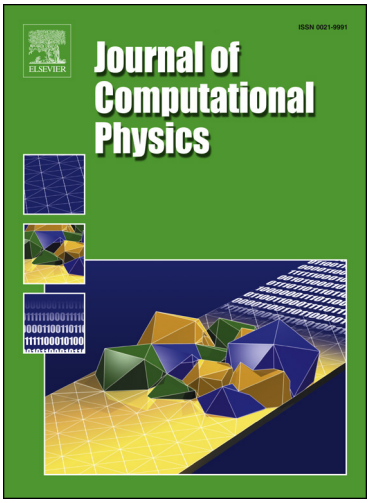


An absolutely convergent fixed-point fast sweeping WENO method on triangular meshes for steady state of hyperbolic conservation laws

Liang Li, Jun Zhu and Yong-Tao Zhang

PII: S0021-9991(24)00464-9
DOI: https://doi.org/10.1016/j.jcp.2024.113215
Reference: YJCPH 113215

To appear in: *Journal of Computational Physics*

Received date: 28 April 2023
Revised date: 27 March 2024
Accepted date: 18 June 2024

Please cite this article as: L. Li, J. Zhu and Y.-T. Zhang, An absolutely convergent fixed-point fast sweeping WENO method on triangular meshes for steady state of hyperbolic conservation laws, *Journal of Computational Physics*, 113215, doi: https://doi.org/10.1016/j.jcp.2024.113215.

Highlights

- High order accuracy fast sweeping method on unstructured meshes is developed for the first time to deal with problems defined on complex domains.
- Multiple reference points on the computational domain are introduced to order all the cells and form alternating sweeping directions on unstructured meshes.
- The local solver is based on a fifth-order finite volume unstructured WENO scheme with unequal-sized sub-stencils for the absolute convergence of the iterations.
- The new high order fast sweeping WENO method on unstructured meshes drives the residue of the iterations to converge to round off errors / machine zero for all benchmark problems tested.
- By comparing with the other popular schemes, the proposed new scheme is very efficient.

An absolutely convergent fixed-point fast sweeping WENO method on triangular meshes for steady state of hyperbolic conservation laws

Liang Li[1], Jun Zhu[2], Yong-Tao Zhang[3,*]

Abstract

High order accuracy fast sweeping methods have been developed in the literature to efficiently solve steady state solutions of hyperbolic partial differential equations (PDEs) on rectangular meshes, while they were not available yet on unstructured meshes. In this paper, we extend high order accuracy fast sweeping methods to unstructured triangular meshes by applying fixed-point iterative sweeping techniques to a fifth-order finite volume unstructured WENO scheme, for solving steady state solutions of hyperbolic conservation laws. Similar as other fast sweeping methods, fixed-point fast sweeping methods use the Gauss-Seidel iterations and alternating sweeping strategy to cover characteristics of hyperbolic PDEs in each sweeping order to achieve fast convergence rate to steady state solutions. An advantage of fixed-point fast sweeping methods which distinguishes them from other fast sweeping methods is that they are explicit and do not require inverse operation of nonlinear local systems. This also provides certain convenience in designing high order fast sweeping methods on unstructured meshes. As in the first order fast sweeping methods on triangular meshes, we introduce multiple reference points to determine alternating sweeping directions on unstructured meshes. All the cells on the mesh are ordered according to their centroids' distances to those reference points, and the resulted orderings provide sweeping directions for iterations. To make the residue of the fast sweeping iterations converge to machine zero / round off errors, we follow the approach in our early work of developing the absolutely convergent fixed-point fast sweeping WENO methods on rectangular meshes, and adopt high order WENO scheme with unequal-sized sub-stencils, specifically here a fifth-order finite volume unstructured WENO scheme for spatial discretization. Extensive numerical experiments, including problems with complex domain geometries, are performed to show the accuracy, computational efficiency, and absolute convergence of the presented fifth-order fast sweeping scheme on triangular meshes. Furthermore, the proposed method is compared with the forward Euler time marching method and the popular third order total variation diminishing Rung-Kutta (TVD-RK3) time-marching method for steady state computations. Numerical examples show that the developed fixed-point fast sweeping WENO method is the most efficient scheme among them, and especially it can save up to 70% CPU time costs than TVD-RK3 in order to converge to steady state solutions.

Key Words: Fixed-point fast sweeping methods, triangular mesh, high order WENO schemes, unequal-sized sub-stencils, steady state, hyperbolic conservation laws.

[1]School of Mathematics and Statistics, Huang Huai University, Zhumadian, Henan 463000, P.R. China. E-mail: liliangnuaa@163.com.

[2]State Key Laboratory of Mechanics and Control of Mechanical Structures and Key Laboratory of Mathematical Modelling and High Performance Computing of Air Vehicles (NUAA), MIIT, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China. E-mail: zhujun@nuaa.edu.cn. Research was supported by NSFC grant 11872210 and Grant No. MCMS-I-0120G01.

[3]Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: yzhang10@nd.edu

[*]Corresponding author.

1 Introduction

Steady state problems of hyperbolic PDEs, such as hyperbolic conservation laws and Hamilton-Jacobi equations, are common mathematical models arising in many applications, e.g. compressible fluid mechanics, optimal control, geometric optics, image processing and computer vision, etc. One of the most important properties of these hyperbolic type boundary value problems is that their solution information propagates along characteristics starting from the boundary. Fast sweeping methods, a class of iterative methods originally developed in the literature to solve static Hamilton-Jacobi equations (see e.g. [57, 34, 35, 15]), take advantage of such property to efficiently solve steady state problems of hyperbolic PDEs. The methods use alternating sweeping strategy to cover a family of characteristics in a certain direction simultaneously in each sweeping order. Combined with the Gauss-Seidel iterations, these methods can achieve a fast convergence speed. High order accuracy fast sweeping methods have been developed and studied extensively on rectangular meshes. In [56], high order weighted essentially non-oscillatory (WENO) fast sweeping schemes for solving static Hamilton-Jacobi equations on rectangular meshes were developed, where an explicit strategy in the iterative schemes was designed to avoid directly solving very complicated local nonlinear equations derived from high order WENO discretizations. The methods were combined with accurate boundary treatment techniques such as the inverse Lax-Wendroff methods in [49]. In an implicit way, fast sweeping methods were also applied in discontinuous Galerkin (DG) methods to efficiently solve static Eikonal equations [26, 46, 52]. This kind of implicit high order fast sweeping methods are very efficient and often have linear computational complexity as the first order fast sweeping methods [57, 34, 35]); however the algorithms are much more complicated than explicit methods such as the WENO fast sweeping methods [56], which makes these implicit high order fast sweeping methods difficult to be applied in solving more complicated hyperbolic PDEs.

Although high order accuracy fast sweeping methods were well developed on rectangular meshes, they have not been explored much on unstructured meshes to solve steady state problems of general hyperbolic PDEs. A key factor in developing a high order accuracy fast sweeping method on unstructured meshes, which is efficient and relatively easy to be implemented, is to design an approach to effectively incorporate an unstructured nonlinearly stable scheme into the fast sweeping iteration framework to accelerate the convergence to steady state solutions. In this paper, we develop high order accuracy fast sweeping methods on unstructured triangular meshes by applying fixed-point iterative sweeping techniques to a fifth-order finite volume unstructured WENO scheme, for solving steady state solutions of hyperbolic conservation laws. The fixed-point fast sweeping methods were first designed in [55] for solving static Hamilton-Jacobi equations on rectangular meshes. They can be considered as a generalization of the explicit high order fast sweeping methods in [56]. As other fast sweeping methods, the fixed-point fast sweeping methods use the Gauss-Seidel iterations and alternating sweeping strategy to cover characteristics of hyperbolic PDEs in each sweeping order to achieve fast convergence rate to steady state solutions. However, this kind of fast sweeping methods have nice properties such as that they are fully explicit and do not involve any inverse operation of nonlinear local systems, and they can be easily adopted to solve complex hyperbolic systems with any monotone numerical fluxes and high order nonlinearly stable schemes. For example, the fixed-point fast sweeping methods were applied to sparse-grid WENO schemes for efficiently solving multidimensional Eikonal equations in [33]; in [47, 27, 28], high order accuracy fixed-point fast sweeping WENO methods for efficiently solving steady state problems of compressible flows modeled by nonlinear hyperbolic conservation laws were developed. As shown in [27], it is interesting to find that the special case of using the parameter $\gamma = 1$ and the Lax-Friedrichs flux in the third order fixed-point fast sweeping schemes is equivalent to the third order Lax-Friedrichs fast sweeping methods in [56, 8]. Here, the "explicit" property of fixed-point fast

sweeping methods also provides convenience for us to develop high order fast sweeping methods on unstructured meshes.

As in the fast sweeping methods on rectangular meshes, there are also two important components in designing fast sweeping methods on unstructured meshes: (1) a nonlinearly stable scheme to discretize the hyperbolic PDEs, which is often called a "local solver"; (2) systematic orderings of all grid points / cells, which can cover all directions of the characteristics. The local solver used in this paper is in the class of WENO schemes. WENO schemes are popular high order accuracy numerical methods for spatial discretizations of nonlinear hyperbolic conservation laws, e.g., see [5, 22, 29, 40, 42]. They have the advantage of attaining uniform high order accuracy in smooth regions of the solution while maintaining sharp and essentially non-oscillatory transitions of discontinuities. On unstructured meshes, WENO schemes were constructed in e.g. [13, 14, 17, 21, 23, 44, 53, 54]. These unstructured WENO schemes follow the idea in original structured WENO schemes [22, 41], i.e., the sub-stencils have comparable size and are smaller than the big stencil of the scheme. They can be classified to two types according to their differences in WENO reconstructions on unstructured meshes, as that defined in [54, 30]. Specifically, the major difference between two types of these schemes is the different method to construct sub-stencils and find linear weights. The first type (type I) reconstruction [13, 14, 17, 44] has an order of accuracy not higher than that of the reconstruction on each sub-stencil. This is similar as essentially non-oscillatory (ENO) schemes [20, 43]. The nonlinear weights in type I WENO reconstructions do not contribute towards the increase of the order of accuracy, but they are designed purely for the purpose of nonlinear stability, or to avoid spurious oscillations. Because type I WENO schemes just need to choose the linear weights as arbitrary positive numbers for better linear stability (e.g. the centered sub-stencil is assigned a larger linear weight than the others), they are easier to construct than the type II WENO schemes discussed in the following. The second type (type II) consists of WENO schemes whose order of accuracy is higher than that of the reconstruction on each sub-stencil (see e.g. [21, 23, 53, 54]). A crucial step in building a type II WENO scheme on unstructured meshes is to construct lower order polynomials whose weighted average will give the same result as the high order reconstruction at each Gaussian quadrature point for the flux integral on the element boundary. This step is actually the most difficult step in designing a robust type II high order WENO schemes on unstructured meshes, since we can not guarantee the quality of the unstructured meshes when the domain geometry is very complicated. Especially, when the spatial domain has higher dimensions (e.g., three dimensional problems) and complex geometry, the quality of the unstructured meshes is hard to control. Distorted local mesh geometries can be easily generated. The local linear system for finding linear weights could have very large condition number or is even singular at the places where mesh quality is bad. This is the reason why type II WENO schemes are more difficult to construct than type I WENO schemes on unstructured meshes. However they have a much more compact stencil than type I WENO schemes of the same accuracy, which is an advantage in applications. In [30], the approaches of type II and type I WENO schemes are combined to achieve a robust unstructured finite volume WENO reconstruction, and the appearance of negative and very large linear weights are avoided no matter how bad the quality of the unstructured meshes is. Recently, high order WENO schemes with unequal-sized sub-stencils were developed on unstructured meshes [59, 61]. As the type I WENO schemes, the linear weights of these WENO schemes with unequal-sized sub-stencils can be chosen as arbitrary positive numbers as long as their sum equals 1, hence they are easier to construct than the type II WENO schemes on unstructured meshes. Also they have the same big stencil as the type II WENO schemes, hence they are as compact as the type II WENO schemes. Another advantage of WENO schemes with unequal-sized sub-stencils is on solving steady state problems. Studies on high order WENO schemes on unequal-sized sub-stencils reveal that they improve the convergence

of high order WENO schemes with equal-sized sub-stencils to steady state solutions so that the residue of time-marching iterations settles down to machine zero / round off errors [60, 61]. The property of an iterative scheme that the residue of its iterations can settle down to machine zero or the round off error level in a finite number of iterations is called "absolute convergence" in [27]. In this paper, to make the residue of the fast sweeping iterations converges to machine zero / round off errors, we follow the approach in our early work of developing the absolutely convergent fixed-point fast sweeping WENO methods on rectangular meshes [27], and adopt high order WENO scheme with unequal-sized sub-stencils, specifically here a fifth-order finite volume unstructured WENO discretization [61] as the local solver of the fast sweeping method.

