_{i+1}. We usually denote the approximated value for $v_{i+1/2}$ in I_i by $v_{i+1/2}^-$, and the approximated value for $v_{i+1/2}$ in I_{i+1} by $v_{i+1/2}^+$.

Generally, the fifth order WENO method, which corresponds to $k = 3$, is widely used in many papers. In this case, the total number of cell used is 5, and each stencil S_r has 3 cells. That is, in the smooth region, the accuracy order is close to 5, and the 3rd order accuracy can be obtained near the shock.

2.4 WENO scheme on triangular meshes

In the previous section, we designed WENO scheme in 1D. In this section we extend WENO scheme into 2D space, consisting of triangular mesh. This process is introduced in [23]. First, let us assume that a smooth function u is defined on given triangular mesh, whose triangles are denoted by $\{\triangle_0, \triangle_1, \cdots, \triangle_N\}$. For WENO reconstruction, we should interpolate the values at given points from the cell averages. The average value of $u(x, y)$ in each cell $\triangle_i$ is given by

$$\bar{u}_i \equiv \frac{1}{|\triangle_i|} \int_{\triangle_i} u(x, y)\, dx\, dy.$$

Let P^k be a set of two-variable polynomials whose order is less than k. Then our aim is to find the reconstruction polynomial $p(x, y)$ in P^k which satisfies following conditions: p approximates u in $\triangle_i$ with order $(k + 1)$, and its cell average value in $\triangle_i$ is same as that of u. Since there are $K = (k + 2)(k + 1)/2$ coefficients in $p(x, y)$, we need at least K triangles to determine these coefficients. We use set of K triangles $S_i = \{\Omega_1, \Omega_2, \cdots, \Omega_K\}$ to find the polynomial in $\triangle_i$, and S_i is usually called a stencil of $\triangle_i$. For example, when $k = 1$, we usually use stencils made of $\triangle_i$ and two neighbors around it.

Furthermore, $p(x, y)$ should be reconstructed so that the cell average of p in Ω_i is the same as that of u, which yields K linear equations. Since $p(x, y)$ has K coefficients, we obtain a linear system $Ax = b$ for $K \times K$ matrix A. When matrix A has full rank, i.e. when the linear equation always has a unique solution, we call S_i the admissible stencil of $\triangle_i$. In previous case $k = 1$, stencils are admissible in most triangulations.

2.4.1 Third order reconstruction

We start with the 3rd order WENO scheme. The first step required to create WENO scheme is to find a reconstruction polynomial p. In 3rd order case, we need a quadratic polynomial interpolation and a proper stencil for each triangle. Let $\triangle_0$ be the cell that we want to approximate. Three neighbors of $\triangle_0$, are denoted by $\triangle_i$, $\triangle_j$, and $\triangle_k$. Again, we need $\triangle_{ia}$, $\triangle_{ib}$, which are two neighbors of $\triangle_i$ different from $\triangle_0$. In the same way, we can also find $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_{ka}$, $\triangle_{kb}$. In this section, we use this 10-triangle stencil for reconstruction.

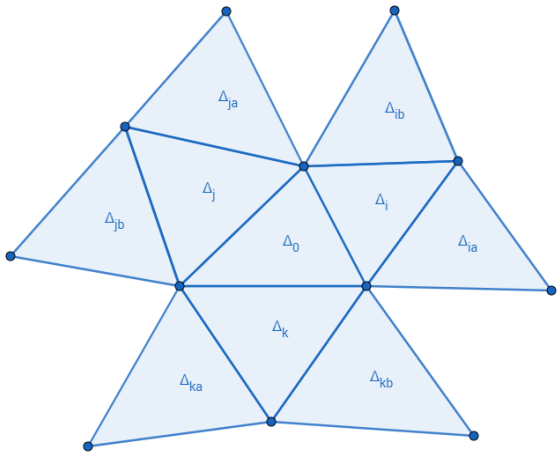

Figure 2.3: Whole stencil used in third, fourth order WENO scheme

Now, we should find a quadratic polynomial $p^{(2)}$, which has the same cell average as that of u in $\triangle_0$. In addition, we also require that cell averages of $p^{(2)}$ and u are equal at all triangles in the stencil. As there are only 6 undetermined coefficients in polynomial $p^{(2)}$, the system for determining coefficient is actually overdetermined. Therefore, in general, we use the least-square method to find these coefficients.

The most important step in developing a high-order WENO scheme is to represent the reconstruction polynomial using lower order polynomials(that is, with smaller stencils). For the 3rd order case, we usually use polynomials of order 1 for this. Since polynomial of order 1 has three coefficients, it is necessary to decompose the stencil into smaller stencils, each of which is made up of three triangles.

Using the 10-triangle stencil defined above, 9 smaller stencils can be made. The following notation is used for 9 substencils :

S_1	$\triangle_0$, $\triangle_j$, $\triangle_k$	S_2	$\triangle_0$, $\triangle_k$, $\triangle_i$	S_3	$\triangle_0$, $\triangle_i$, $\triangle_j$
S_4	$\triangle_0$, $\triangle_i$, $\triangle_{ia}$	S_5	$\triangle_0$, $\triangle_i$, $\triangle_{ib}$	S_6	$\triangle_0$, $\triangle_j$, $\triangle_{ja}$
S_7	$\triangle_0$, $\triangle_j$, $\triangle_{jb}$	S_8	$\triangle_0$, $\triangle_k$, $\triangle_{ka}$	S_9	$\triangle_0$, $\triangle_k$, $\triangle_{kb}$

Table 2.2: Substencils used in the third order WENO scheme

Then, the polynomial p_r of order 1 is reconstructed for each S_r, where

the cell average of p_i in each triangle consisting of S_r is equal to that of u.

For the point (x^G, y^G) we want to approximate, we should find the coefficient γ_r which only depends on local geometry of mesh, so that the following linear combination

$$R(x,y) = \sum_{r=1}^{9} \gamma_r p_r(x,y) \tag{2.4.1}$$

matches $p^{(2)}$ at the point (x^G, y^G) where we want to approximate values, whatever the values $\{\bar{u}_0, \bar{u}_i, \bar{u}_j, \bar{u}_k, \bar{u}_{ia}, \bar{u}_{ib}, \bar{u}_{ja}, \bar{u}_{jb}, \bar{u}_{ka}, \bar{u}_{kb}\}$ are.

$$R(x^G, y^G) = p^{(2)}(x^G, y^G). \tag{2.4.2}$$

In equation (2.4.1) and (2.4.2), all expressions are linear with respect to cell averages. In order to achieve equality for arbitrarily given cell averages, ten linear coefficients should be identically zero when we organize the expression with cell averages as variables. Since there are nine γ_r's to be determined in the equation, one can misunderstand at glance that the equation is overdetermined, but in fact the system has rank 8 so the equation is underdetermined, having one degree of freedom. This is due to the fact that since both $p_r(x)$ and $p^{(2)}(x)$ can reconstruct the first order polynomial exactly, the equality for three cases $u = 1, x, y$ are automatically established with the one constraint $\sum_{r=1}^{9} \gamma_r = 1$. Therefore, we have rank 8 system with 9 variables, resulting in one degree of freedom. For convenience, we express γ_r as a function of γ_1 for $r \geqslant 2$.

Remaining one degree of freedom will be used later to make each linear weights γ_r nonnegative, which is an essential condition when dealing with shocks. Note that such nonnegativity of weights is not possible for many triangulations. One way to overcome this problem is a grouping technique, which will be introduced in next subsection.

2.4.2 Fourth order reconstruction

For the 4th order WENO scheme, a cubic reconstruction polynomial with 10 coefficients is necessary. Thus $\triangle_{ia}$, $\triangle_{ib}$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_{ka}$, $\triangle_{kb}$ should be all distinct. Otherwise, our stencil cannot be admissible. Now, a cubic polynomial $p^{(3)}$ should be reconstructed in a way that cell averages of $p^{(3)}$ in each triangle are same as that of u. In most triangulations, this is possible when

$\triangle_{ia}$, $\triangle_{ib}$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_{ka}$, $\triangle_{kb}$ are all distinct.

As in 3rd order case, we now represent $p^{(3)}$ using lower order polynomials in some substencils. For example, we can create six quadratic polynomials q_r, correspond to each substencil S_r consisting of six triangles.

S_1	$\triangle_0$, $\triangle_i$, $\triangle_{ia}$, $\triangle_{ib}$, $\triangle_k$, $\triangle_{kb}$
S_2	$\triangle_0$, $\triangle_i$, $\triangle_{ia}$, $\triangle_{ib}$, $\triangle_j$, $\triangle_{ja}$
S_3	$\triangle_0$, $\triangle_j$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_i$, $\triangle_{ib}$
S_4	$\triangle_0$, $\triangle_j$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_k$, $\triangle_{ka}$
S_5	$\triangle_0$, $\triangle_k$, $\triangle_{ka}$, $\triangle_{kb}$, $\triangle_j$, $\triangle_{jb}$
S_6	$\triangle_0$, $\triangle_k$, $\triangle_{ka}$, $\triangle_{kb}$, $\triangle_i$, $\triangle_{ia}$

Table 2.3: Substencils used in the fourth order WENO scheme

Again, we wish that following linear combination

$$Q(x,y) = \sum_{r=1}^{6} \gamma_r q_r(x,y) \tag{2.4.3}$$

matches $p^{(3)}$ at the point (x^G, y^G), whatever the values of $\bar{u}$'s are.

$$Q(x^G, y^G) = p^{(3)}(x^G, y^G).$$

In this case, the number of γ_r is 6 and the number of linear equations is 10. However, the single constraint $\sum_{r=1}^{6} \gamma_r = 1$ can incorporate six equations corresponding to $u = 1, x, y, x^2, xy, y^2$ into single equation. Therefore, the linear system is actually a rank 5 system and again has one degree of freedom. One degree of freedom can be used later to make the weights non-negative, but different from the 3rd order case, this is only possible on nearly uniform meshes.

2.4.3 Positivity of linear weights

In previous subsections, one degree of freedom has left when calculating linear weights and we expressed $\gamma_r (r \geqslant 2)$ using γ_1. In order for the final WENO reconstruction to be stable near shock, each linear weight should be non-negative. Therefore, it is necessary to adjust the value of γ_1 to make each linear weight non-negative. However, this adjustment is generally not possible for most triangulations. To overcome this problem, one can seek

grouping techniques. For example, consider the case when you want to incorporate the nine polynomials used in third order reconstruction into three new polynomials and coefficients as follows:

$$\sum_{r=1}^{9} \gamma_r p_r(x,y) = \sum_{r=1}^{3} \widetilde{\gamma}_r \widetilde{p}_r(x,y). \tag{2.4.4}$$

In equation (2.4.4), each $\widetilde{p}_r(x,y)$ is a linear combination of the three linear polynomials in LHS. Note that it is also a linear polynomial which approximates u with second order accuracy. The stencil for $\widetilde{p}_r(x,y)$ is just the union of three corresponding substencils. Here, one should ensure that each new stencil is well separated, so that when the shock is present, at least one stencil do not contain a shock.

The following grouping works for most triangulations. The first group uses p_2, p_4, and p_5, the second group contains p_3, p_6, and p_7, and finally the last group consists of the rest, p_1, p_8, and p_9. Following the grouping mentioned above, three new linear polynomials, their coefficients and corresponding stencils are as follows.

$$\begin{aligned}
\widetilde{p}_1 &= (\gamma_2 p_2 + \gamma_4 p_4 + \gamma_5 p_5)/(\gamma_2 + \gamma_4 + \gamma_5), \quad \widetilde{\gamma}_1 = \gamma_2 + \gamma_4 + \gamma_5, \\
\widetilde{p}_2 &= (\gamma_3 p_3 + \gamma_6 p_6 + \gamma_7 p_7)/(\gamma_3 + \gamma_6 + \gamma_7), \quad \widetilde{\gamma}_2 = \gamma_3 + \gamma_6 + \gamma_7, \\
\widetilde{p}_3 &= (\gamma_1 p_1 + \gamma_8 p_8 + \gamma_9 p_9)/(\gamma_1 + \gamma_8 + \gamma_9), \quad \widetilde{\gamma}_3 = \gamma_1 + \gamma_8 + \gamma_9.
\end{aligned}$$

$\widetilde{S}_1$	$\triangle_0$, $\triangle_i$, $\triangle_{ia}$, $\triangle_{ib}$, $\triangle_k$
$\widetilde{S}_2$	$\triangle_0$, $\triangle_j$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_i$
$\widetilde{S}_3$	$\triangle_0$, $\triangle_k$, $\triangle_{ka}$, $\triangle_{kb}$, $\triangle_j$

Table 2.4: New stencils corresponding to grouped polynomials

The resulting linear combination

$$\widetilde{R}(x,y) = \sum_{r=1}^{3} \widetilde{\gamma}_r \widetilde{p}_r(x,y)$$

is equivalent to the previous $R(x,y)$, and in most cases it is possible to make all $\widetilde{\gamma}_r$'s non-negative by adjusting one degree of freedom. Stencils are

relatively well separated too.

In the case of the 4th order, there is also given one degree of freedom to determine the value of linear weights γ_r. Therefore, making the six linear weights non-negative using this degree of freedom is then equivalent to solving six linear inequalities for γ_1. But experimentally this is only possible if the mesh is nearly uniform. Since the size of each stencil is relatively large compared to 3rd order case, the shock is more likely to enter all the stencils formed by grouping. Thus, the problem is rarely overcomed by using simple grouping techniques.

2.4.4 Smoothness indicators and nonlinear weights

Finally, the smooth indicator is used to calculate the nonlinear weights. As in 1D case, for the polynomial $p(x, y)$ whose order is less than k, we define the measure for smoothness as follows.

$$\beta = \sum_{1 \leqslant |\alpha| \leqslant k} \int_{\triangle} |\triangle|^{|\alpha|-1} (D^{\alpha} p(x, y))^2 \, dx \, dy.$$

Here, the summation is taken for α, a multi-index notation for differentiation. Using smooth indicator β, nonlinear weights ω_r can be defined as follows.

$$\omega_r = \frac{\widetilde{\omega}_r}{\sum_s \widetilde{\omega}_s}, \quad \widetilde{\omega}_r = \frac{\gamma_r}{(\epsilon + \beta_r)^2}. \tag{2.4.5}$$

In equation (2.4.5), γ_r is the linear weights calculated before (especially, for 3rd order case, we may use $\widetilde{\gamma}_r$ instead of γ_r when grouping technique is applied.), β_r is the smooth indicator of r'th reconstruction polynomial (p_r or $\widetilde{p}_r$ in 3rd order case, and q_r in the 4th order case), and ϵ is a small positive number introduced to deal with smooth region. In general, if ϵ is large, we can get better accuracy in the smooth region, but there may be some oscillation near shock. The smaller the ϵ, the better the shock can be handled. The WENO reconstruction is completed by replacing the previously calculated nonlinear weights γ_r with ω_r in (2.4.1) or (2.4.3) according to the order of approximation.