The efficient convergence of fast sweeping methods is due to the fact that all directions of characteristics can be divided into a finite number of groups, and any characteristic can be decomposed into a finite number of pieces that belong to one of the above groups, then systematic orderings can be designed to follow the causality of each group of directions simultaneously. Orderings of all grid points or cells on a rectangular mesh are natural. For example in a two-dimensional (2-D) problem, all directions of the characteristics can be divided into four groups, namely, up-right, up-left, down-left, and down-right. All grid points or cells are ordered by using their indexes to provide four orderings to cover all these four groups of characteristics. However, such natural orderings do not exist on unstructured meshes or triangular meshes considered here. In this paper, we apply the approach in the first order fast sweeping methods on triangular meshes [34] to the high order fast sweeping WENO method developed here. Specifically, multiple reference points are introduced to determine alternating sweeping directions on unstructured triangular meshes. All the cells on the mesh are ordered according to their centroids' distances to those reference points, and the resulted orderings provide sweeping directions for the fixed-point iterations.

To compute steady state of hyperbolic conservation laws, the forward Euler time marching method is preferred rather than a Runge-Kutta method, because it is very simple with a single-step and one stage, and time direction accuracy has no effects on the numerical accuracy of steady state solutions. However, a high order linear spatial scheme (e.g., a fifth order linear scheme) with the forward Euler time marching is linearly unstable. Nonlinearly stable spatial discretizations such as WENO schemes can help to stabilize the computation by the forward Euler method, but often a very small CFL number and large amount of iterations are required to converge to steady state, as shown in [47]. Hence, for both linear and nonlinear stability of high order schemes in solving steady state problems of hyperbolic conservation laws, a common approach is to use a total variation diminishing Runge-Kutta method, e.g. the popular third order scheme (TVD-RK3) [19, 43], to time-march numerical solution to steady state. In [27, 47], it was shown that on rectangular meshes, the fast sweeping technique can largely improve the stability of high order WENO schemes with the forward Euler time marching, and the fast sweeping WENO schemes converge to steady state solution much faster than the popular TVD-RK3 time-marching approach. In this paper, we will draw the similar conclusion for the developed absolutely convergent fixed-point fast sweeping WENO method on unstructured triangular meshes. Numerical experiments show that the proposed fast sweeping scheme is more efficient than both the forward Euler time marching method and the TVD-RK3 scheme, and it converges under comparable CFL numbers as the TVD-RK3 scheme for all examples, while the direct forward Euler time marching method needs a smaller CFL number to converge in general.

The organization of the paper is as follows. The detailed description of the new absolutely convergent fixed-point fast sweeping WENO method on unstructured triangular meshes is provided in Section 2. In Section 3, we perform numerical experiments to study the proposed method, and carry out comparisons of different methods. Extensive numerical examples, including the shock reflection, supersonic flow past a circular cylinder, and the problems of supersonic and subsonic

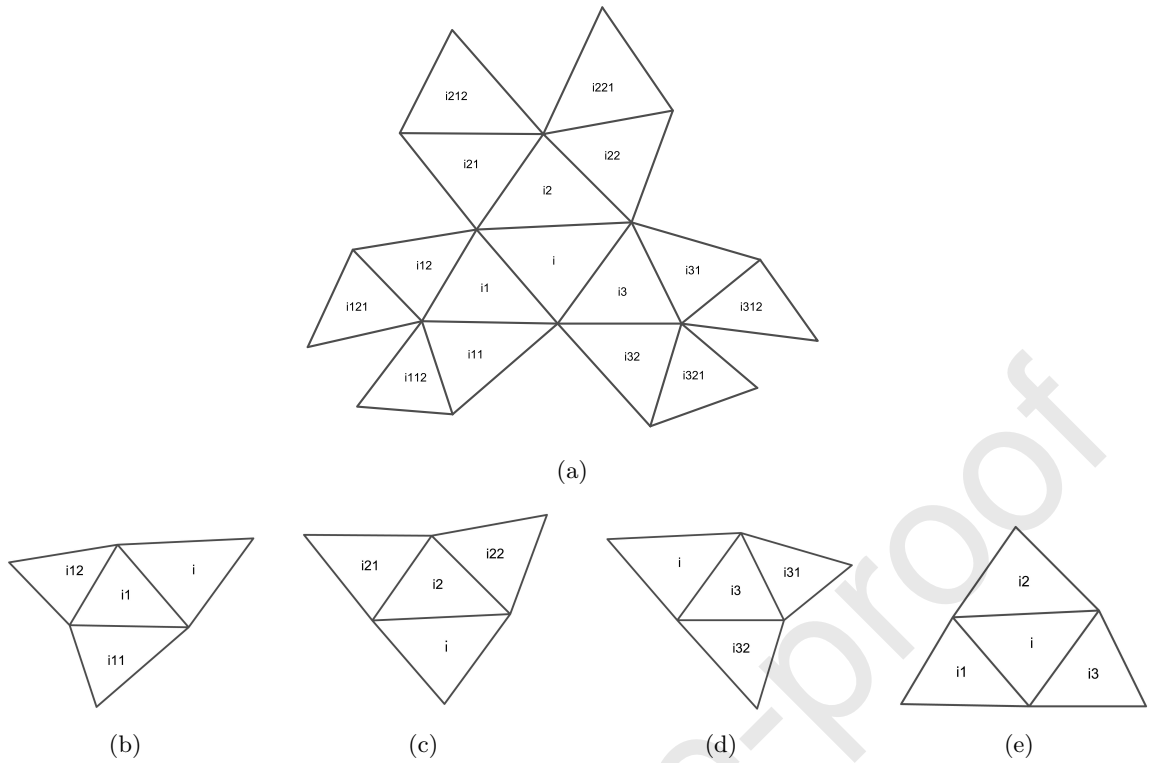


Figure 1: Unequal-sized sub-stencils of a target cell $\triangle_i$ for the fifth order finite volume unstructured WENO local solver. (a) sub-stencil T_1; (b) sub-stencil T_2; (c) sub-stencil T_3, (d) sub-stencil T_4; (e) sub-stencil T_5.

flows past an airfoil which have complex domain geometries, etc., are solved to show the accuracy, computational efficiency, and absolute convergence of the presented fifth-order fast sweeping WENO scheme on unstructured triangular meshes. Concluding remarks are given in Section 4.

2 Description of the numerical methods

We consider steady state problems of hyperbolic conservation laws with appropriate boundary conditions. The two dimensional (2D) case has the following general form

$$\mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = \mathbf{R}, \tag{2.1}$$

where $\mathbf{u}$ is the vector of the unknown conservative variables, $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ are the vectors of flux functions, and $\mathbf{R}(x, y)$ is the source term. For example, the steady Euler system of equations in compressible fluid dynamics has that $\mathbf{u} = (\rho, \rho u, \rho v, E)^T$, $\mathbf{f}(\mathbf{u}) = (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T$, and $\mathbf{g}(\mathbf{u}) = (\rho v, \rho uv, \rho v^2 + p, v(E + p))^T$. Here ρ is the density of fluid, $(u, v)^T$ is the velocity vector, p is the pressure, and $E = \frac{p}{\gamma'-1} + \frac{1}{2}\rho(u^2 + v^2)$ is total energy where the constant $\gamma' = 1.4$ for the case of air. A spatial discretization of (2.1) by a high order WENO scheme leads to a large nonlinear system of algebraic equations with the size determined by the number of spatial triangular cells.

2.1 The local solver

In this section, we describe the fifth-order finite volume WENO scheme with unequal-sized sub-stencils on triangular meshes for discretization of (2.1), which is the local solver of the developed fast sweeping method. The computational domain is partitioned by an unstructured triangular mesh with computational cells $\{\triangle_i, i = 1, \cdots, M\}$, and M is the total number of cells in the mesh. We integrate (2.1) over a target cell $\triangle_i$ to obtain

$$\frac{1}{|\triangle_i|}\int_{\partial\triangle_i} \mathbf{F}\cdot\overrightarrow{\mathbf{n}}ds = \overline{\mathbf{R}}_i, \tag{2.2}$$

in which $\overline{\mathbf{R}}_i = \frac{1}{|\triangle_i|}\int_{\triangle_i}\mathbf{R}(x,y)dxdy$, $\mathbf{F} = (\mathbf{f},\mathbf{g})$, $\partial\triangle_i$ is the boundary of the target cell $\triangle_i$, $|\triangle_i|$ is the area of $\triangle_i$, and $\overrightarrow{\mathbf{n}}$ is the outward unit normal to the boundary of $\triangle_i$. The line integrals in (2.2) are discretized by a three-point Gaussian quadrature formula [21] on each edge of the cell $\triangle_i$

$$\int_{\partial\triangle_i}\mathbf{F}\cdot\overrightarrow{n}ds \approx \sum_{k=1}^{3}|\partial\triangle_{i_k}|\sum_{m=1}^{3}\sigma_m\mathbf{F}(\mathbf{u}(x_{G_{k_m}},y_{G_{k_m}}))\cdot\overrightarrow{\mathbf{n}}_k, \tag{2.3}$$

where $\{|\partial\triangle_{i_k}|\}_{k=1}^3$ are the lengths of the edges, $\{\sigma_m\}_{m=1}^3$ are the quadrature weights, $\{(x_{G_{k_m}},y_{G_{k_m}})\}_{k,m=1}^3$ are the Gaussian quadrature points, $\{\overrightarrow{\mathbf{n}}_k\}_{k=1}^3$ are the outward unit normals at the quadrature points. For linear stability, $\mathbf{F}(\mathbf{u}(x_{G_{k_m}},y_{G_{k_m}}))\cdot\overrightarrow{\mathbf{n}}_k, m = 1,\cdots,3; k = 1,\cdots,3$ are approximated by monotone numerical fluxes such as the Lax-Friedrichs flux

$$\begin{aligned}\mathbf{F}(\mathbf{u}(x_{G_{k_m}},y_{G_{k_m}}))\cdot\overrightarrow{\mathbf{n}}_k \approx \frac{1}{2}\big[&(\mathbf{F}(\mathbf{u}^+(x_{G_{k_m}},y_{G_{k_m}})) + \mathbf{F}(\mathbf{u}^-(x_{G_{k_m}},y_{G_{k_m}})))\cdot\overrightarrow{\mathbf{n}}_k\\ &-\alpha(\mathbf{u}^+(x_{G_{k_m}},y_{G_{k_m}}) - \mathbf{u}^-(x_{G_{k_m}},y_{G_{k_m}}))\big].\end{aligned} \tag{2.4}$$

Here α is taken as an upper bound for the eigenvalues of the flux Jacobian in the $\overrightarrow{\mathbf{n}}_k$ direction, and $\mathbf{u}^-$ and $\mathbf{u}^+$ are the reconstructed values of the numerical solution $\mathbf{u}$ inside and outside of the target cell $\triangle_i$ at different Gaussian quadrature points, based on the cell average values of $\mathbf{u}$ on cells of the computational stencil. We use the fifth-order WENO method [61] to reconstruct the function values of $\mathbf{u}(x,y)$ at different Gaussian quadrature points $(x_{G_{k_m}},y_{G_{k_m}})$ on the cell boundaries. Reconstruction on the target cell $\triangle_i$ provides values for $\mathbf{u}^-$ in (2.4), with detailed algorithm given in the following. Values for $\mathbf{u}^+$ in (2.4) are provided by the reconstruction on one of the neighboring cells of $\triangle_i$.