2.5 Total variation diminishing Runge-Kutta method

The purpose of time-stepping methods is to update u from time to time for given ODE

$$u_t = f(u), \quad u(t_0) = u_0,$$

and the spatial discretization of ODE

$$u_t = L(u).$$

The simplest method is the forward Euler method, which is based on Taylor expansion $u(t_0 + h) \approx u(t_0) + hu'(t_0)$.

$$u^{n+1} = u^n + \Delta t L(u^n).$$

However, since time discretization with the forward Euler method is only first order accurate in time variable, high order discretization is necessary for higher accuracy. The most commonly used method is the total variation diminishing (TVD) Runge-Kutta (RK) method devised by Shu and Osher [29]. This method is called TVD, because of the following property : If some spatial discretizations with forward Euler time update is TVD, then it is still TVD when using TVD-RK method as time discretization. That is, the total variation

$$TV(u) = \sum_{j \in \mathbb{Z}} |u_{j+1} - u_j|$$

does not increase with time:

$$TV(u^{n+1}) \leqslant TV(u^n).$$

This property makes the scheme stable. The following is the commonly used third order TVD-RK method.

$$
\begin{aligned}
u^{(1)} &= u^n + \Delta t L(u^n), \\
u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^1), \\
u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^2).
\end{aligned}
$$

Chapter 3

Proposed models

3.1 FOM-WENO scheme

3.1.1 Motivations for high order FOM scheme

There are many numerical schemes for computing numerical fluxes. FDM(Finite Difference Method) and FVM(Finite Volume Method) are two widely used methodologies. Although these schemes work well for globally smooth problems, due to their interpolation is based on fixed stencil, the high order accurate result become oscillatory near a discontinuity. As we pointed out in section 2.3, WENO scheme makes use of adaptive stencil to avoid including the discontinous cell in the stencil when computing numerical flux. Therefore, WENO scheme maintain high order accurate results in smooth region as well as reduce artificial oscillations near a discontinuity.

We explained brief concept of FOM scheme in section 2.2. As one can see easily, the biggest shortcoming for conventional FOM scheme is that the accuracy order is restricted to 1st order even in smooth region of the interface. This is due to the fact that FOM scheme uses only neighboring face normals to determine offset directions. However, it is clear that in smooth region of the front one can widen stencil to obtain high order accuracy. This is where our idea originates from. We utilize such advantages of WENO schemes when determining normal vectors(in many practical examples, this is equivalent to determine propagating direction of front) in FOM scheme to guarantee high order accurate in smooth regions of front.

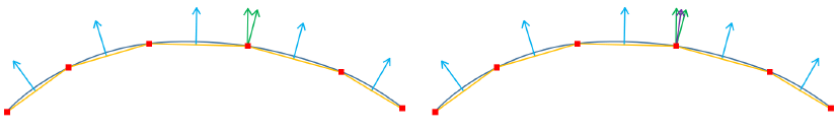

Figure 3.1: Conventional FOM scheme only uses neighboring normals(blue arrow) to approximate the normal at the vertex of front(purple arrow)

Furthermore, there is more room for improvement in FOM scheme. FOM scheme uses two geometric parameters χ_c, χ_r. These parameters are used as a threshold when determining the local shape of the interface. Therefore, in FOM scheme, the local geometry of a vertex can be changed according to the value of χ_c, χ_r, and then errors can be induced when determining correct propagating directions. We want the local shape to be well determined even if the values of χ_c, χ_r are fixed for various shapes, so that the interface moves correctly.

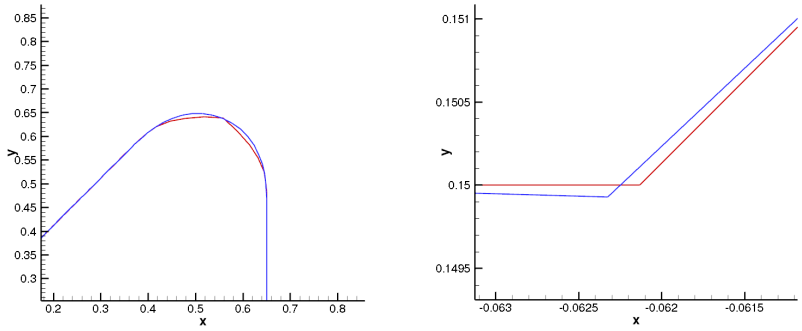

Figure 3.2: Small χ_r may result in sharp shape at expanding corner which should be rounded (left figure), and large χ_r may misdirect contracting corner (right figure). In both figures, blue line corresponds to correct interface and red line corresponds to the interface propagated by using FOM scheme.

For example, for the interface lying in two spatial dimension(only χ_r exists in this case), if χ_r is set to be too small, FOM scheme can hardly smoothen expanding corner(See the left figure of figure 3.2). Similarly, if χ_r is set to be too large, the method cannot find out corners with small angle(See the right figure of figure 3.2). We would like to develop new method easier to determine the value of χ_r, χ_c than the existing FOM scheme.

3.1.2 High order reconstruction of normals

The starting point for implementing FOM-WENO scheme in two dimensions is to approximate the normal vector with high order accuracy using polynomial reconstruction. For better understanding of the situation, we assume that we arrange the faces of the interface in order, and that the point shared by face i and face $(i+1)$ is denoted by the point $(i+1/2)$. Then the normal vectors at $(i+1/2)$ can be approximated using normal informations in a neighborhood of $(i+1/2)$. Let us denote $N^-_{i+1/2}$ and $N^+_{i+1/2}$ by normal approximations for $(i+1/2)$ at face i and $i+1$, respectively. FOM scheme can be viewed as a scheme for calculating the offset direction using these two normal approximations. This similar to the role of the numerical flux calculations in computatinoal fluid dynamics(CFD). Conventional FOM scheme uses the following simple approximations, if we denote normal vectors of face i and face $i+1$ by N_i, N_{i+1}, respectively.

$$N^-_{i+1/2} = N_i, \quad N^+_{i+1/2} = N_{i+1}.$$

Using the numerical flux approximation ideas in CFD fields, it is possible to improve the accuracy of conventional FOM scheme by approximating $N^-_{i+1/2}$ and $N^+_{i+1/2}$ with high order numerical flux approximation scheme. To achieve this, we use 5th order WENO scheme discussed in section 2.3. We first consider only linear coefficients that do not take smooth indicators into account. Then a new reconsturction of the normal vector at point $(i+1/2)$ calculated from face i can be wriiten as

$$\begin{aligned} N^{(r)}_{i+\frac{1}{2}} &= \sum_{j=0}^{k-1} c_{rj} N_{i-r+j}, j = 0. \cdots, k-1, \\ N^-_{i+\frac{1}{2}} &= \sum_{r=0}^{k-1} d_r N^{(r)}_{i+\frac{1}{2}}, \end{aligned}$$

where $N^+_{i+\frac{1}{2}}$, which is computed from face $(i+1)$, can be computed in a similar fashion. Since the faces are not uniform, the linear coefficients c_{ij} and d_i should be calculated as unstructured sense, namely using the equation (2.3.2). For explicit formula of d_i, see [33]. The offset direction is calculated from the obtained $N^-_{i+1/2}$ and $N^+_{i+1/2}$ through the conventional FOM scheme. One can propagate the interface by repeating this process at every point on the interface.

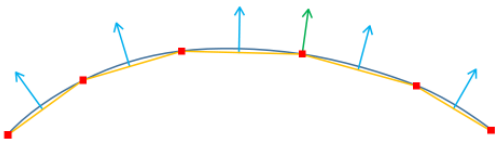

Figure 3.3: FOM-WENO uses normal vector of wider stencils to get high order approximation of normals(green arrow).

When we apply the 5th order WENO scheme, we consider a stencil consisting of 5 faces divided by 3 substencils consisting of 3 faces. However, in the case of interface tracking problem, each face in the stencil is not colinear, thus standard WENO scheme is not directly applicable. There should be some sort of face unfolding process so that each face in the stencil to be colinear(coplanar in 3D). The unfolding process is described in the following figure 3.4.

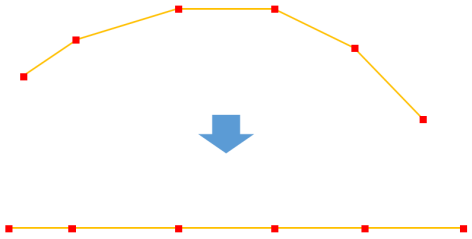

Figure 3.4: Illustration of unfolding process

The main advantage of this unfolding process is that it doesn't matter whether corner exists in the stencil or not. This is due to the fact that the unfolding process is basically reparametrization of curve. Furthermore, this unfolding process is always possible for interfaces with sufficently good mesh qualites. Since the mesh size distribution of the interface is balanced, the face area in the stencil becomes also balanced, therefore the condition number of linear system for linear coefficients is not relatively large.

3.1.3 Modified normal vector and error analysis

As pointed out in subsection 3.1.1, FOM-WENO scheme is designed to achieve high order accuracy of normal vectors in smooth region of the interface by making use of WENO scheme. In this section, we theoretically verify such high order accuracy for normal vector is actually obtained in smooth region of interface for two dimensional case.

Let $X(t)$ be an initial interface, which is a curve parametrized by arc length in xy-plane. All we know about initial interface is the set of $n+1$ discrete points $X_{1/2}, X_{3/2}, .., X_{n+1/2}$ on the curve, where $X_{i+1/2}$ denotes $X(t_{i+1/2})$. We note that the following identity for curve tangents give the similarity between approximating normals and WENO schemes, which also makes use of cell averages to compute numerical fluxes.

$$\frac{1}{t_{i+1/2} - t_{i-1/2}} \int_{t_{i-1/2}}^{t_{i+1/2}} X'(t)dt = \frac{X_{i+1/2} - X_{i-1/2}}{t_{i+1/2} - t_{i-1/2}}.$$

Since normal vectors can be obtained by rotating anti-clockwise the unit tangent vector $X'(t)$ by 90°, if we denote R to be a 90° anti-clockwise rotation matrix, we apply ENO/WENO schemes with the cell average $\bar{v}_i = R\dfrac{X_{i+1/2} - X_{i-1/2}}{t_{i+1/2} - t_{i-1/2}}$. From the known accuracy theory for ENO/WENO schemes, we have the following approximation for $RX'(t)$

$$RX'(t) = \sum_{i=1}^{n} c_i \bar{v}_i + O(\Delta t^n),$$

where c_i are nth order ENO/WENO coefficients and

$$\Delta t = \max_{1 \leqslant i \leqslant n} (t_{i+1/2} - t_{i-1/2}).$$

However, since we approximate the interface by union of lines, we don't have exact information of arc length parameter t. Therefore we should replace the mathematical quantity $R\dfrac{X_{i+1/2} - X_{i-1/2}}{t_{i+1/2} - t_{i-1/2}}$ by the computable line normals $N_i = \dfrac{X_{i+1/2} - X_{i-1/2}}{|X_{i+1/2} - X_{i-1/2}|}$. Then, one cannot expect how much the or-

der of normal approximation degrades. Somehow, we seek maximum possible values of large m, satisfying the following equation.

$$RX'(t) = \sum_{i=1}^{n} c_i N_i + O(h^m),$$

where N_i is the normal vector of line connecting $X_{i-1/2}, X_{i+1/2}$, and

$$h = \max_{1 \leqslant i \leqslant n} |X_{i+1/2} - X_{i-1/2}|.$$

To see the feasibility of approximation using N_i's, we seek to estimate the error $\bar{v}_i - N_i$. For simplicity, let's introduce two temporary variables $h_i = t_{i+1/2} - t_{i-1/2}$, $\widetilde{h}_i = |X_{i+1/2} - X_{i-1/2}|$. Then from the Taylor expansion, we have

$$\begin{aligned}
\widetilde{h}_i^2 &= \left(h_i X'(t_{i-1/2}) + \frac{h_i^2}{2} X''(t_{i-1/2}) + \frac{h_i^3}{6} X'''(t_{i-1/2}) + O(h_i^4) \right) \times \\
&\quad \left(h_i X'(t_{i-1/2}) + \frac{h_i^2}{2} X''(t_{i-1/2}) + \frac{h_i^3}{6} X'''(t_{i-1/2}) + O(h_i^4) \right) \\
&= h_i^2 - \frac{h_i^4}{12} |X''(t_{i-1/2})|^2 + O(h_i^5).
\end{aligned}$$

During the above computation, we use the following two identities which can be obtained by taking derivatives in unit speed condition of $X(t)$.

$$X' \cdot X'' = 0, \quad |X''|^2 + X' \cdot X''' = 0.$$

Note that

$$\begin{aligned}
\widetilde{h}_i &= h_i \sqrt{1 - \frac{h_i^2}{12} |X''(t_{i+1/2})|^2 + O(h_i^3)} \\
&= h_i \left(1 - \frac{h_i^2}{24} |X''(t_{i+1/2})|^2) + O(h_i^4 \right) \\
&= O(h_i).
\end{aligned}$$

Now,

$$\begin{aligned}
\bar{v}_i - N_i &= R(X_{i+1/2} - X_{i-1/2})\left(\frac{1}{h_i} - \frac{1}{\widetilde{h}_i}\right) \\
&= R(X_{i+1/2} - X_{i-1/2})\left(\frac{\frac{-h_i^3}{24}|X''(t_{i-1/2})|^2 + O(h_i^4)}{h_i\widetilde{h}_i}\right) \\
&= R\left(\frac{X_{i+1/2} - X_{i-1/2}}{|X_{i+1/2} - X_{i-1/2}|}\right)\left(-\frac{h_i^2}{24}|X''(t_{i-1/2})|^2 + O(h_i^3)\right) \\
&= O(h_i^2).
\end{aligned}$$