Reconstruction algorithm:

Step 1. Select a big central spatial stencil $T_1 = \big\{\triangle_i, \triangle_{i1}, \triangle_{i2}, \triangle_{i3}, \triangle_{i11}, \triangle_{i12}, \triangle_{i21}, \triangle_{i22}, \triangle_{i31}, \triangle_{i32}, \triangle_{i112}, \triangle_{i121}, \triangle_{i212}, \triangle_{i221}, \triangle_{i312}, \triangle_{i321}\big\}$ (see Figure 1), which are formed by several layers of neighbor cells of the target cell $\triangle_i$. Based on the cell average values of $\mathbf{u}$ on cells of T_1, we construct a quartic polynomial $\mathbf{p}_1(x,y)$ on T_1 to obtain a fifth-order approximation of the conservative variable $\mathbf{u}$. Since in general the number of cells on T_1 are more than the degree of freedom of the quartic polynomial, the least-square method is used to find $\mathbf{p}_1(x,y)$ as in [21], namely, it is required that $\mathbf{p}_1(x,y)$ has the same cell average of $\mathbf{u}$ on the target cell $\triangle_i$ and matches the cell averages of $\mathbf{u}$ on the other cells of the stencil T_1 in the least-square way:

$$\frac{1}{|\triangle_i|}\int_{\triangle_i}\mathbf{p}_1(x,y)dxdy = \overline{\mathbf{u}}_i, \quad \mathbf{p}_1(x,y) = \arg\min\sum_{ll\in W_1}\left(\frac{1}{|\triangle_{ll}|}\int_{\triangle_{ll}}\mathbf{p}(x,y)dxdy - \overline{\mathbf{u}}_{ll}\right)^2, \tag{2.5}$$

$W_1 = \{i1, i2, i3, i11, i12, i21, i22, i31, i32, i112, i121, i212, i221, i312, i321\}$.

Note that if the number of cells located inside the stencil T_1 is less than that is required for reconstructing the polynomial because some of the cells coincide with each other, neighboring cells in the next layer of the mesh will be added to the stencil to provide enough number of cells for the reconstruction.

Step 2. Form three sectorial sub-stencils $T_2 = \{\triangle_i, \triangle_{i1}, \triangle_{i11}, \triangle_{i12}\}, T_3 = \{\triangle_i, \triangle_{i2}, \triangle_{i21}, \triangle_{i22}\}$, $T_4 = \{\triangle_i, \triangle_{i3}, \triangle_{i31}, \triangle_{i32}\}$ and one central sub-stencil $T_5 = \{\triangle_i, \triangle_{i1}, \triangle_{i2}, \triangle_{i3}\}$ (see Figure 1). Note that in forming these sectorial sub-stencils T_2, T_3, T_4, we use straight lines to connect the barycenter of the target cell $\triangle_i$ with its three vertices and split the plane into three sectors. Every sectorial sub-stencil consists of the target cell $\triangle_i$ and its neighboring cells whose barycenters lie in the same sector. Such kind of distributions of sub-stencils are important to obtain the non-oscillatory performance for non-smooth solutions, as shown in [30]. Then four linear polynomials $\mathbf{p}_m(x, y), m = 2, \cdots, 5$ on such sub-stencils are constructed to obtain second-order approximations of the conservative variable $\mathbf{u}$. Again, the least-square method is used to find $\mathbf{p}_m(x, y)$ by requiring that $\mathbf{p}_m(x, y)$ has the same cell average of $\mathbf{u}$ on the target cell $\triangle_i$ and matches the cell averages of $\mathbf{u}$ on the other cells of the sub-stencil T_m in the least-square way:

$$\frac{1}{|\triangle_i|}\int_{\triangle_i} \mathbf{p}_m(x,y)dxdy = \overline{\mathbf{u}}_i, \quad \mathbf{p}_m(x,y) = \arg\min \sum_{ll\in W_m}\left(\frac{1}{|\triangle_{ll}|}\int_{\triangle_{ll}} \mathbf{p}(x,y)dxdy - \overline{\mathbf{u}}_{ll}\right)^2, \tag{2.6}$$
$$m = 2, \cdots, 5; W_2 = \{i1, i11, i12\}, W_3 = \{i2, i21, i22\}, W_4 = \{i3, i31, i32\}, W_5 = \{i1, i2, i3\}.$$

Step 3. Define the linear weights. We emphasize that different from the previous unstructured WENO schemes [21, 54], the WENO reconstruction with unequal-sized sub-stencils here uses the big stencil T_1 as one of the sub-stencils and the other sub-stencils T_2, T_3, T_4, T_5 are much smaller, while in the WENO schemes [21, 54], every sub-stencil is only part of the big stencil and their union is the big stencil. To use the reconstruction $\mathbf{p}_1(x, y)$ on the sub-stencil T_1 for the final high order accuracy, we apply the similar ideas for the central WENO schemes [24, 25] and rewrite $\mathbf{p}_1(x, y)$ as

$$\mathbf{p}_1(x,y) = \gamma_1\left(\frac{1}{\gamma_1}\mathbf{p}_1(x,y) - \sum_{m=2}^{5}\frac{\gamma_m}{\gamma_1}\mathbf{p}_m(x,y)\right) + \sum_{m=2}^{5}\gamma_m\mathbf{p}_m(x,y) \tag{2.7}$$

with $\sum_{m=1}^{5}\gamma_m = 1$ and $\gamma_1 \neq 0$. In these expressions, γ_m for $m = 1, \cdots, 5$ are the linear weights. Following the practice in [13, 30, 58, 62], we take the linear weights as $\gamma_1 = 0.96$ and $\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = 0.01$.

Step 4. Compute the smoothness indicators β_k, which measure how smooth the functions $\mathbf{p}_k$ for $k = 1, \cdots, 5$ are in the target cell $\triangle_i$. We use the same recipe for the smoothness indicators as that in [21, 22]:

$$\beta_k = \sum_{|\alpha|=1}^{r}\int_{\triangle_i} |\triangle_i|^{|\alpha|-1}(\mathbf{D}^\alpha \mathbf{p}_k(x,y))^2 dxdy, \quad k = 1, \cdots, 5, \tag{2.8}$$

where $\mathbf{D}$ is the differential operator and α is the multi-index for derivative orders. $r = 4$ for $l = 1$, and $r = 1$ for $l = 2, \cdots, 5$.

Step 5. Compute the nonlinear weights based on the linear weights and the smoothness indicators. We adopt the WENO-Z type nonlinear weights as specified in [5, 7, 12], the nonlinear weights are defined as

$$\omega_k = \frac{\overline{\omega}_k}{\sum_{k=1}^{5}\overline{\omega}_k}, \quad \overline{\omega}_k = \gamma_k\left(1 + \frac{\tau}{\varepsilon + \beta_k}\right), \quad k = 1, \cdots, 5. \tag{2.9}$$

According to [58, 59], we define $\tau = \left(\frac{|\beta_1-\beta_2|+|\beta_1-\beta_3|+|\beta_1-\beta_4|+|\beta_1-\beta_5|}{4}\right)^2$, and take $\varepsilon = 10^{-6}$.

Step 6. The final reconstruction polynomial for the approximation of $u(x, y)$ at any points of the target cell $\triangle_i$ is given as

$$\mathbf{u}(x,y) \approx \omega_1 \left(\frac{1}{\gamma_1}\mathbf{p}_1(x,y) - \sum_{m=2}^{5} \frac{\gamma_m}{\gamma_1}\mathbf{p}_m(x,y) \right) + \sum_{m=2}^{5} \omega_m \mathbf{p}_m(x,y). \tag{2.10}$$

Remark. Note that other high order WENO methods on unstructured meshes can also be used as the local solver of the fast sweeping method developed here. For example, an efficient class of WENO schemes with adaptive order, called WENO-AO schemes, were developed first in [3] for structured meshes, then in [2] for unstructured meshes. The WENO-AO schemes and the WENO schemes with unequal-sized sub-stencils [58, 61] share the similar idea of using a large stencil and a telescoping set of smaller stencils for high order WENO reconstructions. They are very flexible in choosing linear weights, hence are simpler to construct than the type II WENO schemes (e.g. [21, 54]) on unstructured meshes. Furthermore, in [2] an efficient approach was developed to simplify the finite volume reconstruction by using a favorable Taylor series basis with an extension of the Parallel Axis Theorem, which makes the resulting method only require the solution of a smaller least squares problem on each stencil, instead of solving a constrained least squares problem. It is an interesting topic and will be explored in the future research to apply these efficient WENO-AO schemes to the fast sweeping method proposed in this paper for solving steady state problems.

2.2 Absolutely convergent fixed-point fast sweeping WENO scheme

After the spatial discretization of the equation (2.1) by the finite volume scheme described in the last section, we have

$$\begin{aligned} 0 = -\frac{1}{|\triangle_i|} \sum_{k=1}^{3} |\partial \triangle_{i_k}| \sum_{m=1}^{3} \sigma_m \frac{1}{2} \big[(\mathbf{F}(\mathbf{u}^+(x_{G_{k_m}}, y_{G_{k_m}})) + \mathbf{F}(\mathbf{u}^-(x_{G_{k_m}}, y_{G_{k_m}}))) \cdot \overrightarrow{\mathbf{n}}_k \\ - \alpha(\mathbf{u}^+(x_{G_{k_m}}, y_{G_{k_m}}) - \mathbf{u}^-(x_{G_{k_m}}, y_{G_{k_m}})) \big] + \overline{\mathbf{R}}_i, \quad i = 1, \cdots, M, \end{aligned} \tag{2.11}$$

where M is the number of triangular cells. The aforementioned fifth-order WENO reconstructions at Gaussian quadrature points $(x_{G_{k_m}}, y_{G_{k_m}})$ on the cell boundaries lead to a nonlinear algebraic system

$$L(\overline{\mathbf{u}}_i, \overline{\mathbf{u}}_{i1}, \cdots, \overline{\mathbf{u}}_{i321}) = 0, \quad i = 1, \cdots, M. \tag{2.12}$$