Therefore, we can only guarantee

$$RX'(t) = \sum_{i=1}^{n} c_i N_i + O(h^2).$$

From the above calculation, however, we see that better accuracy can be achieved if we replace N_i by a suitable modified normal vector $\widetilde{N}_i$, which cancels $-\frac{h_i^3}{24}|X''(t_i)|^2$ term when computing $\bar{v}_i - \widetilde{N}_i$. That is, we only need to find $\widetilde{N}_i$ such that

$$\bar{v}_i - \widetilde{N}_i = R\frac{X_{i+1/2} - X_{i-1/2}}{|X_{i+1/2} - X_{i-1/2}|}O(h_i^3).$$

Simplest choice for $\widetilde{N}_i$ may be

$$\begin{aligned}
\widetilde{N}_i &= N_i - R\frac{X_{i+1/2} - X_{i-1/2}}{|X_{i+1/2} - X_{i-1/2}|}\left(\frac{h_i^2}{24}|X''(t_{i-1/2})|^2\right) \\
&= R\frac{X_{i+1/2} - X_{i-1/2}}{|X_{i+1/2} - X_{i-1/2}|}\left(1 - \frac{|X''(t_{i-1/2})|^2}{24}\right).
\end{aligned}$$

With the above modified normal vector $\widetilde{N}_i$, we finally get the following desired 3rd approximation $RX'(t)$.

$$RX'(t) = \sum_{i=1}^{n} c_i \widetilde{N}_i + O(h^3).$$

3.1.4 Setting functions reflecting geometric shocks

In the previous subsection, we proposed to use modified normal vectors $\widetilde{N}_i$ which approximate exact normal of the interface and proved that they are 3rd order approximations for exact normal vectors in smooth regions of the interface. Note that linear coefficients c_{rj} and d_r are independent of the cell averages. They only depend on numerical grid information. Now, to compute the WENO coefficients ω_r, we need to find the smooth indicator β_r. Without the smooth indicator, the normal direction is well approximated in the smooth region, but the normal vector dramatically changes near the corner(geometric shock), causing the high order interpolation to induce oscillation as shown in the following figure.

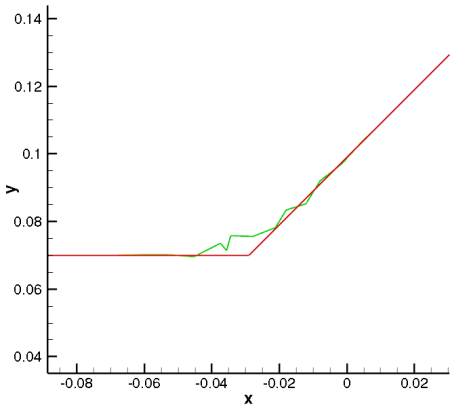

Figure 3.5: Oscillation may occur near corner or ridge, as each component of normal vectors may have shock.

From the equation (2.3.9), however, one can see that the smooth indicator requires a reconstruction polynomial $p_r(x)$ which does not depend solely on numerical grid information. In other words, we have to define appropriate values for each face to make the smooth indicator recognize the ridge or corner as a geometric shock. The natural phenomenon that occurs near

geometric shock is a dramatic change of face normals, therefore it is natural to constitute a function using neighbor face normals.

In practice, we apply WENO scheme to $\bar{v}_j = f(j;i) = N_{i+j} \cdot N_i$ $(j = -k+1, \cdots, k-1)$ to compute WENO coefficients at $x_{i+\frac{1}{2}}$. Here, N_p denotes the normal vector of face p. From the following figure 3.6, one can see that the function $f(j;i)$ has a shock at the position where there is a corner or ridge in the stencil as expected.

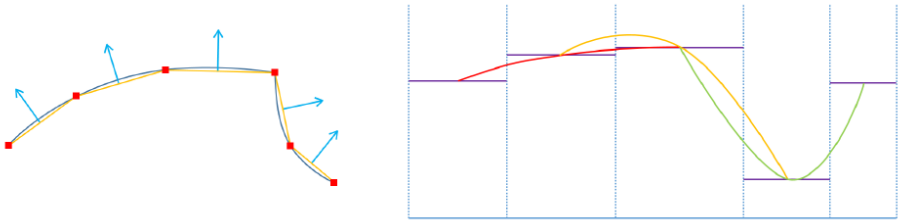

Figure 3.6: Left figure : local geometry of the interface. Exact interface is represented by a blue curve, and its linear approximation is presented with yellow line. Modified normal vectors at each line is given by sky blue arrows. Right figure : Cell averages of the function $\bar{v}_j = N_{i+j} \cdot N_i$(marked with purple line) and three possible quadratic polynomial interpolations(marked with red, yellow, green line) based on three cell averages in each stencil.

We can finally calculate WENO coefficient ω_r from equations (2.3.7) and (2.3.8). For example, when reconstructing a normal vector at the point $(i + 1/2)$, we can calculate a normal vector from the obtained WENO coefficient as follows.

$$\begin{aligned} \widetilde{N}^{(r)}_{i+\frac{1}{2}} &= \sum_{j=0}^{k-1} c_{rj} \widetilde{N}_{i-r+j}, j = 0. \cdots, k-1, \\ \widetilde{N}^{-}_{i+\frac{1}{2}} &= \sum_{r=0}^{k-1} \omega_r \widetilde{N}^{(r)}_{i+\frac{1}{2}}. \end{aligned}$$

Similarly, one can calculate $\widetilde{N}^{+}_{i+1/2}$ at face $(i+1)$, and now proceed to calculate the offset direction using $\widetilde{N}^{-}_{i+1/2}$ and $\widetilde{N}^{+}_{i+1/2}$ as in conventional face offsetting method.

3.1.5 Mesh smoothing method

While propagating, we need to smooth vertices after every one iteration. In this thesis, we utilized the null-space smoothing method introduced in section 2.2. The only difference is, the null space direction is also calculated from a normal vector approximated with high order accuracy. Nevertheless, with the simple null-space smoothing method, a smoothing error occurs, which seriously degrades the order as the interface moves.

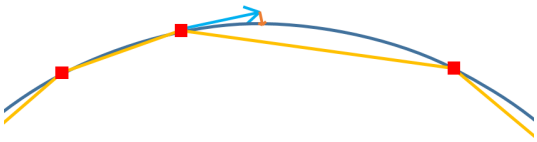

Figure 3.7: With a simple null-space smoothing, smoothing error can accumulate to a large amount during propagation.

To better understand this, the figure above shows that even if you get the null-space direction correctly, simply moving the point in that direction will prevent the point from being placed on the surface, and this error will accumulate as propagation proceeds. To overcome this problem, after smoothing process we should project each vertex to high order reconstructed surface. We reconstructed surfaces via polynomial reconstruction, in which artifical oscillations can occur near ridge or corner(in three dimensions) if we use all neighbor points to reconstruct surfaces. To avoid oscillation, one should utilize local shape information computed from FOM-WENO to use points on smooth region as possible as one can when reconstructing surface.

3.1.6 FOM-WENO algorithm

In this section, we describe whole FOM-WENO algorithm in detail.

1. Compute each face normal $N_i = \dfrac{X_{i+1/2} - X_{i-1/2}}{|X_{i+1/2} - X_{i-1/2}|}$.
2. Construct cell averages $\bar{v}_j = N_{i+j} \cdot N_i$ $(j = -k+1, \cdots, k-1)$ to compute $(2k-1)$th order WENO coefficients ω_r for each computational stencil around $x_{i+\frac{1}{2}}$.
3. Using the WENO coefficients ω_r from 2, normals can be reconstructed by $\widetilde{N}^-_{i+1/2} = \sum_{r=0}^{k-1} \omega_r \widetilde{N}^{(r)}_{i+1/2}$. Here, $\widetilde{N}^{(r)}_{i+1/2} = \sum_{j=0}^{k-1} c_{rj} \widetilde{N}_{i-r+j}$ is the linear combination of the modified normal vectors $\widetilde{N}_i$.
4. Repeat 3 to the other stencil to get $\widetilde{N}^+_{i+1/2}$.
5. Find offset direction using $\widetilde{N}^-_{i+1/2}, \widetilde{N}^+_{i+1/2}$. This process is done same as finding offset direction for the conventional FOM scheme.
6. In case of wavefrontal motion, find wavefrontal length l to correct the movement of vertex in offset direction d. Whereas the conventional FOM scheme defines θ_i and θ_{i+1} by the angle between offset direction d and N_i, N_{i+1} , respectively, in FOM-WENO, replace N_i and N_{i+1} by $\widetilde{N}^-_{i+1/2}$ and $\widetilde{N}^+_{i+1/2}$.

During the computation of modified normal vectors $\widetilde{N}_i$, the curvature $|X''(t_{i-1/2})|$ is approximated by

$$\min\left(\frac{1}{r_1}, \frac{1}{r_2}\right),$$

where r_1 : radius of circle passing through $X(t_{i-1/2}), X(t_{i+1/2}), X(t_{i+3/2})$ and r_2 : radius of circle passing through $X(t_{i-3/2}), X(t_{i-1/2}), X(t_{i+1/2})$.

Remark) We have only considered the movement under uniform velocity. When the velocity function also has a shock at corner or ridge, the WENO scheme can be applied to the velocity function as well.

3.2 FOM-WENO scheme in three dimension

In this subsection, we use the WENO method on the triangular meshes discussed in section 2.4 to show FOM-WENO method can be extended well into three dimensions. All methodologies used in FOM-WENO scheme in two dimensions seem to extend well into three dimensions. In particular, we discuss high order normal vector reconstruction here, since it is the most important part in determining correct local shape of interface.

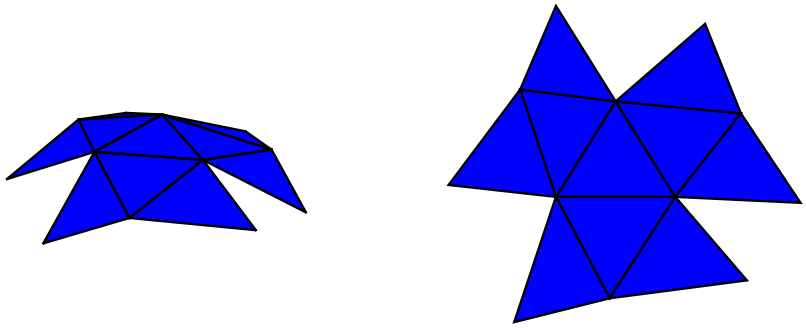

Figure 3.8: Unfolding process in three dimensions

Now consider a triangle $\triangle_0$ and a point p above it. To approximate a normal vector at p within a stencil of $\triangle_0$, triangles around $\triangle_0$ should be on the same plane. In general, this is not the case since the interface is usually a curved surface. To put triangles around $\triangle_0$ up on the same plane, as in two dimensional case, we need unfolding process, which is illustrated in figure 3.8. Now that the stencil of $\triangle_0$ lies in the same plane, one can calculate the WENO coefficient as described in section 2.4. As in two dimensional case, the function for calculating smooth indicator can also be taken in the same fashion.

$$\bar{v}_j = N_j \cdot N_0,$$

where N_0, N_j are normal vectors of $\triangle_0$, $\triangle_j$, respectively. In various numerical test, this function seems to be able to distinguish geometric shocks well.

When dealing with the WENO scheme on triangular mesh of poor quality, there can be several problems. First of all, the condition number of the matrix for finding linear weights can be very large. Even if the reconstruction polynomials are well calculated, it may be impossible to make all linear weights non-negative. In [23], WENO method is used for solving computational fluid dynamics problems on two-dimensional triangle mesh. In this case, since the initial mesh remains fixed during the analysis, once the WENO coefficient has been determined, the problem has not been repeated anymore. However, in interface tracking problem, the interface continues to move. As the interface moves, the shape of the stencil also changes. Therefore, the WENO coefficient must be continuously calculated, and the above problems may recur.

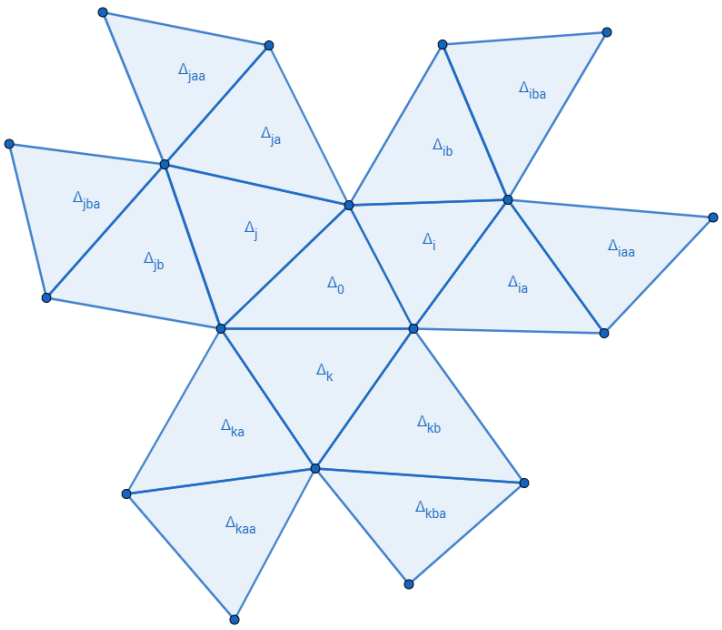

Figure 3.9: Wider stencils used for the third method

To avoid this problem, we prioritize three different WENO reconstruction methods and apply them to each triangle by chaning the method until the two previous problems do not occur. The first two methods follow the 3rd order WENO scheme introduced in section 2.4, and the only difference is

the way of grouping nine polynomials into three groups. In the first method, the first group is p_2, p_4, p_5, and the second group is p_3, p_6, p_7, and the third group consists of p_1, p_8, p_9. In the second method, we slightly modified these groups. p_3, p_4, p_5 is used for first group, and p_1, p_6, p_7 is used for second group, and finally the last group consists of p_2, p_8, p_9.

If two prior reconstruction results in large condition number of system or if non-negative linear weights cannot be found, the third reconstructionl method is then used. It uses more triangles(i,e. wider stencils) as shown in figure 3.9. In this case, since the reconstruction must be successful, it does not go through the process of obtaining high order in the smooth region by selecting linear weights well. That is, for each of the six stencils S_r shown in the following table 3.1, the second order reconstruction polynomial p_r is found in a least square sense, and the corresponding linear weights are set to be $\gamma_r = 1/6$. In this way, linear weights are always non-negative, and reconstruction polynomials in each stencil are also found well for most triangulations. For determining final non-linear weights ω_r via calculating the smooth indicator, it is sufficient to follow the procedure illustrated in subsection 2.4.4.

S_1	$\triangle_0$, $\triangle_i$, $\triangle_{ia}$, $\triangle_{ib}$, $\triangle_{iaa}$, $\triangle_{iba}$, $\triangle_j$
S_2	$\triangle_0$, $\triangle_i$, $\triangle_{ia}$, $\triangle_{ib}$, $\triangle_{iaa}$, $\triangle_{iba}$, $\triangle_k$
S_3	$\triangle_0$, $\triangle_j$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_{jaa}$, $\triangle_{jba}$, $\triangle_k$
S_4	$\triangle_0$, $\triangle_j$, $\triangle_{ja}$, $\triangle_{jb}$, $\triangle_{jaa}$, $\triangle_{jba}$, $\triangle_i$
S_5	$\triangle_0$, $\triangle_k$, $\triangle_{ka}$, $\triangle_{kb}$, $\triangle_{kaa}$, $\triangle_{kba}$, $\triangle_i$
S_6	$\triangle_0$, $\triangle_k$, $\triangle_{ka}$, $\triangle_{kb}$, $\triangle_{kaa}$, $\triangle_{kba}$, $\triangle_j$

Table 3.1: Stencils used for the reconstruction in the third method

FOM-WENO method does not contain the fourth order method introduced in section 2.4, because even small stencils are so large that shocks tend to be included all possible stencils. In this case, the normal vector can be reconstructed incorrectly.

3.3 Numerical experiments

In this section, we will present some numerical results based on our novel proposed FOM-WENO algorithm. Since our algorithm is based on the high order approximation of normal vectors at the interface, we first verify numerically that modified normal vector indeed gives high order accurate results. Then we also check high accurate result stay in long time simulation as well. Especially, numerical results presented for both types of corner propagation. As we pointed out in subsection 3.1.1, we show that FOM-WENO scheme is able to give highly accurate propagation results which are less dependent on geometric parameters. Furthermore, we compare the volume loss caused by conventional FOM scheme and FOM-WENO scheme. In subsection 3.3.5, we also extend our results to a non-uniform speed interface propagation. The numerical results for the interface under mean curvature motions. Finally, we present various surface propagation results using FOM-WENO schemes in two dimensions as well as in three dimensions to examine the extensibility of FOM-WENO scheme in three dimensional space.

3.3.1 Accuracy for normal vector approximation

We approximated normal vector of the ellipse $x^2 + \dfrac{y^2}{0.6^2} = 1$ at $x = 0.6, y = 0.48$. Exact normal vector is $N = (9/\sqrt{481}, 20/\sqrt{481})$. We measure the error by decreasing the average mesh size by half from $h_{avg} = 0.08$ to $h_{avg} = 0.0025$. Here, the error is measured by L^2 norm of the diffrence of exact normal vector and approximated normal vector.

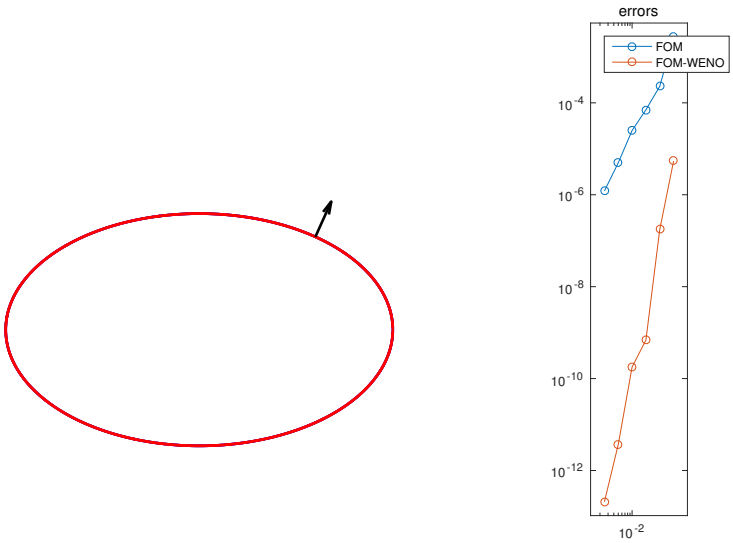

Figure 3.10: Left figure - Normal vector at $(0.6, 0.48)$, Right figure - Absolute error for normal approximation in conventional FOM scheme and FOM-WENO scheme.

Approximation Method	Accuracy Order
FOM(Left Normal)	0.9881
FOM(Right Normal)	1.0199
FOM(Averaged Normal)	2.2274
FOM-WENO(With N_i, No Normalization, Left Normal)	2.0268
FOM-WENO(With N_i, No Normalization, Right Normal)	1.9701
FOM-WENO(With N_i, No Normalization, Averaged Normal)	5.3013
FOM-WENO(With $\widetilde{N}_i$, No Normalization, Left Normal)	3.2320
FOM-WENO(With $\widetilde{N}_i$, No Normalization, Right Normal)	3.1969
FOM-WENO(With $\widetilde{N}_i$, No Normalization, Averaged Normal)	4.9265
FOM-WENO(With N_i, Normalization, Left Normal)	4.8983
FOM-WENO(With N_i, Normalization, Right Normal)	4.9884
FOM-WENO(With N_i, Normalization, Averaged Normal)	5.3014
FOM-WENO(With $\widetilde{N}_i$, Normalization, Left Normal)	4.9540
FOM-WENO(With $\widetilde{N}_i$, Normalization, Right Normal)	4.9375
FOM-WENO(With $\widetilde{N}_i$, Normalization, Averaged Normal)	4.9265

Table 3.2: Various approximation methods for normal vectors and their order of accuracy

Here, we need to explain various approximation methods used for approximating normal vectors. $N^-_{i+1/2}$ and $N^+_{i+1/2}$ are represented as left normal and right normal, respectively. For example, in the conventional FOM scheme, $N^-_{i+1/2} = N_i$ and $N^+_{i+1/2} = N_{i+1}$. In FOM-WENO scheme, $N^-_{i+1/2}$ and $N^+_{i+1/2}$ denote the approximation of normals using the stencils at left and right sides, respectively. In the conventional FOM scheme, the normal vector at point $(i+1/2)$ is finally obtained from the weighted average of left and right normal, and we represent this normal by averaged normal.

In addition to above circumstances, we have further tested the effect of using modified normal vector $\widetilde{N}_i$ described in subsection 3.1.3 as well as the effect of normalization. Here, normalization refers to the normalization of $N^-_{i+1/2}$ and $N^+_{i+1/2}$ computed by WENO reconstruction to a unit vector. That is, in approximation with normalization, we use the following $N^-_{i+1/2}$.

$$N^-_{i+\frac{1}{2}} = \frac{\sum_{r=0}^{k-1} \omega_r N^{(r)}_{i+\frac{1}{2}}}{\left| \sum_{r=0}^{k-1} \omega_r N^{(r)}_{i+\frac{1}{2}} \right|}.$$

From the table, we can confirm the following facts. First, as we wanted, FOM-WENO yields a high order accurate normals than conventional FOM scheme. Specifically, the first two results show that conventional FOM scheme could only approximate $N^-_{i+1/2}$ and $N^+_{i+1/2}$ with order 1, while FOM-WENO approximate them two vectors with order 2. One can see this order can be improved approximately by 1 with using modified normal vectors $\widetilde{N}_i$ as we have discussed in error analysis section. The order can also be improved nearly 5 using the normalization technique described above.

One can observe an interesting point in calculating the offset direction from the normal vectors using FOM scheme. Even though $N^-_{i+1/2}$ and $N^+_{i+1/2}$ are not calculated as high order, order can be improved during the calculation of offset direction. This seems to be due to the fact that length adjusting process for the wavefrontal type actually has the same effect as normalization process discussed above.

Next, we consider 5-flower shape interface, given by the following equation.

$$x = (1 + 0.2\sin 5\theta)\cos\theta, \quad y = (1 + 0.2\sin 5\theta)\sin\theta.$$

At $\theta = \dfrac{\pi}{3}$, exact normal vector is $N = (5+2\sqrt{3}, 5\sqrt{3}-4)/(2\sqrt{32-5\sqrt{3}})$. We again measure the error by decreasing the average mesh size by half from $h_{avg} = 0.08$ to $h_{avg} = 0.0025$. Numerical results are summarized in the following table 3.3.

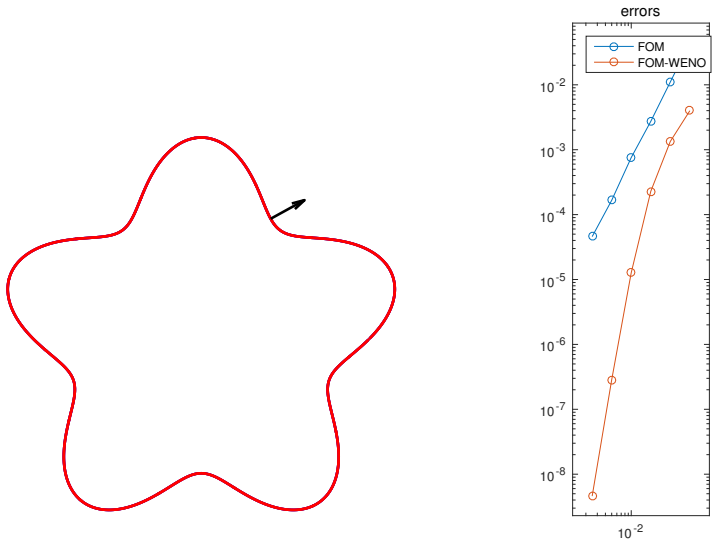

Figure 3.11: Left figure - Normal vector at $\theta = \frac{\pi}{3}$, Right figure - Absolute error for normal approximation in conventional FOM scheme and FOM-WENO scheme.

In this case, although FOM-WENO did not achieve up to fifth order, we can confirm that it still approximates normal vector higher order than conventional FOM scheme.

Method	Accuracy Order
FOM(Left Normal)	0.8811
FOM(Right Normal)	1.1110
FOM(Averaged Normal)	1.9860
FOM-WENO(With $\widetilde{N}_i$, Normalization, Left Normal)	3.7383
FOM-WENO(With $\widetilde{N}_i$, Normalization, Right Normal)	3.6627
FOM-WENO(With $\widetilde{N}_i$, Normalization, Averaged Normal)	3.9457

Table 3.3: Numerical results for normal approximation in 5-flower shape

Now we present numerical test results for normal vector approximations at geometric shock, especially for a expanding corner point. The geometric object we have chosen is the quadrant of a circle with radius 1. At $(0, 1)$, exact offset direction for corner point is given by $N = (1/\sqrt{2}, 1/\sqrt{2})$ (i,e. corner point moves with a speed 1). Note that exact left normal is $(0, 1)$ and exact right normal is $(1, 0)$. Numerical experiments were carried out under the same conditions as above to confirm the order. The results are presented in table 3.4.

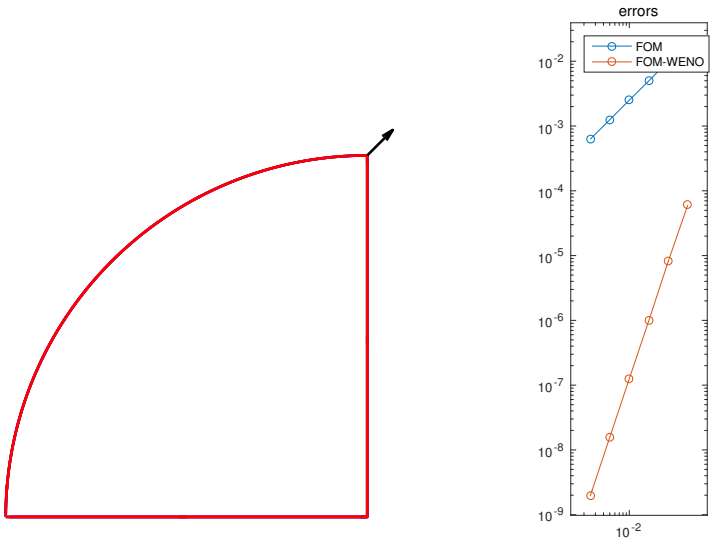

Figure 3.12: Left figure - Exact offset direction at $(0, 1)$, Right figure - Absolute error for conventional FOM scheme and FOM-WENO scheme.

In this case, one can observe that the order of FOM-WENO is close to 3. This is because only one-sided stencil is mainly used in 5th order WENO reconstruction, resulting the WENO scheme almost similar to 3rd order ENO scheme. Here, the order for right normal approximation is not computed, since the right normal is constant $(1, 0)$ and any consistent scheme would give exact right normal approximations.

Method	Accuracy Order
FOM(Left Normal)	0.9945
FOM(Right Normal)	-
FOM(Averaged Normal)	0.9945
FOM-WENO(With $\widetilde{N}_i$, Normalization, Left Normal)	2.9821
FOM-WENO(With $\widetilde{N}_i$, Normalization, Right Normal)	-
FOM-WENO(With $\widetilde{N}_i$, Normalization, Averaged Normal)	2.9821

Table 3.4: Numerical results for normal approximation at an expanding corner point

Test for contracting corner point is also done. The curve to the left of corner point $(0, 0)$ is composed of a quadrant of circle with radius 1. Exact offset direction at $(0, 0)$ is $(1, 1)$ (i,e. The corner points moves with a speed $\sqrt{2}$.). The following table 3.5 shows corresponding numerical results.