Note that here $\overline{\mathbf{u}}_i$ is the cell average of the numerical solution of the unknown function $\mathbf{u}$ on the target cell $\triangle_i$. $\overline{\mathbf{u}}_{i1}, \cdots, \overline{\mathbf{u}}_{i321}$ are the cell averages on the WENO scheme's stencil of the target cell $\triangle_i$. L denotes the spatial discretization operator applied to the PDE using this fifth-order finite volume WENO local solver, which is a nonlinear function of the cell averages of the numerical solution on the computational stencil of the WENO scheme. The fixed-point fast sweeping schemes for solving the nonlinear system (2.12) are based on iterative schemes of time marching type for solving steady state problems. Time marching methods for solving steady state problems are essentially Jacobi type fixed-point iterative schemes. The forward Euler (FE) time marching method with time step size Δt_n to solve the nonlinear system (2.12) is the following Jacobi type fixed-point iterative scheme

$$\overline{\mathbf{u}}_i^{n+1} = \overline{\mathbf{u}}_i^n + \triangle t_n L(\overline{\mathbf{u}}_i^n, \overline{\mathbf{u}}_{i1}^n, \cdots, \overline{\mathbf{u}}_{i321}^n), \quad i = 1, \cdots, M, \tag{2.13}$$

where $\overline{\mathbf{u}}_i^{n+1}$ and $\overline{\mathbf{u}}_i^n$ etc. are the numerical solution values at iteration steps $n+1$ and n. The popular TVD-RK3 time marching method to solve the nonlinear system (2.12) is the following

Jacobi type fixed-point iterative scheme:

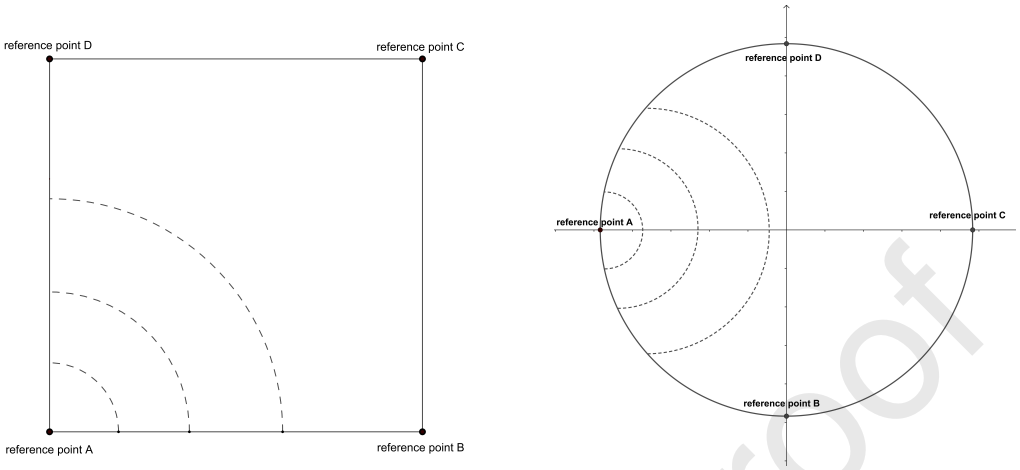
$$\begin{aligned}
\overline{\mathbf{u}}_i^{(1)} &= \overline{\mathbf{u}}_i^n + \triangle t_n L(\overline{\mathbf{u}}_i^n, \overline{\mathbf{u}}_{i1}^n, \cdots, \overline{\mathbf{u}}_{i321}^n), \quad i = 1, \cdots, M, \\
\overline{\mathbf{u}}_i^{(2)} &= \frac{3}{4}\overline{\mathbf{u}}_i^n + \frac{1}{4}\overline{\mathbf{u}}_i^{(1)} + \frac{1}{4}\triangle t_n L(\overline{\mathbf{u}}_i^{(1)}, \overline{\mathbf{u}}_{i1}^{(1)}, \cdots, \overline{u}_{i321}^{(1)}), \quad i = 1, \cdots, M, \\
\overline{\mathbf{u}}_i^{n+1} &= \frac{1}{3}\overline{\mathbf{u}}_i^n + \frac{2}{3}\overline{\mathbf{u}}_i^{(2)} + \frac{2}{3}\triangle t_n L(\overline{\mathbf{u}}_i^{(2)}, \overline{\mathbf{u}}_{i1}^{(2)}, \cdots, \overline{\mathbf{u}}_{i321}^{(2)}), \quad i = 1, \cdots, M.
\end{aligned} \tag{2.14}$$

To obtain faster convergence to steady state solutions of high order unstructured WENO schemes for solving hyperbolic PDEs than the above Jacobi type fixed-point iterations, we apply the fast sweeping techniques so that the important characteristics property of hyperbolic PDEs can be utilized in the iterations, as in the WENO fast sweeping methods on structured meshes [47, 27]. The fast sweeping techniques can be applied to either the FE scheme (2.13) or the TVD-RK3 scheme (2.14). However, as that found in [47, 33], the fast sweeping scheme resulted by applying the fast sweeping techniques to the TVD-RK3 scheme is less efficient than the obtained method by applying the fast sweeping techniques to the FE scheme. Hence here we apply the fast sweeping techniques to the FE scheme (2.13) and obtain the high order FE type fixed-point fast sweeping scheme on unstructured meshes. As shown in the numerical experiments of the following section, the FE fixed-point fast sweeping scheme permits larger CFL numbers than the original FE scheme. In fact it has comparable CFL numbers as that of the TVD-RK3 scheme, and it is much more efficient than the TVD-RK3 scheme to reach steady state of the solutions. The form of the FE fixed-point fast sweeping scheme is:

$$\overline{\mathbf{u}}_i^{n+1} = \overline{\mathbf{u}}_i^n + \triangle t_n L(\overline{\mathbf{u}}_i^n, \overline{\mathbf{u}}_{i1}^*, \cdots, \overline{\mathbf{u}}_{i321}^*), \quad i = k_1, \cdots, k_M. \tag{2.15}$$

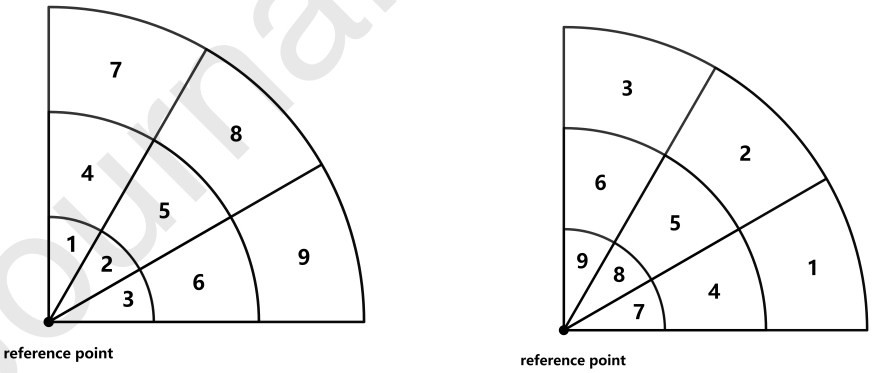
Note that the fast sweeping methods have two essential components, i.e., the Gauss-Seidel philosophy and alternating direction sweeping iterations. The Gauss-Seidel philosophy requires that the newest numerical values of $\overline{\mathbf{u}}$ are used in the finite volume WENO reconstruction stencils as long as they are available. Alternating direction sweepings cover characteristics in different directions. Here the iteration direction in the FE fixed-point fast sweeping scheme (2.15), which is marked as "$i = k_1, \cdots, k_M$", means that the iterations in the scheme (2.15) do not just proceed in only one direction "$i = 1, \cdots, M$" as that in the Jacobi type schemes (2.13) and (2.14), but in the alternating directions repeatedly.

Design of alternating sweeping directions on a rectangular mesh is natural. However, it is not straightforward on unstructured meshes. We use the method in [34] for the first order fast sweeping scheme on triangular meshes to form the sweeping directions of the high order scheme here, and verify that such alternating sweeping directions are still effective for the high order scheme. The method is briefly explained in the following. The alternating sweeping directions on unstructured meshes are determined by ordering all the cells on the mesh according to their centroids' distances to some reference points on the domain. For example, let $\vec{x}$ denote the centroid of a cell and $\vec{x}_{ref}$ denote one of the reference points. We compute the l^2- distance $||\vec{x} - \vec{x}_{ref}||_2$ for each cell of the computational mesh, and sort all the cells according to $||\vec{x} - \vec{x}_{ref}||_2$ in the ascent (or descent) order. Then these two orderings of all the cells provide two sweeping directions. If we use these two sweeping directions as the iteration direction "$i = k_1, \cdots, k_M$" in the fixed-point fast sweeping scheme (2.15), the sweeping directions through all cells on an unstructured mesh are actually approximations of an outgoing (or incoming) spherical wave with respect to the reference point $\vec{x}_{ref}$, which is called the sweeping wavefront in [34]. See Figure 2 for an illustration. This method shares some similarities with the expanding wavefront model in [36, 45]. To explain the idea more clearly, instead of an unstructured mesh, we use a very coarse Cartesian and circular mesh, with



(a) (b)

Figure 2: Examples of reference points for: (a) a rectangular computational domain; (b) a circular computational domain. Sweeping wavefronts (the dashed curves) based on the reference point A are shown.



(a) the forward direction (b) the reverse direction

Figure 3: An example on sweeping number of the zones of a coarse Cartesian and circular mesh, with respect to one of the reference points. (a): in the forward direction (i.e., the ascent order); (b): in the reverse direction (i.e., the descent order).

only a few zones to show the sweeping directions with respect to one of the reference points. Note that in this example, the sweeping directions are exactly the same as the sweeping wavefronts. Figure 3 shows the sweeping number (i.e., the order) of each cell on the mesh, in both the forward direction and the reverse direction. The forward sweeping direction is obtained by sorting all the cells according to their centroids' distances to the reference point in the ascent order, while the reverse sweeping direction is obtained by sorting all the cells according to their centroids' distances to the reference point in the descent order. Note that in this special example, there are cells which are exactly on the same sweeping wavefront, namely, their centroids' distances to the reference point are the same (e.g., the cells marked as "1, 2, 3", and so on, in Figure 3). For these cells, the order of updates of them, which may depend on the sorting method used and its implementation, does not matter and their order does not affect the convergence of the iterations. In the other words, it is fine to update the cells marked as "1, 2, 3" in the order of "2, 1, 3", or "3, 2, 1", etc. here, and similarly for the cells marked as "4, 5, 6" and the cells marked as "7, 8, 9". This point is also emphasized in the summary of the algorithm in the following.

If the propagating direction of the sweeping wavefront forms an acute angle with the direction of the characteristic, then the causality (i.e., upwind information) along this characteristic can be correctly captured in this sweeping direction. A straight characteristic in any direction can be partitioned into two pieces by the tangent point to a particular spherical sweeping wavefront, and each piece forms an acute angle to the outgoing or incoming sweeping wavefront. If all characteristics are straight lines, we cover almost all characteristics by sweeping all cells according to their l^2- distances to a single reference point in both ascent and descent orders alternately, except for these characteristics at the tangent point where the normal of the sweeping wavefront is orthogonal to the direction of characteristics and upwind information is unable to propagate across the tangent point from one piece to other pieces effectively. This issue can be resolved by introducing another reference point. Then all directions of characteristics can be covered effectively by the four orderings of all cells (i.e., both ascent and descent orders based on two reference points) except one direction, which is orthogonal to the line connecting these two reference points. Hence at least three non-collinear reference points are needed, and the sweepings are performed through all the cells according to their l^2- distances to these reference points in ascent and descent orderings to cover all directions of information propagating along characteristics. When characteristics are not straight lines, any characteristic can be divided into a finite number of pieces so that each piece can still be covered effectively by one of the sweeping directions as shown in [57, 34], although practically we need more than three non-collinear reference points for efficient convergence of the iterations.

As discussed in [34], the criterion for an optimal choice of reference points and their locations on an unstructured mesh is that all directions of characteristics should be covered with minimal redundancy. In practice, it is better if these reference points are evenly spaced both spatially and angularly in the computational domain. For the high order WENO fast sweeping method for hyperbolic conservation laws here, we use four reference points which are evenly (or roughly evenly) distributed on the domain boundary, and numerical experiments show that the resulted alternating sweeping directions are effective for fast convergence of the scheme. For example, we use four corners as reference points if the computational domain is rectangular. Also see Figure 2 for an illustration of reference points for a rectangular computational domain and a circular computational domain, and the sweeping wavefronts based on one of the reference points. Here we would like to emphasize that the sweeping wavefronts (e.g., the arcs in Figure 2) can be considered as "continuous limit" of the sweeping directions, and they are used to explain this method about constructing the sweeping directions on unstructured meshes. However, these sweeping wavefronts do not appear in the implementation of the method and their ordering (i.e., the ordering of the arcs in Figure 2) does not need to be preserved in the computational codes. Instead, we need to

save the ascent and descent orderings of all the cells on the computational mesh according to their centroids' distances to the chosen reference points on the domain, and use them as the sweeping directions in the iterations. The implementation details are in the following.

For each reference point R_l, $l = 1, \cdots, 4$, we sort all cells on the unstructured mesh according to their centroids' distances to the reference point R_l in ascent and descent orders, and put their index numbers into arrays, e.g., the array $S_l^+(i), i = 1, 2, \cdots, M$ to store cell index numbers in the resulted ascent order and the array $S_l^-(i), i = 1, 2, \cdots, M$ to store cell index numbers in the resulted descent order. With four reference points, we obtain eight sweeping orders. Then in the proposed fixed-point WENO fast sweeping scheme (2.15), we alternatively take $k_i = S_l^+(i), i = 1, 2, \cdots, M$ and $k_i = S_l^-(i), i = 1, 2, \cdots, M$, for $l = 1, \cdots, 4$, as the sweeping iteration directions. Note that these initial orderings of cells only need to be performed for a fixed mesh once and for all, hence their cost is just a small part of the whole simulation cost. In the numerical tests of this paper, we use a comparison-based sorting method [10]. The complete algorithm is summarized as the following.

Step 1. Choose four reference points: $R_l, l = 1, \cdots, 4$.

Step 2. For each reference point $R_l, l = 1, \cdots, 4$, all triangle cells are sorted according to the l^2-distances of their centroids to the reference points R_l in ascent and descent orders, and store their index numbers in arrays:

$$S_l^+(i) : \text{ascent order}, i = 1, 2, \cdots, M;$$

$$S_l^-(i) : \text{descent order}, i = 1, 2, \cdots, M.$$

Note that if the distances of two cells' centroids to a reference point is very close to each other, for example, their difference is at the round-off error level, the order of updates for these two cells, which may depend on the sorting method used and its implementation, does not matter and their order does not affect the convergence of the iterations. This point has been discussed for the first order scheme in [34], and also confirmed in our numerical simulations for the high order fast sweeping method developed here.

Step 3. Gauss-Seidel iterations using the fast sweeping scheme (2.15) are performed for $n = 1, 2, \cdots$, according to the sweeping iteration directions obtained above:

do $l = 1, \cdots, 4$
  do $i = 1, \cdots, M$
    $k_i = S_l^+(i)$;
    Apply (2.15) and update $\overline{\mathbf{u}}_{k_i}^{n+1}$ on the triangle k_i.
  enddo
  Perform convergence check.
  do $i = 1, \cdots, M$
    $k_i = S_l^-(i)$;
    Apply (2.15) and update $\overline{\mathbf{u}}_{k_i}^{n+1}$ on the triangle k_i.
  enddo
  Perform convergence check.
enddo

Note that the convergence check is performed via first computing the iteration residue $ResA$ defined in the following section, then judging whether the convergence criterion $ResA \leq \delta$ for $\delta > 0$

given is satisfied. If it is satisfied, we stop the Gauss-Seidel iterations. Since the strategy of alternating direction sweepings utilizes the characteristics property of hyperbolic PDEs, combining with the Gauss-Seidel philosophy, we observe the acceleration of convergence to steady state solutions as shown in the following numerical experiments. By the Gauss-Seidel philosophy, we use the newest numerical values on the computational stencil of the WENO scheme whenever they are available in the current iteration step. This is why we use the notation $\overline{\mathbf{u}}_{il}^{*}, l = 1, 2, 3, 11, \cdots, 321$, to represent the numerical values in the scheme (2.15), and $\overline{\mathbf{u}}_{il}^{*}$ could be the value $\overline{\mathbf{u}}_{il}^{n}$ in the previous iteration step n, or the new value $\overline{\mathbf{u}}_{il}^{n+1}$ which has been updated and available in the current iteration step, depending on the current sweeping direction of the iteration. To guarantee that the fixed-point iteration (2.15) is a contractive mapping and converges, suitable values of $\triangle t_n$ which depends on the CFL number, need to be taken. This is similar to choose a suitable CFL number for stability when a high order WENO scheme is used to solve time-dependent hyperbolic PDEs.

3 Numerical experiments

In this section, we carry out numerical experiments to test the proposed absolutely convergent fixed-point fast sweeping WENO method on triangular meshes for solving some benchmark steady-state problems of hyperbolic conservation laws. Computational efficiency of the fast sweeping scheme and the other two time marching schemes is compared. For the convenience of presentation, we call the proposed absolutely convergent fixed-point fast sweeping WENO scheme (2.15) "FE fast sweeping scheme", the forward Euler time marching scheme (2.13) "FE Jacobi scheme", and the TVD-RK3 time marching scheme (2.14) "RK Jacobi scheme". Mesh refinement studies are performed to compute L_1 and L_∞ numerical errors and accuracy orders of these iterative schemes. Iteration numbers and CPU times for each iterative scheme to converge are recorded to compare their computational efficiency. The convergence of the iterations is measured by the average residue which is defined as

$$ResA = \sum_{i=1}^{M} \frac{|R1_i| + |R2_i| + |R3_i| + |R4_i|}{4 \times M}. \tag{3.1}$$

Here $R*_i$'s are the local residuals of the conservative variables, namely, $R1_i = \frac{\overline{\rho}_i^{n+1} - \overline{\rho}_i^{n}}{\Delta t_n}, R2_i = \frac{(\overline{\rho u})_i^{n+1} - (\overline{\rho u})_i^{n}}{\Delta t_n}, R3_i = \frac{(\overline{\rho v})_i^{n+1} - (\overline{\rho v})_i^{n}}{\Delta t_n}, R4_i = \frac{\overline{E}_i^{n+1} - \overline{E}_i^{n}}{\Delta t_n}$. M is the total number of triangular cells. The convergence criterion is set to be $ResA \leq \delta$, where the threshold value δ is taken to be at round off error level $10^{-12} \sim 10^{-11}$. $\triangle t_n$ depends on the CFL number as the following

$$\triangle t_n = \frac{CFL}{\max\limits_{1 \leq i \leq M} \left(\frac{\sum_{ll=1}^{3} \left[|(\overline{u}_i^n, \overline{v}_i^n) \cdot \overrightarrow{\mathbf{n}}_{i_{ll}}| + c_i^n \right] |\partial \triangle_{i_{ll}}|}{2|\triangle_i|} \right)}, \tag{3.2}$$

where $c_i^n = \sqrt{\frac{\gamma' \overline{p}_i^n}{\overline{\rho}_i^n}}$ is the sound speed with $\gamma' = 1.4$. In this paper, the number of iterations reported in the numerical simulations counts a complete update of numerical values in all cells once as one iteration. Hence these eight alternating directions in the FE fast sweeping scheme are counted as eight iterations. In the numerical experiments, for various examples we compare the computational efficiency of different schemes by using the largest possible CFL numbers which lead to iteration convergence with the fastest speed for each method. To identify the largest possible CFL number for a problem, we gradually increase / decrease the values of CFL number from an initial value.

We first test these schemes by solving two problems with a smooth solution to verify their accuracy, and compare their efficiency to converge to a steady-state solution.

Example 1. An Euler system of equations with source terms
In this example, we solve for steady-state solution of the following two dimensional Euler system of equations with source terms

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix}=\begin{pmatrix}0.4\cos(x+y)\\ 0.6\cos(x+y)\\ 0.6\cos(x+y)\\ 1.8\cos(x+y)\end{pmatrix}. \tag{3.3}$$

The exact steady-state solutions of the problem are $\rho(x,y,\infty)=1+0.2\sin(x+y), u(x,y,\infty)=1, v(x,y,\infty)=1, p(x,y,\infty)=1+0.2\sin(x+y)$. The computation domain is $(x,y)\in[0,2\pi]\times[0,2\pi]$, and the exact steady-state solutions are applied on the domain boundaries. Three different iterative methods, i.e., the FE Jacobi scheme, the RK Jacobi scheme, and the FE fast sweeping scheme, are used to solve the problem on successively refined unstructured triangular meshes. The coarsest mesh used here is shown in Figure 4. In the mesh refinement study to test the accuracy orders of the schemes, the mesh is refined by cutting each triangle in the coarse mesh into four smaller similar ones. As discussed in Section 2, we take four corners of the domain, i.e., $(0,0),(0,2\pi),(2\pi,0)$ and $(2\pi,2\pi)$, as the reference points to form the alternating sweeping directions in the FE fast sweeping scheme. To start the iterations for the schemes, we take the numerical initial conditions to be the same as the exact steady-state solutions, which does not satisfy the numerical schemes and will be driven by the iterative schemes to the numerical steady states. In Table 1, we report the numerical accuracy for the density variable, iteration numbers and CPU times when these three different iterative schemes reach the average residue threshold value 10^{-12} of the convergence criterion. It is observed that all three schemes achieve basically the same numerical errors and the fifth order accuracy when the iterations converge. This is as expected since although they are different iterative schemes, they converge to the solution of the same nonlinear algebraic system resulted from the fifth order finite volume WENO local solver. However, these methods exhibit very different computational efficiency by comparing the iteration numbers and CPU times required by them to reach steady state. As shown in Table 1, the FE Jacobi scheme requires a very small CFL number 0.1 to achieve the convergence. The reason is that a forward Euler time discretization with a high order linear upwind spatial discretization suffers from linear stability issue. When the nonlinearly stable WENO discretization is applied, it alleviates the linear instability problem. As a result of balance, the forward Euler time marching scheme, i.e. the FE Jacobi scheme, converges under a tiny CFL number in this example. That leads to the largest iteration numbers and the most CPU time costs among these three iterative schemes. By using the TVD-RK3 scheme (i.e. the RK Jacobi scheme), both linear and nonlinear stability are maintained. Hence here a larger CFL number 0.6 can be used to make the iterations converge. Although the RK Jacobi scheme has three stages in one time step, in this example it needs fewer number of iterations and less CPU costs to converge to the steady state, and is more efficient than the FE Jacobi scheme. As shown in Table 1, the proposed fixed-point fast sweeping method (i.e. the FE fast sweeping scheme) is the most efficient one among all three iterative methods. Furthermore, it is important to notice that the FE fast sweeping scheme permits a CFL number 0.6 which is the same as the TVD-RK3 scheme. This shows that the fixed-point fast sweeping method can improve the linear stability of the forward Euler scheme when it is coupled with a high order spatial discretization. Hence, it suggests that by applying the fixed-point fast sweeping technique to the forward Euler scheme, the forward Euler scheme with a high order WENO spatial discretization becomes practically useful and efficient in solving steady-state hyperbolic conservation laws. In Table 1, it is observed that on refined meshes, the number of iterations for the FE fast sweeping method to converge to the steady state is only about $13\% \sim 15\%$ of that for the forward Euler time marching scheme, and