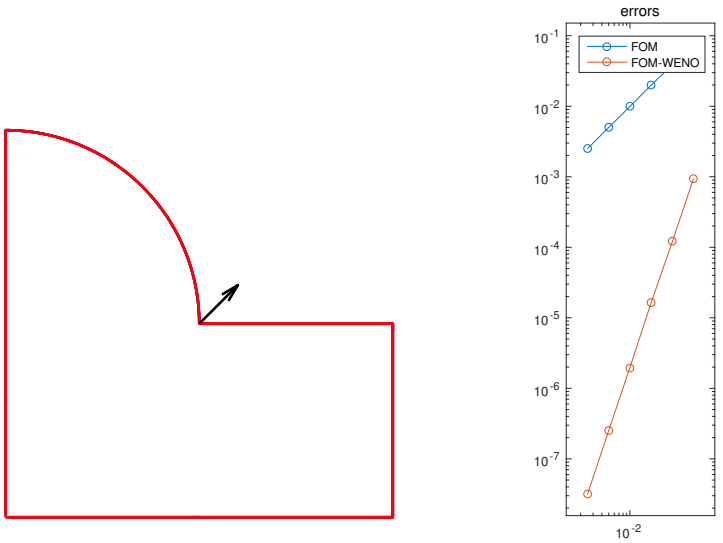

Figure 3.13: Left figure - Exact offset direction at $(0, 0)$, Right figure - Absolute error for conventional FOM scheme and FOM-WENO scheme.

In this case, the order of normal approximation of FOM-WENO is close to 3 because of the same reason as above.

Method	Accuracy Order
FOM(Left Normal)	0.9945
FOM(Right Normal)	-
FOM(Averaged Normal)	0.9839
FOM-WENO(With $\widetilde{N}_i$, Normalization, Left Normal)	2.9774
FOM-WENO(With $\widetilde{N}_i$, Normalization, Right Normal)	-
FOM-WENO(With $\widetilde{N}_i$, Normalization, Averaged Normal)	2.9775

Table 3.5: Numerical results for normal approximation at a contracting corner point

Finally, let us consider the worst case where geometric shock detect function $f(j;i) = n_{i+j} \cdot n_i$, does not work well. The following shape is made by attaching two quadrants of a circle with radius 1. At the intersection of two quadrants, $(0,0)$, the first-order derivative of the interface is continuous but the second derivative is discontinuous. Exact normal at $(0,0)$ is given by $N = (1,0)$.

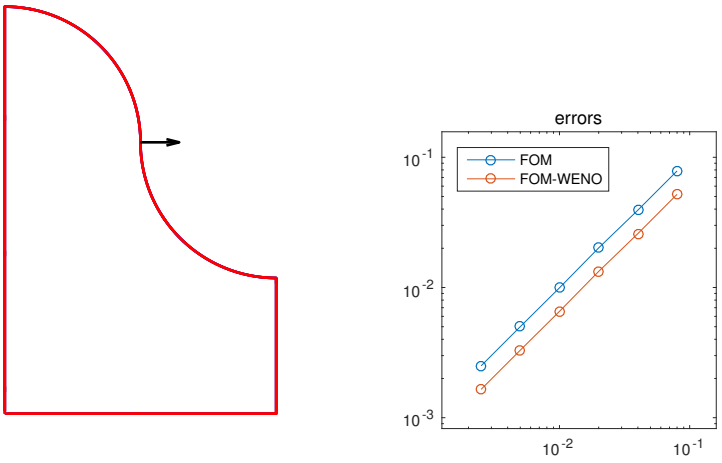

Figure 3.14: Left figure - Exact normal vector at $(0,0)$, Right figure - Absolute error for conventional FOM scheme and FOM-WENO scheme.

Method	Accuracy Order
FOM(Left Normal)	0.9945
FOM(Right Normal)	0.9945
FOM(Averaged Normal)	0.9945
FOM-WENO(With $\widetilde{N}_i$, Normalization, Left Normal)	0.9953
FOM-WENO(With $\widetilde{N}_i$, Normalization, Right Normal)	0.9953
FOM-WENO(With $\widetilde{N}_i$, Normalization, Averaged Normal)	0.9953

Table 3.6: Numerical results for normal approximation at a saddle point

Since the first derivative is continuous, the WENO smooth indicator is difficult to distinguish between the appropriate stencil to be used and therefore the accuracy is limited. Nevertheless, one can see that the magnitude of the absolute error itself is smaller in FOM-WENO than conventional FOM scheme.

3.3.2 Long time accuracy for corner propagation

We pointed out that normal vectors computed by conventional FOM scheme have low accuracy even in smooth region, corner point tends to move in inexact direction. In this subsection, we numerically compare the performance of conventional FOM scheme and FOM-WENO scheme in long time propagation by measuring error norm of a corner point as time evolves. For this experiment, the geometric shape which we used earlier to check the accuracy of normal vector at the contracting corner point(figure 3.13) is chosen. The normal velocity of interface is fixed to constant 1, and the time progressed from 0 to 1. We set time step to be sufficiently small, so that we can avoid additional process(such as remeshing procedure for maintaining mesh quality). Specifically, we set timestep to 1/20 times the average mesh size in each experiment. Again, average mesh is set to decrease by half from $h_{avg} = 0.08$ to $h_{avg} = 0.0025$. The order of accuracy for location of corner point is then calculated. For time stepping, we used TVD-RK3 illustrated in section 2.5. In addition, 10 null-space smoothing steps were performed for each move to prevent overlapping of meshes. Exact corner location can be computed as follows.

$$\begin{cases} (-0.5 + \sqrt{t + 0.25}, t) & \text{for } 0 \leqslant t \leqslant 0.75 \\ (0.5t + 0.125, \sqrt{0.75t^2 + 0.375t - 0.140625}), & \text{for } t \geqslant 0.75 \end{cases}.$$

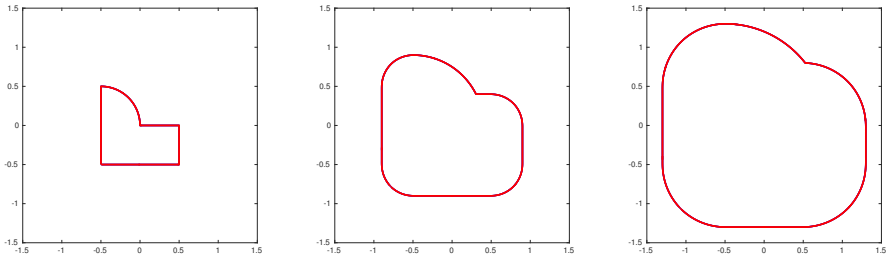

Figure 3.15: Interfaces at $t = 0, 0.4, 0.8$, Blue : FOM scheme, Red : FOM-WENO scheme

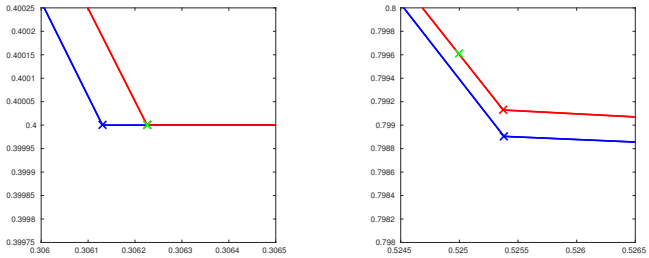

Figure 3.16: Interfaces near contracting corner points at $t = 0.4, 0.8$, Blue : FOM scheme, Red : FOM-WENO scheme, Green : Exact corner point location

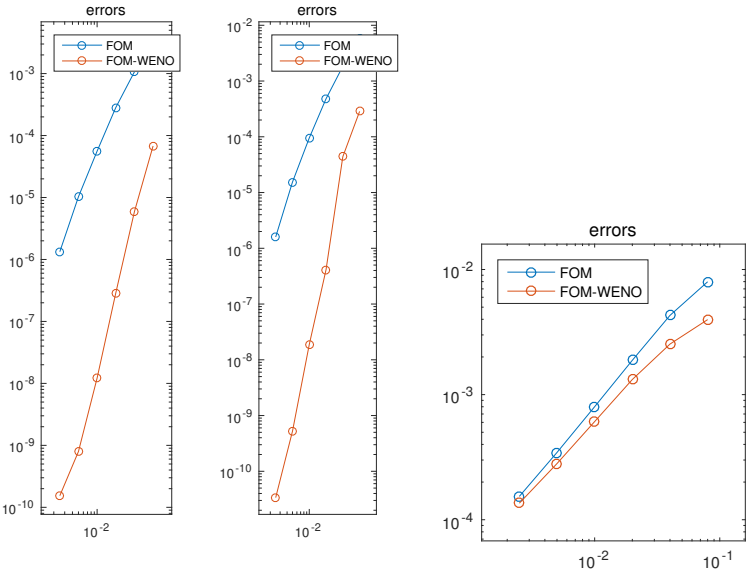

Figure 3.17: Absolute error for contracting corner location at $t = 0.08, 0.4, 0.8$, Blue : FOM scheme, Red : FOM-WENO scheme

For $t < 0.75$, the error norm of FOM-WENO is much smaller than that of $t > 0.75$. This is because the magnitude of the error norm induced by colliding two circular wavefront originating from $(0, 0)$ and $(1, 0)$ is very large.

$t = 0.08$, Method	Accuracy Order
FOM	2.2650
FOM-WENO(With $\widetilde{N}_i$, Normalization)	3.7501
$t = 0.4$, Method	Accuracy Order
FOM	2.3638
FOM-WENO(With $\widetilde{N}_i$, Normalization)	4.6021
$t = 0.8$, Method	Accuracy Order
FOM	1.1398
FOM-WENO(With $\widetilde{N}_i$, Normalization)	0.9750

Table 3.7: Accuracy order for contracing corner location at $t = 0.08, 0.4, 0.8$

Therefore, the FOM-WENO is better than conventional FOM scheme before $t < 0.75$ in terms of accuracy order, but there is no significant difference thereafter.

Now, we fix the mesh size to $h_{avg} = 0.01$. The follwing graph shows the error norm of the corner point location in conventional FOM scheme and FOM-WENO scheme until time $t = 1$.

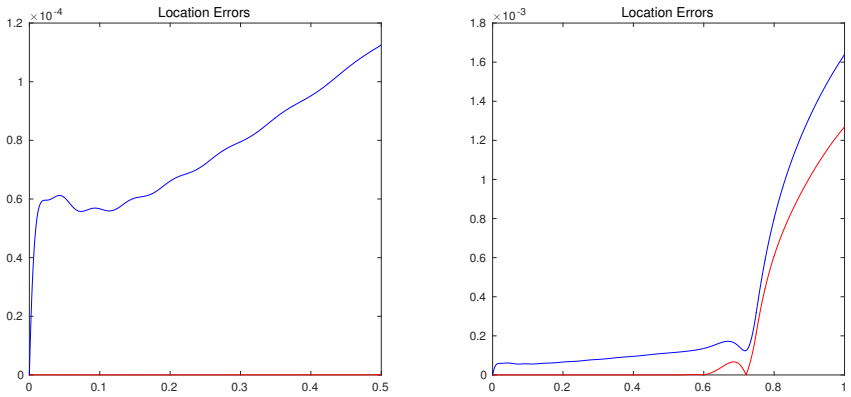

Figure 3.18: Error of the contracting corner point location with $h_{avg} = 0.01$ from $t = 0$ to $t = 1$, Blue : FOM scheme, Red : FOM-WENO scheme

In the beginning, the error norm of the FOM-WENO is close to 0 and much smaller than that of conventional FOM scheme(see the graph on the left), but the difference between two errors decreases as the time starts approaching to 0.75. Over the entire time, the magnitude of error is always smaller in FOM-WENO scheme.

To confirm the accuracy of propagation at an expanding corner point, we take initial interface as a quadrant of circle(given in figure 3.12) and propagate it by unit speed in normal direction. This time, the time step is taken equal to the average mesh size. This is because, in the case of expanding motion, it is less likely that mesh overlaps or the quality becomes worse even if the time step is larger than the contracting motion. The expanding corner point is set to be fixed during null-space smoothing process, otherwise it moves in the smoothing step since the local shape near expanding corner point is usually recognized to smoothing region. Other expermiental conditions are set to be equal to the experiment of contracting corner point. We measure errors and the order of accuracy at time $t = 0.08, 0.32$. The exact corner location is given by $(0.5 + t/\sqrt{2}, 0.5 + t/\sqrt{2})$.

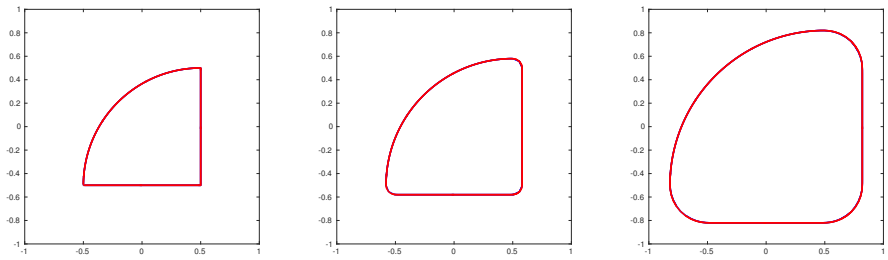

Figure 3.19: Interfaces at $t = 0, 0.08, 0.32$, Blue : FOM scheme, Red : FOM-WENO scheme

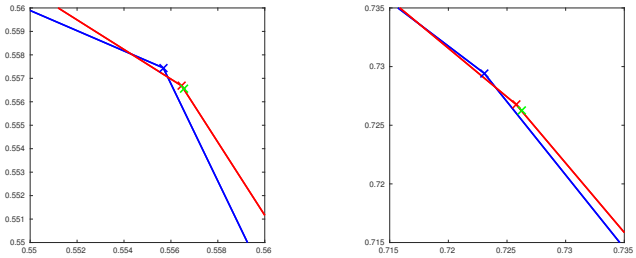

Figure 3.20: Interfaces near expanding corner points at $t = 0, 0.08, 0.32$, Blue : FOM scheme, Red : FOM-WENO scheme, Green : Exact corner point location

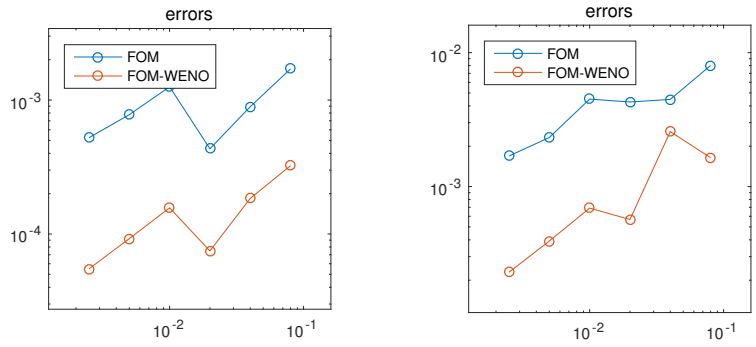

Figure 3.21: Absolute error for expanding corner point location at $t = 0.08, 0.32$, Blue : FOM scheme, Red : FOM-WENO scheme

$t = 0.08$, Method	Accuracy Order
FOM	0.3417
FOM-WENO(With $\widetilde{N}_i$, Normalization)	0.5120
$t = 0.32$, Method	Accuracy Order
FOM	0.4490
FOM-WENO(With $\widetilde{N}_i$, Normalization)	0.5680

Table 3.8: Accuracy order for expanding corner location at $t = 0.08, 0.32$

In the experiment regarding an expanding corner point, both schemes have poor order of accuracy because the process for rounding the sharp corner point harms order of accuracy seriously. Nevertheless, it is noteworthy that the error norm of the FOM-WENO scheme is only about 1/10 times compared to that of conventional FOM scheme. This is mainly due to the fact that the FOM-WENO scheme rounds expanding corner much better than conventional FOM scheme. This feature become more pronounced as the expanding corner becomes more acute. In order to see this property clearly, we moved the following sharp diamond shape centered at $(0, 0)$ with acute angle $\theta = 28.0725^\circ$ by a unit speed.

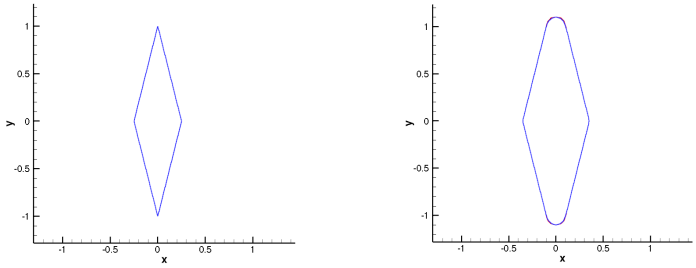

Figure 3.22: Interfaces at $t = 0, 0.1$, Blue : FOM scheme, Red : FOM-WENO scheme

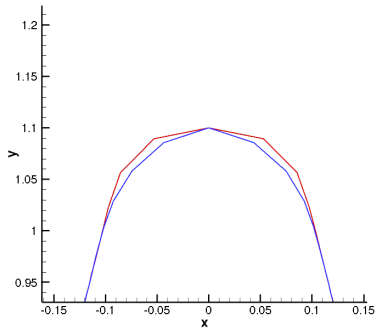

Figure 3.23: Interface near expanding corner points at $t = 0.1$, Blue : FOM scheme, Red : FOM-WENO scheme

One can easily see that FOM-WENO rounds the expanding corner much better than conventional FOM scheme.

Finally, to measure the long time accuracy in smooth region, we again take ellipse $x^2 + \dfrac{y^2}{0.6^2} = 1$ and a point $(x, y) = (0.6, 0.48)$ on it, which is used for verifying high order normal approximations in subsection 3.3.1. As in the case of expanding corners, the point is fixed during null-smoothing process. All other numerical experimental conditions are set equal. As a function of time, the exact location is given by $(0.6 + 9t/\sqrt{481}, 0.48 + 20t/\sqrt{481})$.

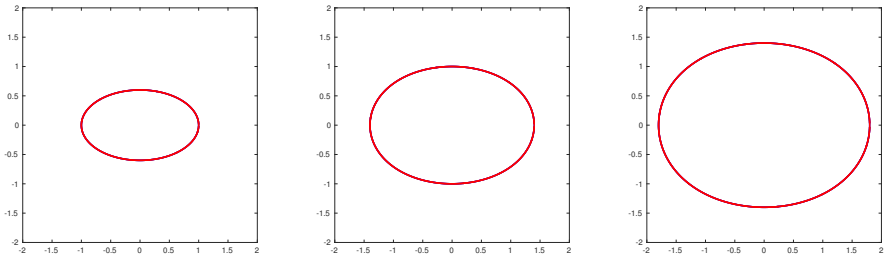

Figure 3.24: Interfaces at $t = 0, 0.4, 0.8$, Blue : FOM scheme, Red : FOM-WENO scheme

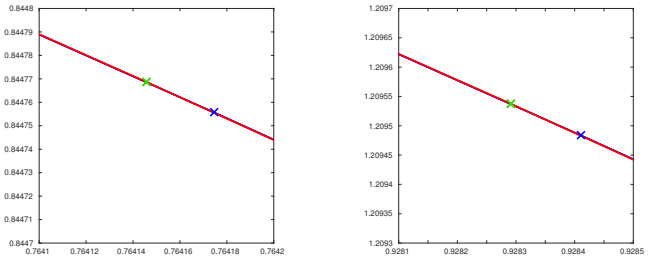

Figure 3.25: Interfaces near a tracking point at $t = 0.4, 0.8$, Blue : FOM scheme, Red : FOM-WENO scheme, Green : Exact point location. In the figure, the point location of the FOM-WENO result overlaps with the exact point location.

One can see that both conventional FOM scheme and FOM-WENO scheme capture the interface relatively well, and results of two methods are almost same. However, if one magnify interfaces near a tracking point, difference between two schemes can be clearly seen.

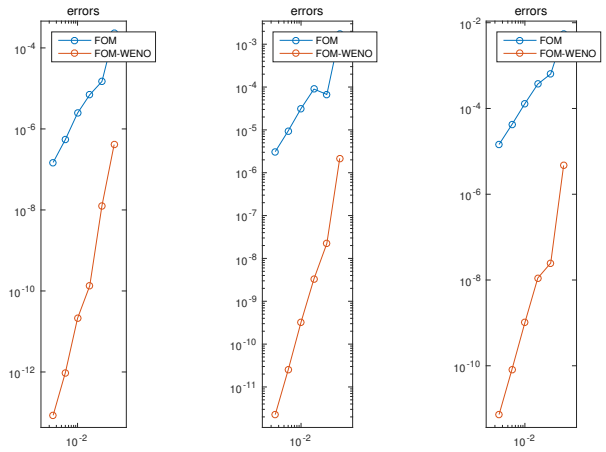

Figure 3.26: Absolute error for the point location in smooth region at $t = 0.08, 0.4, 0.8$, Blue : FOM scheme, Red : FOM-WENO scheme

$t = 0.08$, Method	Accuracy Order
FOM	2.1180
FOM-WENO(With $\widetilde{N}_i$, Normalization)	4.4384
$t = 0.4$, Method	Accuracy Order
FOM	1.8247
FOM-WENO(With $\widetilde{N}_i$, Normalization)	3.9730
$t = 0.8$, Method	Accuracy Order
FOM	1.7108
FOM-WENO(With $\widetilde{N}_i$, Normalization)	3.8593

Table 3.9: Accuracy order for the point location in smooth region at $t = 0.08, 0.4, 0.8$

In the graphs and tables above, the order of accuracy for the FOM-WENO is much higher than that of the conventional FOM scheme and the error norm is much smaller. From this, it can be confirmed once again that the FOM-WENO captures the interface with high order accuracy in the smooth region. The following graph compares the error norm of conventional FOM scheme and FOM-WENO scheme when the average mesh size is fixed to $h_{avg} = 0.01$ and the timestep is equal to mesh size during $t = 0 \sim 1$.

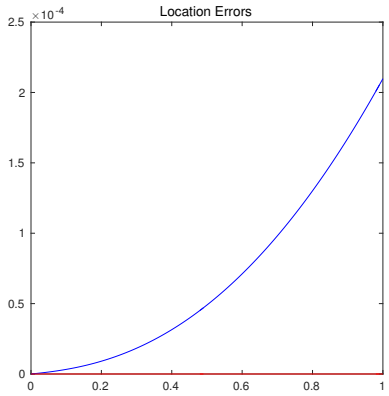

Figure 3.27: Error norm of the point location in smooth region with $h_{avg} = 0.01$ during $t = 0 \sim 1$, Blue : FOM scheme, Red : FOM-WENO scheme

In the graph, it can be seen that the FOM-WENO scheme has a much smaller error norm than conventional FOM scheme over the entire time.

3.3.3 Geometric stability of FOM-WENO scheme

In this subsection, we show by example that the FOM-WENO scheme can stably distinguish geometric shocks. In conventional FOM scheme, the propagation result is very sensitive to geometric parameter ϕ_r. However, in FOM-WENO scheme, by setting the geometric value ϕ_r to be small enough, we can get correct propagation results. Let's consider propagation of an initial interface with unit speed from $t = 0$ to $t = 0.3$, especially focusing on propagation of three distinct corner points.

Figure 3.28: Initial geometry containing both expanding and contracting corner points.

For FOM-WENO scheme, the feature parameter ϕ_r is fixed to 10° and the final propagation result is shown with red lines. In order to compare with this result, we propagate initial surface with conventional FOM scheme as well, where the parameter value is set to two different values, $\phi_r = 10^\circ$ and $\phi_r = 30^\circ$, respectively.

Figure 3.29: For small feature parameters ($\phi_r = 10^\circ$), expanding corner which should be rounded can be sharpen in conventional FOM scheme. Blue : FOM scheme, Red : FOM-WENO scheme at $t = 0.3$

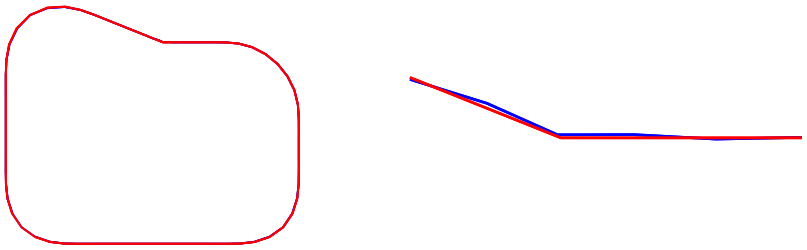

Figure 3.30: For large feature parameters ($\phi_r = 30^\circ$), conventional FOM scheme may misdirect contracting corner points. Blue : FOM scheme, Red : FOM-WENO scheme at $t = 0.3$

In FOM scheme, depending on feature parameters, ambiguous points are considered to have different geometry. Small feature parameters may result in sharp shape at expanding corner which should be rounded. Large feature parameters may misdirect contracting corner. Therefore, to move the interface correctly, the parameter value should not be too large or to small, and it is not easy to find the appropriate parameter value. In particular, misdirection of contracting corner caused by large feature parameters is a very serious problem, so parameter values must be set to avoid all of these misdirections. In this case, sharp shapes occurring at the expanding corner are difficult to avoid. On the other hand, since FOM-WENO scheme captures the smooth region more accurately through high-order reconstruction, even if the initial feature parameter value is set small enough to avoid all misdirects in the contracting corner, such sharp shape occurring at expanding corner can be avoided. In other words, the FOM-WENO scheme has a significantly lower dependence of the results on the parameter values than the conventional method, and can more stably propagate interfaces.

3.3.4 Comparison of volume loss

Now, relative volume loss of the conventional FOM scheme and the FOM-WENO scheme for ellipse propagation under unit normal velocity is compared. The ellipse is again given by the equation $x^2 + \frac{y^2}{0.6^2} = 1$. The following figure 3.31 shows the volume loss calculated at time $t = 0.08, 0.4$ according to the average mesh size varying from $h_{avg} = 0.005$ to $h_{avg} = 0.08$. Here, timestep for propagation is set equal to average mesh size and TVD-RK3 is used for time stepping.

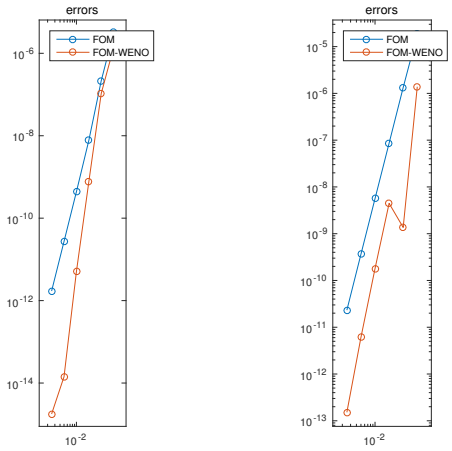

Figure 3.31: Volume loss of both methods at $t = 0.08, 0.4$, Blue : FOM scheme, Red : FOM-WENO scheme

The order is high in both methods, but FOM-WENO scheme has almost 100 times smaller absolute error compared to conventional FOM scheme. The following figure 3.32 shows the volume error of both methods at each time $t \in [0, 20]$. Here, the average mesh size and time step are fixed to 0.01. One can check that the volume error for the FOM-WENO scheme is much smaller than that of conventional FOM scheme at all times.

$t = 0.08$, Method	Accuracy Order
FOM	4.1685
FOM-WENO(With $\widetilde{N}_i$, Normalization)	5.9067
$t = 0.4$, Method	Accuracy Order
FOM	3.9213
FOM-WENO(With $\widetilde{N}_i$, Normalization)	4.6248

Table 3.10: Accuracy order for volume loss at $t = 0.08, 0.4$

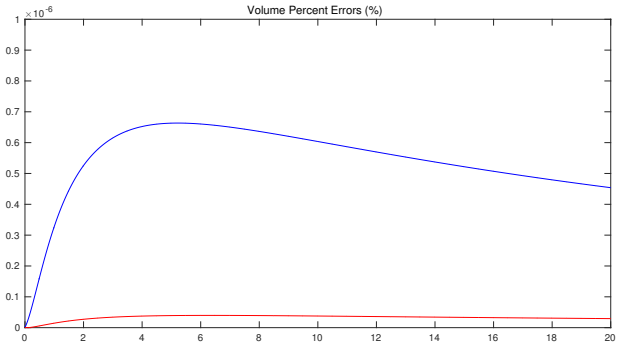

Figure 3.32: Volume loss caused by both methods at each time $t \in [0, 20]$, Blue : FOM scheme, Red : FOM-WENO scheme

Here one should note that depending on how the volume is measured, the accuracy order for volume loss can be predetermined regardless of which interface tracking method used. For example, if you simply calculate the volume of a polygon, the volume error between the exact volume and the polygon is already restricted to second order, regardless of the type of interface tracking method. In this case, the performance of FOM scheme and FOM-WENO scheme can not be compared properly. Therefore, we need a high order volume computation method that does not affect to evaluate the performance of two interface tracking methods. In this experiment, the volume is estimated by using 6th order polynomial reconstruction.