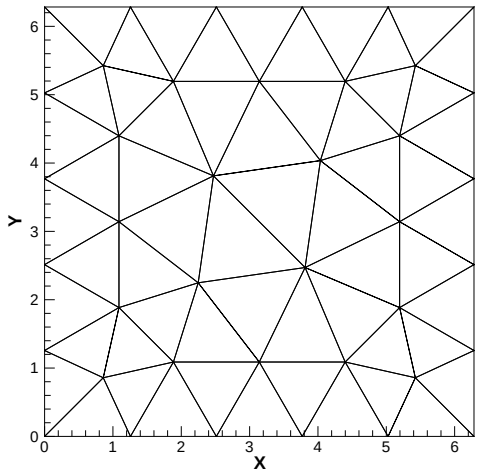


Figure 4: The coarsest unstructured triangular mesh for Example 1 and Example 2.

about $25\% \sim 30\%$ of that for the TVD-RK3 scheme. Since the Gauss-Seidel iterations in the FE fast sweeping method (2.15) require that the newest numerical values of $\overline{\mathbf{u}}$ are used in the finite volume WENO reconstruction stencils as long as they are available, the WENO reconstructions for both $\mathbf{u}^-$ and $\mathbf{u}^+$ in (2.4) need to be performed twice for the same Gaussian quadrature points on the shared boundary of two different target cells. This is the reason that the CPU time cost of the FE fast sweeping method is about $26\% \sim 30\%$ of that for the forward Euler time marching scheme, and about $50\% \sim 60\%$ of that for the TVD-RK3 scheme. Hence on refined meshes in this example, the FE fast sweeping method saves about 70% CPU time cost for the forward Euler time marching scheme, and saves around 50% CPU time cost for the TVD-RK3 scheme.

Example 2. An Euler system of equations without source terms
In this example, we solve for steady-state solution of the following two dimensional Euler system of equations. Different from Example 1, the system does not involve any source terms. The equations are

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix}=\begin{pmatrix}0\\ 0\\ 0\\ 0\end{pmatrix}. \tag{3.4}$$

We consider the case that the system has the exact steady-state solution $\rho(x,y,\infty)=1+0.2\sin(x-y), u(x,y,\infty)=1, v(x,y,\infty)=1, p(x,y,\infty)=1$, and solve the system by the FE Jacobi scheme, the RK Jacobi scheme and the FE fast sweeping scheme to compare their performance. The computational domain, boundary conditions, triangular meshes used for the simulations, and the reference points to form the alternating sweeping directions in the FE fast sweeping scheme, are all the same as these in Example 1. Also, to start the iterations for the schemes, we take the numerical initial conditions to be the same as the exact steady-state solutions, which does not satisfy the numerical schemes and will be driven by these iterative schemes to their numerical steady states. In Table 2, we report the numerical accuracy for the density variable, iteration numbers and CPU times when these three different iterative schemes reach the average residue threshold value 10^{-12} of the convergence criterion. It can be observed that all three schemes achieve basically the same numerical errors and the fifth order accuracy when the iterations converge. Again, these methods

FE Jacobi, CFL=0.1						
M	L_1 error	order	L_∞ error	order	iter♯	CPU time
58	1.24E-02	-	3.79E-02	-	5168	10.13
232	4.95E-04	4.64	1.63E-03	4.54	7387	49.33
928	1.58E-05	4.97	5.19E-05	4.97	9306	231.73
3712	5.10E-07	4.95	1.54E-06	5.07	13919	1350.48
14848	1.62E-08	4.97	5.94E-08	4.70	24950	10003.34
59392	5.13E-10	4.98	2.56E-09	4.54	40437	64344.13
RK Jacobi, CFL=0.6						
M	L_1 error	order	L_∞ error	order	iter♯	CPU time
58	1.24E-02	-	3.79E-02	-	2589	5.16
232	4.95E-04	4.64	1.63E-02	4.54	3687	25.08
928	1.58E-05	4.97	5.19E-05	4.97	4638	117.20
3712	5.10E-07	4.95	1.54E-06	5.07	6945	696.03
14848	1.62E-08	4.97	5.94E-08	4.70	12465	5086.49
59392	5.13E-10	4.98	2.55E-09	4.54	20301	32920.73
FE fast sweeping, CFL=0.6						
M	L_1 error	order	L_∞ error	order	iter♯	CPU time
58	1.24E-02	-	3.79E-02	-	686	1.99
232	4.95E-04	4.64	1.63E-03	4.54	990	11.36
928	1.58E-05	4.97	5.19E-05	4.97	1384	59.08
3712	5.10E-07	4.95	1.54E-06	5.07	1870	350.07
14848	1.62E-08	4.97	5.94E-08	4.70	3326	2756.83
59392	5.14E-10	4.98	2.56E-09	4.54	5926	21219.23

Table 1: Example 1, an Euler system of equations with source terms. Accuracy, iteration numbers and CPU times of three different iterative schemes. CPU time unit: second.

FE Jacobi, CFL=0.1						
N	L_1 error	order	L_∞ error	order	iter♯	CPU time
58	2.28E-02		7.62E-02		933	1.82
232	1.08E-03	4.40	4.88E-03	3.97	1191	8.01
928	3.97E-05	4.76	1.97E-04	4.63	1577	40.14
3712	1.52E-06	4.71	7.85E-06	4.65	2463	252.57
14848	5.13E-08	4.89	2.73E-07	4.85	4532	1777.16
RK Jacobi, CFL=0.6						
N	L_1 error	order	L_∞ error	order	iter♯	CPU time
58	2.28E-02		7.62E-02		705	1.58
232	1.08E-03	4.40	4.88E-03	3.97	876	6.46
928	3.97E-05	4.76	1.97E-04	4.63	1182	31.25
3712	1.52E-06	4.71	7.85E-06	4.65	1884	195.02
14848	5.13E-08	4.89	2.73E-07	4.85	3360	1372.57
FE fast sweeping, CFL=0.6						
N	L_1 error	order	L_∞ error	order	iter♯	CPU time
58	2.28E-02		7.62E-02		136	0.42
232	1.08E-03	4.40	4.88E-03	3.97	156	1.84
928	3.97E-05	4.76	1.97E-04	4.63	210	10.21
3712	1.52E-06	4.71	7.85E-06	4.65	326	64.69
14848	5.13E-08	4.89	2.73E-07	4.85	580	460.33

Table 2: Example 2, an Euler system of equations without source terms. Accuracy, iteration numbers and CPU times of three different iterative schemes. CPU time unit: second.

exhibit very different computational efficiency by comparing the iteration numbers and CPU times required by them to reach steady state. As shown in Table 2, the proposed FE fast sweeping scheme is the most efficient one among all three iterative methods. It permits a larger CFL number than the FE Jacobi scheme. On refined meshes, the number of iterations for the FE fast sweeping method to converge to the steady state is only about 13% of that for the FE Jacobi scheme, and about 17% of that for the RK Jacobi scheme. The Gauss-Seidel iterations in the FE fast sweeping method require that the newest numerical values are used in the finite volume WENO reconstruction stencils, so the WENO reconstructions for both $\mathbf{u}^-$ and $\mathbf{u}^+$ in (2.4) need to be performed twice for the same Gaussian quadrature points on the shared boundary of two different target cells. Hence, we observe that the CPU time cost of the FE fast sweeping method is about 26% of that for the FE Jacobi scheme, and about 33% of that for the RK Jacobi scheme. On refined meshes in this example, the FE fast sweeping method saves about 74% CPU time cost for the forward Euler time marching scheme, and saves about 67% CPU time cost for the TVD-RK3 scheme.

In the following examples, We study and compare these schemes by solving problems with discontinuous solutions.

Example 3. Regular shock reflection

In this example, we test the absolute convergence of the proposed fixed-point fast sweeping method on unstructured meshes, by solving the regular shock reflection problem. The problem is a 2D steady-state Euler system of equations with a reflection condition along the bottom boundary, as described in e.g. [47, 50, 51]. It is a typical and difficult benchmark problem of using high

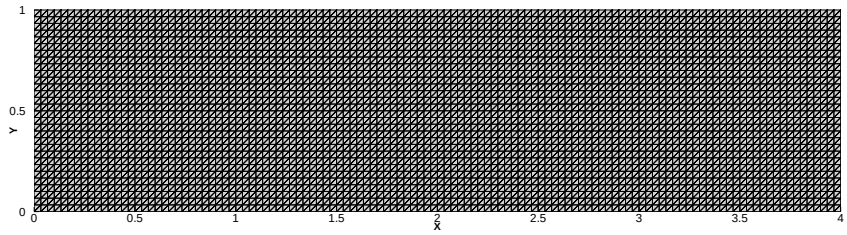


Figure 5: The computational mesh for Example 3, regular shock reflection.

order schemes to simulate steady flow. As that found in [50, 51], even with advanced techniques to improve the steady-state convergence, it is still difficult for the residue of high order WENO schemes to converge to the level of round off errors. The iteration residue of the fast sweeping method in [47], which is based on the WENO scheme with equal-sized sub-stencils, hangs at the level above $10^{-3.5}$. However, the proposed unstructured fast sweeping method with unequal-sized sub-stencils can drive the iteration residue to the level of round off errors, as shown in the following numerical results.

The computational domain is a rectangle and it has the length 4 and the height 1. The boundary conditions include a reflection condition along the bottom boundary, supersonic outflow along the right boundary, and the Dirichlet conditions on the other two boundaries:

$$(\rho, u, v, p)^T = \begin{cases} (1.0, 2.9, 0, 5/7)^T \mid_{(0,y)}, \\ (1.69997, 2.61934, -0.50632, 1.52819)^T \mid_{(x,1)} . \end{cases}$$

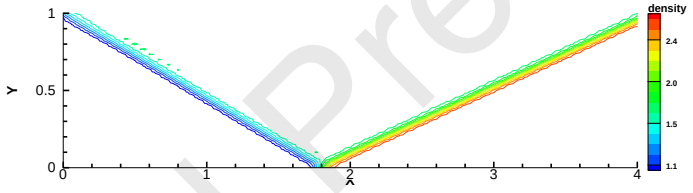
The initial values used to start the iterations in the entire domain are taken to be the same as those at the left boundary. The computational mesh is shown in Figure 5. Four corners of the domain are taken as the reference points to form the alternating sweeping directions in the FE fast sweeping scheme. In Table 3, number of iterations required to reach the convergence criterion threshold value 10^{-11}, and total CPU time when the schemes converge under various CFL numbers are reported for the FE Jacobi scheme, the RK Jacobi scheme and the FE fast sweeping scheme. As the previous examples, the FE Jacobi scheme requires a very small CFL number 0.1 to achieve the convergence, which results in many iterations and large CPU time cost. If the RK Jacobi (TVD-RK3) scheme is used, the CFL number can be enlarged to 1.0 and its iterations converge much more efficiently than the FE Jacobi scheme. Among these three methods, the FE fast sweeping scheme is still the most efficient one. It can converge under the similar CFL numbers to the RK Jacobi scheme. With the largest CFL number permitted for each method to reach steady state solution, the proposed FE fast sweeping method on triangular meshes saves about 85% CPU time cost for the forward Euler method, and saves about 50% CPU time cost for the TVD-RK3 method. In Figure 6, density contours of the converged steady state solutions of these three schemes are shown, and the similar numerical results are obtained as expected. In Figure 7, residue history in terms of iterations for these three schemes with different CFL numbers is shown. It is observed that the residue of iterations settles down to very small values at the level of round off errors for all cases. The absolute convergence for the proposed fast sweeping method on triangular meshes is verified for this difficult benchmark problem.