We now compare relative volume loss of conventional FOM scheme and the FOM-WENO scheme for square propagation under unit normal velocity. All numerical quantities such as mesh size and timestep are taken values with which we used in ellipse propagation.

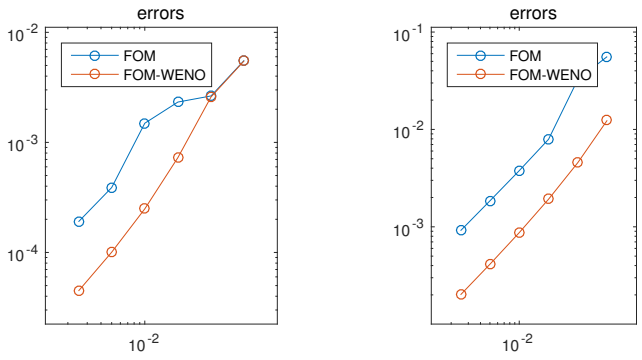

Figure 3.33: Volume loss of both methods at $t = 0.08, 0.4$, Blue : FOM scheme, Red : FOM-WENO scheme

$t = 0.08$, Method	Accuracy Order
FOM	0.9738
FOM-WENO(With $\widetilde{N}_i$, Normalization)	1.3896
$t = 0.4$, Method	Accuracy Order
FOM	1.1876
FOM-WENO(With $\widetilde{N}_i$, Normalization)	1.1900

Table 3.11: Accuracy order for volume loss at $t = 0.08, 0.4$

As discussed above, the order of accuracy is limited in rounding expanding corner points. In both cases, the order of accuracy is close to 1. However, the absolute error is smaller in FOM-WENO scheme than that of conventional FOM scheme. To confirm this more accurately, we again set mesh size and timestep to 0.01 and extracted volume error of both methods at each time $t \in [0, 20]$.

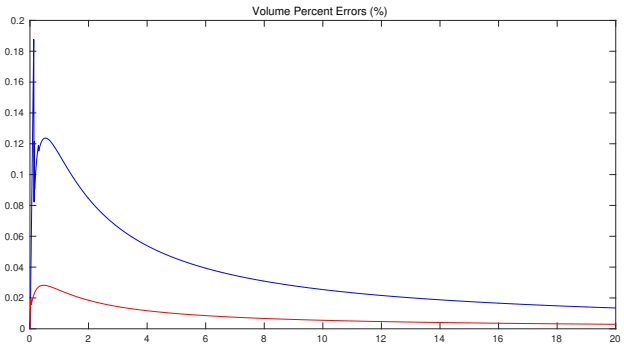

Figure 3.34: Volume loss caused by both methods at each time $t \in [0, 20]$, Blue : FOM scheme, Red : FOM-WENO scheme

In the graph, it can be confirmed once again that volume error induced by the FOM-WENO scheme is smaller than that of conventional FOM scheme. In other words, using the FOM-WENO scheme, the volume loss can be kept lower than the conventional FOM scheme in a smooth region as well as in a region rounding occurs.

3.3.5 Propagating under non-uniform normal velocity

In all previous numerical examples, only interfaces moving at constant speed in normal direction were considered. Now, to consider the non-uniform normal velocity, we propagate interfaces by mean curvature flow, whose equation is given as follows.

$$\frac{\partial \mathbf{x}}{\partial t} = -\kappa \mathbf{n}(\mathbf{x}, t), \quad \kappa = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{\frac{3}{2}}}.$$

The interface moving under mean curvature flow has the property that the protruding part of the initial interface gradually moves inward as time passes, and the sunken part of the initial interface gradually moves to the outside. Therefore, interface becomes close to the circular shape. if when the interface once reaches the circle, as the time passes, the circle becomes smaller and eventually it converges to the center. Here, we present the mean curvature motion of 5-flower shape during $t = 0 \sim 0.2$.

The following figure shows the moving interface every 0.04 sec. It can be seen that the FOM-WENO scheme moves the interface well even under circumsatances where non-uniform normal speed is given. As discussed in the remarks of subsection 3.1.6, normal speed at given point is also reconstructed from a value given at each face with high order using the WENO scheme.

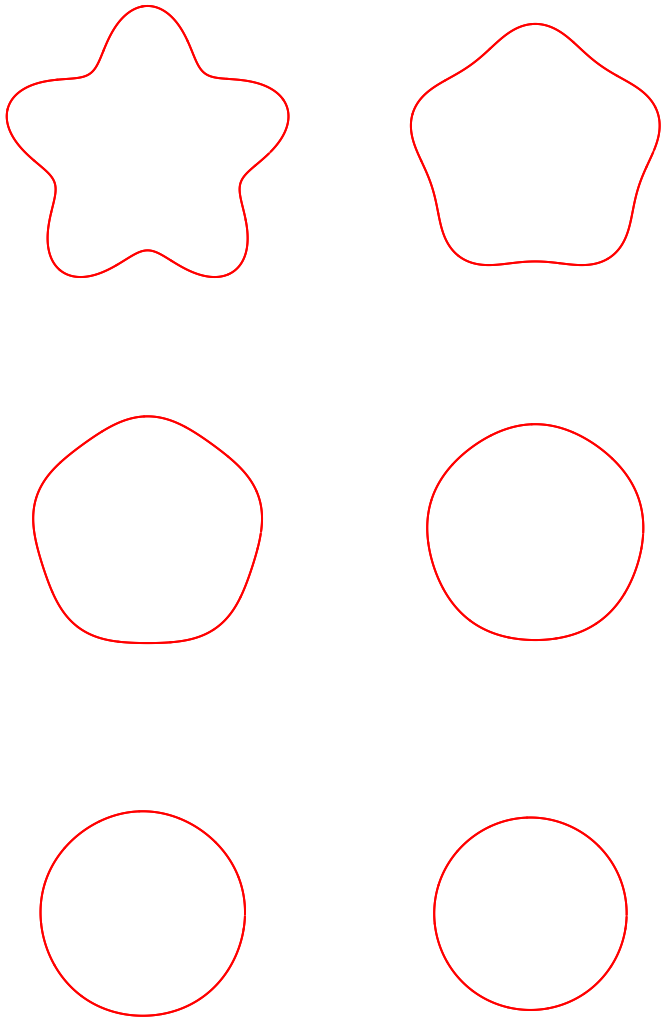

Figure 3.35: Curvature flow evolution of 5-flower shape using FOM-WENO scheme at $t = 0, 0.04, 0.08, 0.12, 0.16, 0.2$.

3.3.6 Results in three dimension

To illustrate the implementation of the FOM-WENO scheme in three dimensions, we consider a unit speed normal propagation of unit cube interface from $t = 0$ to $t = 1$. The average mesh size is fixed to 0.04 and the timestep is set to 0.01, which is 1/4 of average mesh size. The parameter values are selected to $\phi_r = 20^\circ$, $\phi_c = 45^\circ$.

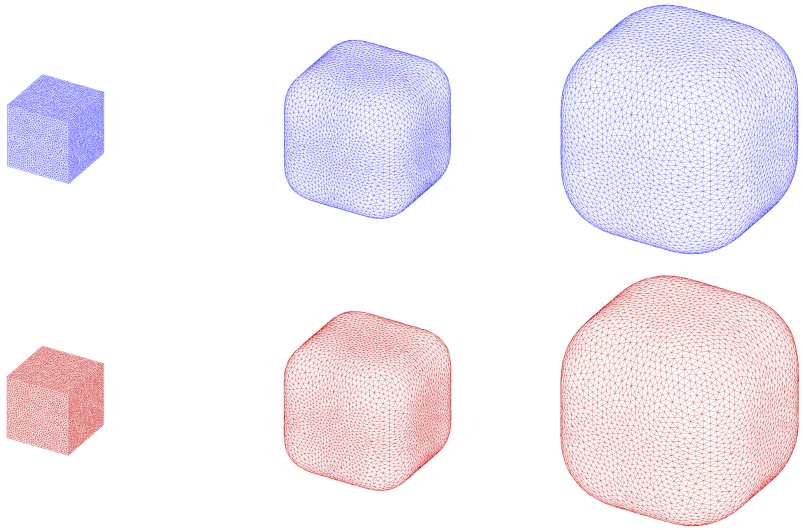

Figure 3.36: Propagation results for unit cube interface at $t = 0, 0.5, 1$, Blue : FOM scheme, Red : FOM-WENO scheme

It can be seen from the above figure 3.36 that both conventional FOM scheme and FOM-WENO scheme propagate interface well enough. To see the difference, we enlarge the interface near expanding corners after projecting the interface at $t = 0.3$ to xy plane.

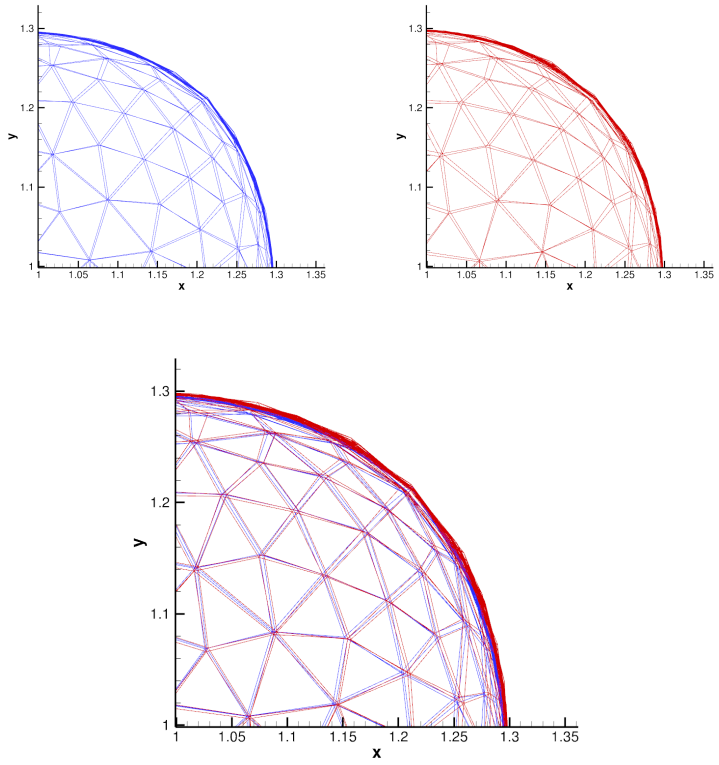

Figure 3.37: Interfaces near an expanding corner point(magnified and overlapped) at $t = 0.3$, Blue : FOM scheme, Red : FOM-WENO scheme

In a parallel with 2D results, the interface propagated from FOM-WENO scheme is more rounded near the expanding corners than the interface propagated from conventional FOM scheme. Therefore, it can be seen that FOM-WENO scheme can deal with expanding corners better than conventional FOM scheme by approximating normal vector reconstruction to higher order.

To compare the dependence of geometric parameters in both methods, we lowered the value of ϕ_r to 10°. In the figure 3.38, the results of conventional FOM and FOM-WENO scheme when $t = 0.3$ is presented. One can see that the resulting mesh for conventional FOM scheme is completely distorted.

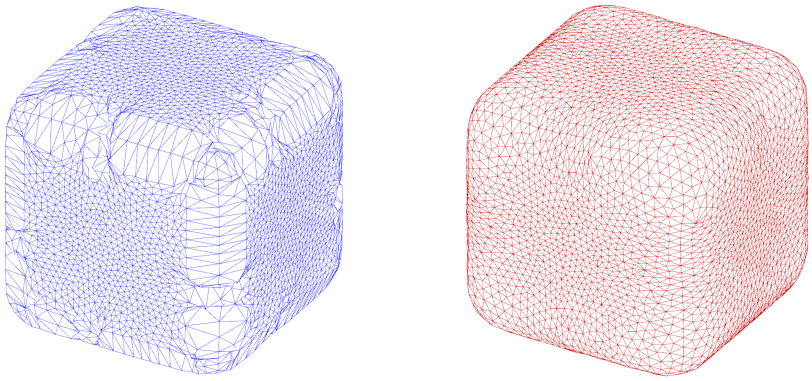

Figure 3.38: Propagation Results at $t = 0.3$ with $\phi_r = 10^\circ$, Blue : FOM scheme, Red : FOM-WENO scheme

The resason for such distortion is shown in the figure 3.39, which shows the local shape of each point determined by conventional FOM scheme at $t = 0.12$. In the figure, in addition to the points on the edge that should be judged as ridges, one can see that the local shapes are misjudged as ridges even in the points actually located in the smooth region on the immediate side of actual ridges. Since conventional FOM scheme only uses neighboring face to approximate normal vector, when ϕ_r is small, the method is vulnerable to judge a point to be in the ridge even though a point is actually located in the smooth region. Points that are judged to have a wrong local geometry will be propagated in the wrong direction and will not be smoothed properly in null-space smoothing. Particularly for three dimensions, since the geometric shape can be more complicated, this sort of error often occurs. Therefore, wrong judgement of the local geometry depending on geometric parameters can yield the completely wrong result.

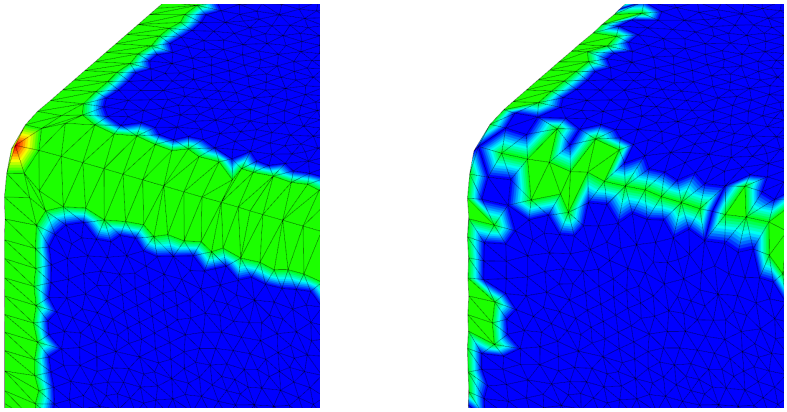

Figure 3.39: Local shape of interface at $t = 0.12$ in conventional FOM scheme(left) and FOM-WENO scheme(right), Blue : smooth region, Green : ridge, Red : corner. In conventional FOM scheme, the local shapes are misjudged as ridges for the points actually located in the smooth region.

However, FOM-WENO scheme distinguishes between ridges and smooth regions much better, and as a result, the resulting mesh is also much better than conventional FOM scheme. As discussed in two dimensions, it can be seen that FOM-WENO scheme which uses high order reconstruction of normals is less sensitive to geometric parameters in three dimensions than conventional FOM scheme as well.

Finally, we compared how FOM scheme and FOM-WENO scheme capture the initial local shape using the submarine interface which has more complex geometries, to confirm that FOM-WENO scheme distinguishes local geometry better than conventional FOM scheme.

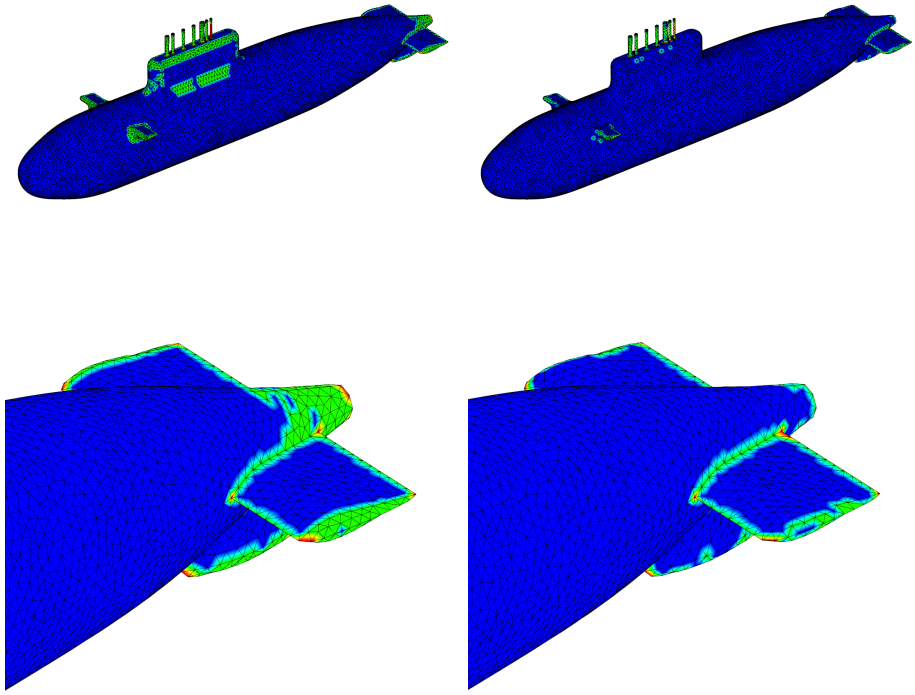

Figure 3.40: Local shape of submarine interface determined by conventional FOM scheme(Left) and FOM-WENO scheme(Right), Blue : smooth region, Green : ridge, Red : corner

In the figure 3.40, one can see that FOM-WENO scheme catches the local shape much more neatly than conventional FOM scheme. Nevertheless, neither of these methods can completely capture the local shape because the face offsetting method itself has limitations that make it impossible to perfectly capture the local shape in three dimensions.

Chapter 4

Conclusion

In this thesis, we have developed a new high order interface tracking method, FOM-WENO scheme, by combining the face offsetting method(FOM) [18] and the weighted essentially non-oscillatory scheme(WENO) [21]. FOM-WENO scheme maintains the advantages of conventional FOM scheme as well as overcomes disadvantages of conventional FOM scheme. That is, the scheme is also fast as it is based on the Lagrangian method, it can round the expanding corner well, and it can automatically consider the local geometry at each vertex, while overcoming the lower accuracy and sensitivity to geometric parameters. We have mathematically proved the accuracy of the FOM-WENO scheme in smooth regions through error analysis. Furthermore, we have designed a smooth indicator using appropriate function to make it work reliably even near non-smooth regions such as ridges and corners.

We also tested the FOM-WENO scheme through various numerical experiments. In subsection 3.3.1, it is confirmed that the FOM-WENO scheme approximates the normal vector better than the conventional FOM scheme in various types of interfaces. For the smooth curve, the two methods showed a large difference in the order of accuracy. Near the expanding / contracting corner, the FOM-WENO scheme had the higher accuracy order and the error norm was smaller in all cases, including the worst case at corner points having discontinuous second derivative. In subsection 3.3.2, we compared the accuracy of the interface moving with time using the third order TVD-RK method [29]. For points lying in the smooth region and corner points moving under contracting motions, the FOM-WENO scheme has a higher order of accuracy and able to capture the position of the interface much better. In expanding corner propagation, there was a limit on the order

caused by rounding effect. Nevertheless, the FOM-WENO scheme exhibited smaller error than conventional FOM scheme. This is due to the fact that the FOM-WENO scheme rounds expanding corner more gentle and approximates the interface more accurately. In subsection 3.3.3, we compared the dependence of the parameter in both methods. In the case of FOM scheme, it is difficult to fix the appropriate parameter value because large and small values of χ_r can yield different problems. However, FOM-WENO scheme is stable if the value of χ_r is set to be small enough. In subsection 3.3.4, the volume loss for interface propagation is measured. For smooth surfaces, FOM-WENO scheme has a higher order of accuracy and a much smaller volume loss than the conventional FOM scheme. In the case where rounding at the corner occurs, although the order of FOM-WENO scheme is limited, by rounding the corner more gently than conventional FOM scheme, the resulting volume loss is much smaller than conventional FOM scheme. Finally, in subsection 3.3.5, we have tested the mean curvature flow as an example of the case where the normal speed is not uniform. It can be seen that the FOM-WENO scheme moves the interface well even when the non-uniform normal velocity is given.

In this thesis, we do not consider remeshing processes, such as mesh splitting, collapsing, and flipping process, which are used when the mesh is distorted as the interface moves, to compare only the accuracy of the interface tracking method. For the same reason, we also did not discuss the methods to deal with topological changes. When applying such additional procedure depicted in [34], combining with suitable higher order surface reconstruction, such as the method presented in subsection 3.1.5, will yield higher accuracy. In addition, the null-space smoothing method we used in the smoothing process is actually based on the weighted Laplacian smoothing method. In Laplacian smoothing method, when the initial mesh size is not uniform throughout the interface, the mesh size distribution can not be maintained during smoothing process. By using more general smoothing methods such as mass spring method [35] or methods in [36, 37, 38], it seems to be able to overcome such a limitation.

We also demonstrated the feasibility of the FOM-WENO scheme methodology using WENO scheme on triangular meshes [23] even for three dimensions. Various numerical experiments show that the FOM-WENO scheme judges local geometry better than the conventional FOM scheme, implements rounding more accurately as the expanding corner advances, and has less dependence on the geometric parameters. However, it is difficult to completely distinguish three types of local shapes by using three eigenvalues in the face offsetting method. In particular, it is easy to determine the lo-

cal shape wrong in the vicinity of a special geometry such as the saddle point, and such misjudgement can yield the interface move in wrong direction. Therefore, it is further expected that some additional methods used in [39, 40, 32] can be combined with the FOM-WENO scheme for more accurate detection of ridges and corners, resulting in a more precise scheme.

Bibliography

[1] R. Fedkiw, S. Osher, The level set methods and dynamic implicit surfaces, Springer.

[2] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49.

[3] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A pde-based fast local level set method, Journal of Computational Physics 155 (2) (1999) 410–438.

[4] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, Journal of Computational Physics 169 (2) (2001) 463–502.

[5] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, Journal of Computational Physics 183 (1) (2002) 83–116.

[6] J.-J. Xu, H.-K. Zhao, An eulerian formulation for solving partial differential equations along a moving interface, Journal of Scientific Computing 19 (1) (2003) 573–594.

[7] B. Lee, M. Kang, Full 3d simulations of two-phase core–annular flow in horizontal pipe using level set method, Journal of Scientific Computing 66 (3) (2016) 1025–1051.

[8] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1) (1992) 25–37.

[9] J. Glimm, J. W. Grove, X. L. Li, K. M. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM Journal on Scientific Computing 19 (3) (1998) 703–727.

[10] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, Y. J. Tauber, W.and Jan, A front-tracking method for the computations of multiphase flow, Journal of Computational Physics 169 (2) (2001) 708–759.

[11] X. Li, X. He, X. Liu, J. J. Zhang, B. Liu, E. Wu, Multiphase interface tracking with fast semi-lagrangian contouring, IEEE Transactions on Visualization and Computer Graphics 22 (8) (2016) 1973–1986.

[12] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, Journal of Computational Physics 180 (2) (2002) 427–470.

[13] Z.-G. Feng, E. E. Michaelides., Proteus: a direct forcing method in the simulations of particulate flows, Journal of Computational Physics 202 (1) (2005) 20–51.

[14] J. Hua, J. F. Stene, P. Lin, Numerical simulation of 3d bubbles rising in viscous liquids using a front tracking method, Journal of Computational Physics 227 (6) (2008) 3358–3382.

[15] H. Terashima, G. Tryggvason, A front-tracking/ghost-fluid method for fluid interfaces in compressible flows, Journal of Computational Physics 228 (11) (2009) 4012–4037.

[16] F. Da, C. Batty, E. Grinspun, Multimaterial mesh-based surface tracking, ACM Trans. Graph 33 (4) (2014) 112:1–112:11.

[17] M. K. Misztal, K. Erleben, A. Bargteil, J. Fursund, B. B. Christensen, J. A. Bærentzen, R. Bridson, Multiphase flow of immiscible fluids on unstructured moving meshes, IEEE Transactions on Visualization and Computer Graphics 20 (1) (2014) 4–16.

[18] X. Jiao, Face offsetting: A unified approach for explicit moving interfaces, Journal of Computational Physics 220 (2) (2007) 612–625.

[19] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted eno schemes, Journal of Computational Physics 126 (1) (1996) 202–228.

[20] C.-W. Shu, High order eno and weno schemes for computational fluid dynamics, High-order Methods for Computational Physics (1999) 439–582.

[21] X.-D. Liu, S. Osher, T. Chan., Weighted essentially non-oscillatory schemes, Journal of Computational Physics 115 (1) (1994) 200–212.

[22] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, iii, Journal of Computational Physics 71 (2) (1987) 231–303.

[23] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, Journal of Computational Physics 150 (1) (1999) 97–127.

[24] Y.-T. Zhang, C.-W. Shu, Third order weno scheme on three dimensional tetrahedral meshes, Communications in Computational Physics 5 (2-4) (2009) 836–848.

[25] A. K. Henrick, T. D. Aslam, J. M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, Journal of Computational Physics 207 (2) (2005) 542–567.

[26] R. Borges, M. Carmona, B. Costa, W. S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, Journal of Computational Physics 227 (6) (2008) 3191–3211.

[27] Y. Ha, C. H. Kim, Y. J. Lee, J. Yoon, An improved weighted essentially non-oscillatory scheme with a new smoothness indicator, Journal of Computational Physics 232 (1) (2013) 68–86.

[28] R. J. LeVeque, Numerical methods for conservation laws, Birkhauser Verlag (1992) 1–214.

[29] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (2) (1988) 439–471.

[30] S. Gottlieb, C.-W. Shu, Total variation diminishing runge-kutta schemes, Mathematics of Computation of the American Mathematical Society 67 (221) (1998) 73–85.

[31] M. Sussman, P. Smereka, S. Osher., A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1) (1994) 146–159.

[32] X. Jiao, Volume and feature preservation in surface mesh optimization, Proceedings of the 15th International Meshing Roundtable (2006) 359–373.

[33] R. Wang, H. Feng, , R. J. Spiteri., Observations on the fifth-order weno method with non-uniform meshes, Applied Mathematics and Computation 196 (1) (2008) 433–447.

[34] T. Brochu, R. Bridson, Robust topological operations for dynamic explicit surfaces, SIAM Journal on Scientific Computing 31 (4) (2009) 2472–2493.

[35] J. Teran, N. Molino, R. Fedkiw, R. Bridson, Adaptive physics based tetrahedral mesh generation using level sets, Engineering with Computers 21 (1) (2005) 2–18.

[36] V. Surazhsky, C. Gotsman., Explicit surface remeshing, Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (2003) 20–30.

[37] L. A. Freitag, C. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, International Journal for Numerical Methods in Engineering 40 (21) (1997) 3979–4002.

[38] T. Zhou, K. Shimada., An angle-based approach to two-dimensional mesh smoothing, IMR (2000) 373–384.

[39] X. Jiao, M. T. Heath, Feature detection for surface meshes, Proceedings of 8th International Conference on Numerical Grid Generation in Computational Field Simulations (2002) 705–714.

[40] X. Jiao, P. J. Alexander., Parallel feature-preserving mesh smoothing, International Conference on Computational Science and Its Applications (2005) 1180–1189.

국문초록

본 학위논문에서는 표면에 수직 방향의 속력이 주어졌을 때 고차 정확도로서 표면을 추적하기 위한 방법을 제시하였다. 우리가 제안하는 모델은 잘 알려진 라그랑지안 표면 추적 기법을 불연속 점 포착을 위한 고차 정확도 보간법과 결합한 것이다. 제안된 방법은 기존 방법에 비해 높은 정확도로 표면을 추적할 뿐 아니라, 매개 변수들에 대한 의존도가 낮고 표면의 국부적 모양들을 정확히 찾아내어 안정적으로 표면을 추적할 수 있다. 일차원 곡선 위에서 개발된 모델은 삼각형 메쉬 위에서의 고차 정확도 보간법을 이용하여 이차원 곡면 상으로도 자연스럽게 확장될 수 있다.

주요어휘: 라그랑지안 표면 추적, 국부적 모양 검출, 파형 운동, 페이스 오프셋 방법, 위노 보간법, 페이스 오프셋-위노 방법
학번: 2012-20248