Example 4. The supersonic flow past a circular cylinder

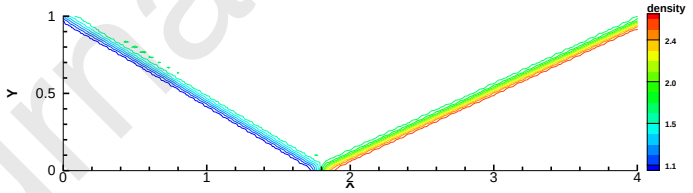
In this example, we consider an inviscid, compressible flow which initially moves toward a circular cylinder from the left at a Mach number of $Ma = 2$ [31, 32]. The cylinder with the radius 0.5

FE Jacobi scheme		
CFL number	iteration number	CPU time
0.1	20926	6681.16
0.2	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
0.6	10308	3257.98
1.0	6183	1959.48
1.1	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
0.6	3328	1851.67
1.0	1808	1020.90
1.1	Not convergent	-

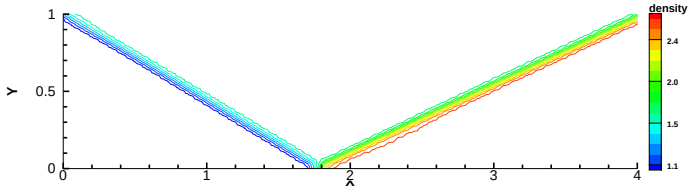
Table 3: Example 3, regular shock reflection. Number of iterations and total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11}. CPU time unit: second.



(a) FE Jacobi scheme

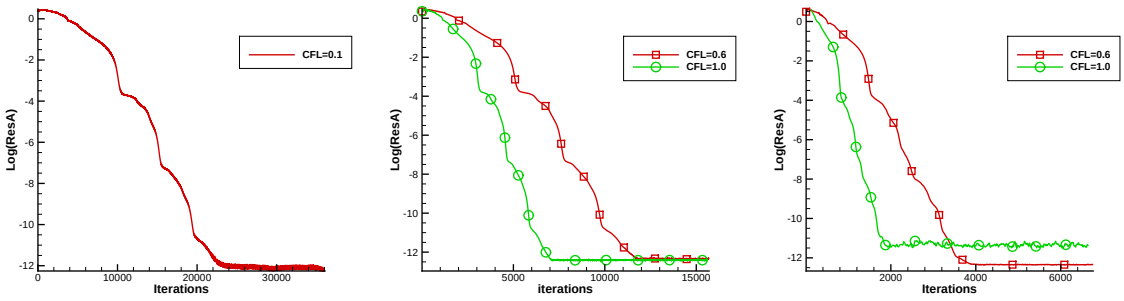


(b) RK Jacobi scheme



(c) FE fast sweeping scheme

Figure 6: Example 3, regular shock reflection. 30 equally spaced density contours from 1.1 to 2.6 of the converged steady states of numerical solutions by three different iterative schemes.



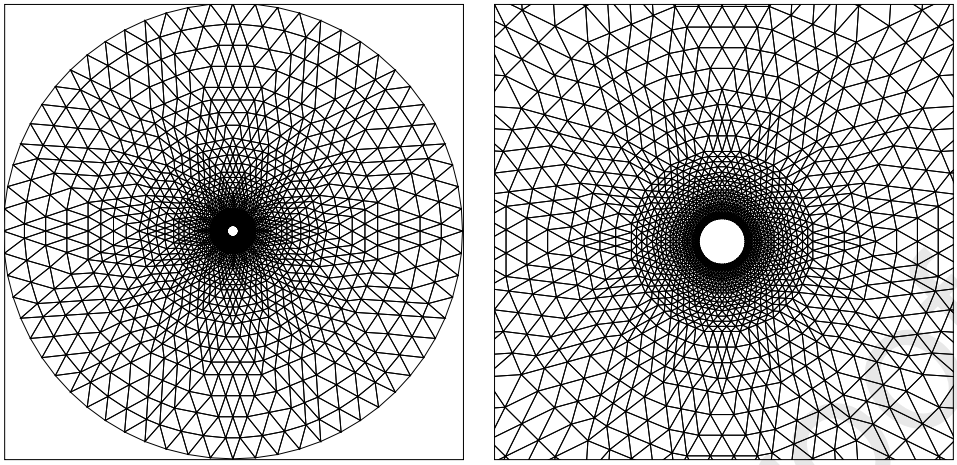
(a) FE Jacobi scheme (b) RK Jacobi scheme (c) FE fast sweeping scheme

Figure 7: Example 3, regular shock reflection. The convergence history of the residue as a function of number of iterations for three schemes with different CFL numbers.

is located at the origin, and the computational domain is $\{(x, y) : 0.5 \leq \sqrt{x^2 + y^2} \leq 20\}$. The unstructured triangular mesh used in this example is shown in Figure 8, in which the number of grid points on the boundaries is 64. As discussed in Section 2, for such circular domain here, we take four points which are evenly distributed on the outer boundary, $(-20, 0), (0, -20), (20, 0)$ and $(0, 20)$, as the reference points to form the alternating sweeping directions in the FE fast sweeping scheme. In Table 4, number of iterations required to reach the convergence criterion threshold value 10^{-11}, and total CPU time when the schemes converge under various CFL numbers are reported for the FE Jacobi scheme, the RK Jacobi scheme, and the FE fast sweeping scheme. Different from the previous examples, in this example the FE Jacobi scheme converges under a reasonable CFL number. Although the RK Jacobi scheme still converges under a larger CFL number than the FE Jacobi scheme, it is less efficient here due to its multi-stage structure. Consistent with the previous examples, the FE fast sweeping scheme is still the most efficient one among three methods, and it permits the similar CFL numbers to the RK Jacobi scheme. With the largest CFL number allowed for the methods to reach steady state solution, the FE fast sweeping method on unstructured triangular meshes saves about 70% CPU time cost of that by the RK Jacobi scheme (the TVD-RK3 scheme). In Figure 9, the pressure contours of the converged steady state solutions of these three schemes are presented. We observe similar numerical steady states of these schemes. The residue history of these three schemes with different CFL numbers is reported in Figure 10, which shows that the residue of iterations settles down to very small values at the level of round off errors and verifies the absolute convergence of the proposed high order fast sweeping method on triangular meshes.

Example 5. Supersonic and subsonic flows past an NACA001035 airfoil

In this example, we solve the problem of supersonic and subsonic flows past a single NACA001035 airfoil configuration in [11, 61]. Following the setup in [61], we consider the case of supersonic flow with Mach number $Ma = 2$, angle of attack $\alpha = 1°$; and the case of subsonic flow with Mach number $Ma = 0.8$, angle of attack $\alpha = 1.25°$. The computation domain is $[-16, 16] \times [-16, 16]$. The unstructured mesh containing 5593 triangles used for this example is shown in Figure 11. Four corners of the computational domain are taken as the reference points to form the alternating sweeping directions in the FE fast sweeping scheme. In Table 5, number of iterations required to reach the convergence criterion threshold value 10^{-11}, and total CPU time when the schemes converge under various CFL numbers are reported for the FE Jacobi scheme, the RK Jacobi scheme,

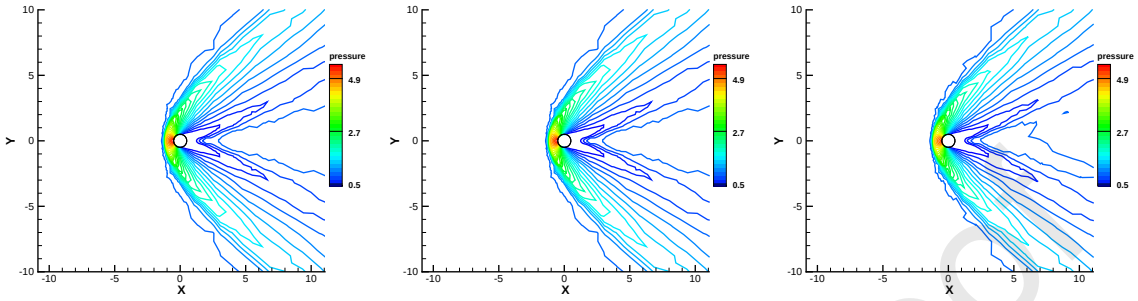


(a) (b)

Figure 8: The computational mesh for Example 4, the supersonic flow past a circular cylinder. Left: the whole region; right: zoomed near the circular cylinder.

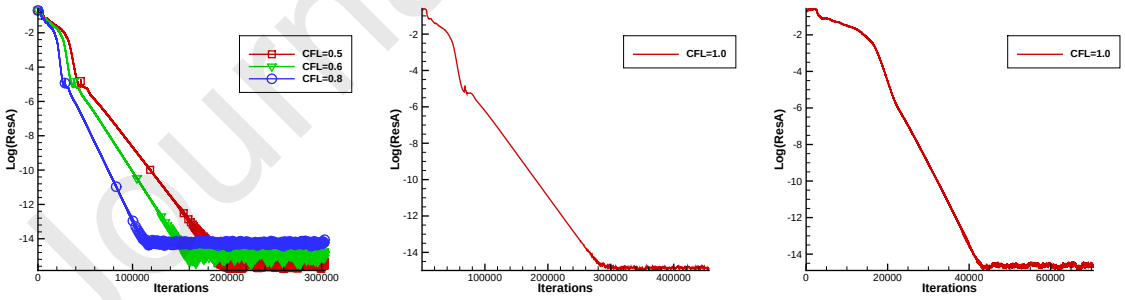
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.5	132862	22369.13
0.6	110709	20776.28
0.8	83015	15220.13
0.9	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
1.0	201529	36827.34
1.1	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
1.0	34482	10957.96
1.1	Not convergent	-

Table 4: Example 4, the supersonic flow past a circular cylinder. Number of iterations and total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11}. CPU time unit: second.



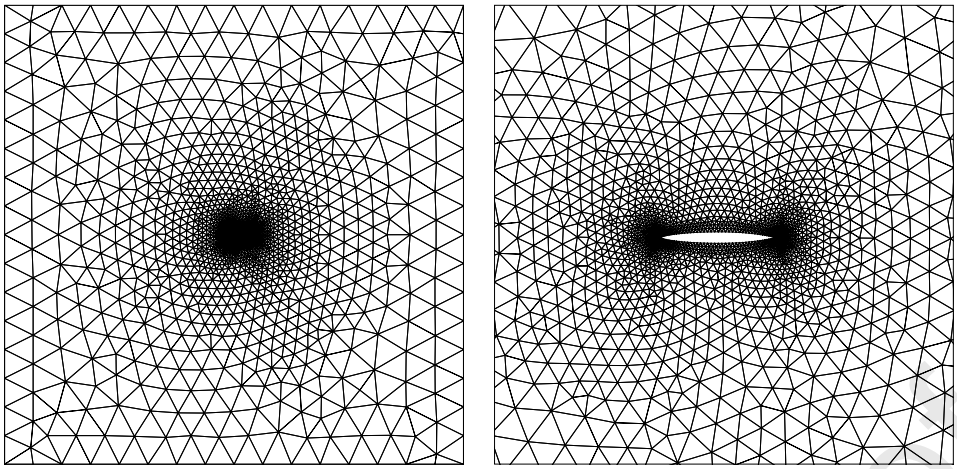
(a) FE Jacobi scheme (b) RK Jacobi scheme (c) FE fast sweeping scheme

Figure 9: Example 4, the supersonic flow past a circular cylinder. 30 equally spaced pressure contours from 0.5 to 5.4 of the converged steady states of numerical solutions by three different iterative schemes.



(a) FE Jacobi scheme (b) RK Jacobi scheme (c) FE fast sweeping scheme

Figure 10: Example 4, the supersonic flow past a circular cylinder. The convergence history of the residue as a function of number of iterations for three schemes with different CFL numbers.



(a) (b)

Figure 11: The computational mesh for Example 5, supersonic and subsonic flows past an NACA001035 airfoil. Left: the whole domain; right: zoomed region near the airfoil.

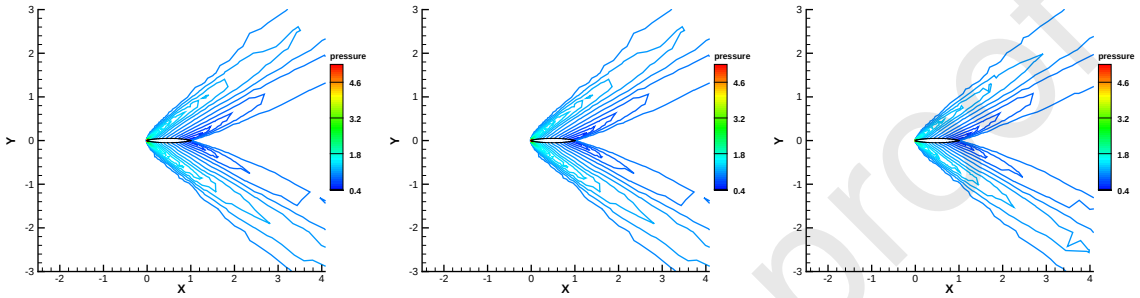
and the FE fast sweeping scheme. In this example, the CFL number constraint for the FE Jacobi scheme to converge is not as severe as that in the examples 1, 2 and 3. Using the largest CFL number permitted and due to its simple one-stage structure, the FE Jacobi scheme takes less number of iterations and less CPU times to converge to steady states than the RK Jacobi scheme, for both the supersonic flow case and the subsonic flow case. Again, the FE fast sweeping scheme is the most efficient iterative method among three methods. Here it permits much larger CFL numbers than the FE Jacobi scheme, and also slightly larger CFL numbers than the RK Jacobi scheme. It also has a simple one-stage structure. With the largest CFL number allowed for the methods to reach steady state solution, the FE fast sweeping method on unstructured triangular meshes saves about $60\% \sim 65\%$ CPU time cost of that by the RK Jacobi scheme (the TVD-RK3 scheme) for both the supersonic flow case and the subsonic flow case. In Figure 12, the pressure contours of the converged steady state solutions of these three schemes are presented for the supersonic flow case and the subsonic flow case. We observe comparable numerical steady states of these schemes. The residue history of these three schemes with different CFL numbers is reported in Figure 13, which shows that the residue of iterations can settle down to tiny values around 10^{-14} to 10^{-15}, at the level of round off errors. Again, the absolute convergence of the proposed high order fast sweeping method on triangular meshes is verified in this example.

Example 6. Supersonic flows past an NACA0012 airfoil

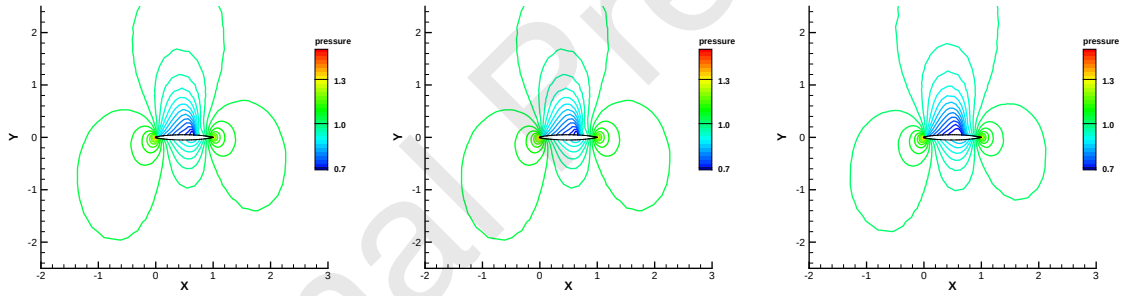
In this example, we consider the problem of inviscid Euler supersonic flows past a single NACA0012 airfoil configuration in [32]. Following the setup in [61], we consider the case of flow with Mach number $Ma = 3$, angle of attack $\alpha = 10^\circ$; and the case of flow with Mach number $Ma = 2$, angle of attack $\alpha = 1^\circ$. The computational domain is $[-15, 15] \times [-15, 15]$. The unstructured mesh for this example, which consists of 9340 triangles, is shown in Figure 14. As the previous example, four corners of the computational domain are taken as the reference points to form the alternating sweeping directions in the FE fast sweeping scheme. In Table 6, number of iterations required to reach the convergence criterion threshold value 10^{-11}, and total CPU time when the schemes converge under various CFL numbers are reported for the FE Jacobi scheme, the RK Jacobi scheme, and the FE fast sweeping scheme. Similar to Example 5, the CFL number constraint for the FE

supersonic, $Ma = 2, \alpha = 1°$		
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.3	8118	2527.66
0.4	6077	1904.80
0.5	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
0.6	12192	3709.35
0.9	8130	2488.23
1.0	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
0.7	2031	1183.33
1.0	1508	867.25
1.1	Not convergent	-
subsonic, $Ma = 0.8, \alpha = 1.25°$		
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.4	95877	31918.58
0.5	76794	24608.73
0.6	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
0.6	185661	63876.05
1.0	111396	37444.54
1.1	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
1.0	32854	22258.67
1.2	24876	15613.47
1.3	Not convergent	-

Table 5: Example 5, supersonic and subsonic flows past an NACA001035 airfoil. Number of iterations and total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11}. CPU time unit: second.

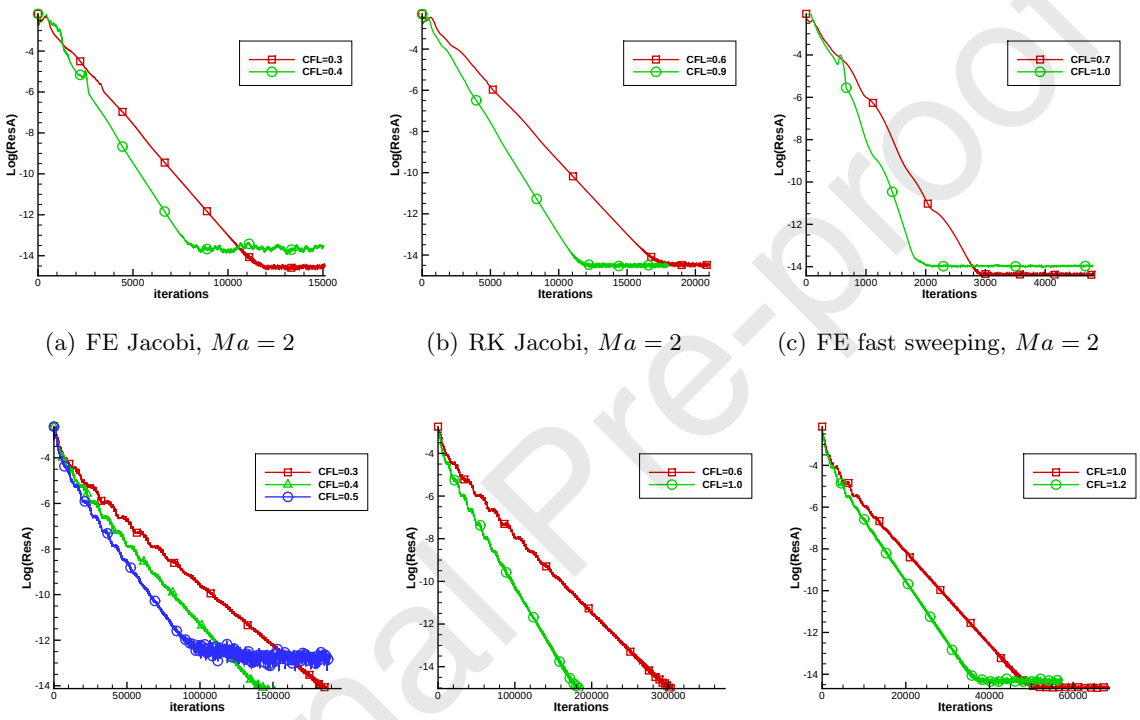


(a) FE Jacobi, $Ma = 2$ (b) RK Jacobi, $Ma = 2$ (c) FE fast sweeping, $Ma = 2$



(d) FE Jacobi, $Ma = 0.8$ (e) RK Jacobi, $Ma = 0.8$ (f) FE fast sweeping, $Ma = 0.8$

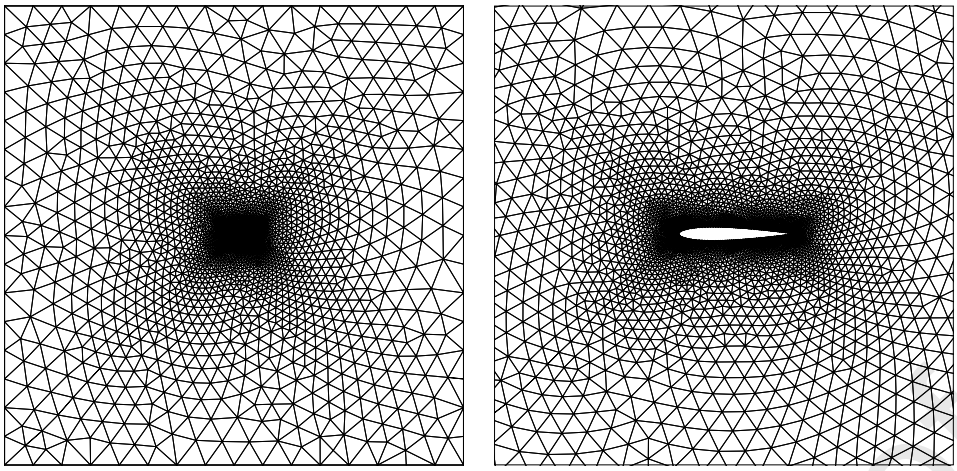
Figure 12: Example 5, supersonic and subsonic flows past an NACA001035 airfoil. The converged steady states of numerical solutions by three different iterative schemes. (a) (b) (c): 60 equally spaced pressure contours from 0.36 to 5.27 for the case of $Ma = 2$, angle of attack $\alpha = 1°$; (d) (e) (f): 30 equally spaced pressure contours from 0.7 to 1.45 for the case of $Ma = 0.8$, angle of attack $\alpha = 1.25°$.



(a) FE Jacobi, $Ma = 2$ (b) RK Jacobi, $Ma = 2$ (c) FE fast sweeping, $Ma = 2$

(d) FE Jacobi, $Ma = 0.8$ (e) RK Jacobi, $Ma = 0.8$ (f) FE fast sweeping, $Ma = 0.8$

Figure 13: Example 5, supersonic and subsonic flows past an NACA001035 airfoil. The convergence history of the residue as a function of number of iterations for three schemes with different CFL numbers. (a), (b), (c): for the case of $Ma = 2$, angle of attack $\alpha = 1°$; (d) (e) (f): for the case of $Ma = 0.8$, angle of attack $\alpha = 1.25°$.



(a) (b)

Figure 14: The computational mesh for Example 6 and Example 7, supersonic and subsonic flows past an NACA0012 airfoil. Left: the whole domain; right: zoomed region near the airfoil.

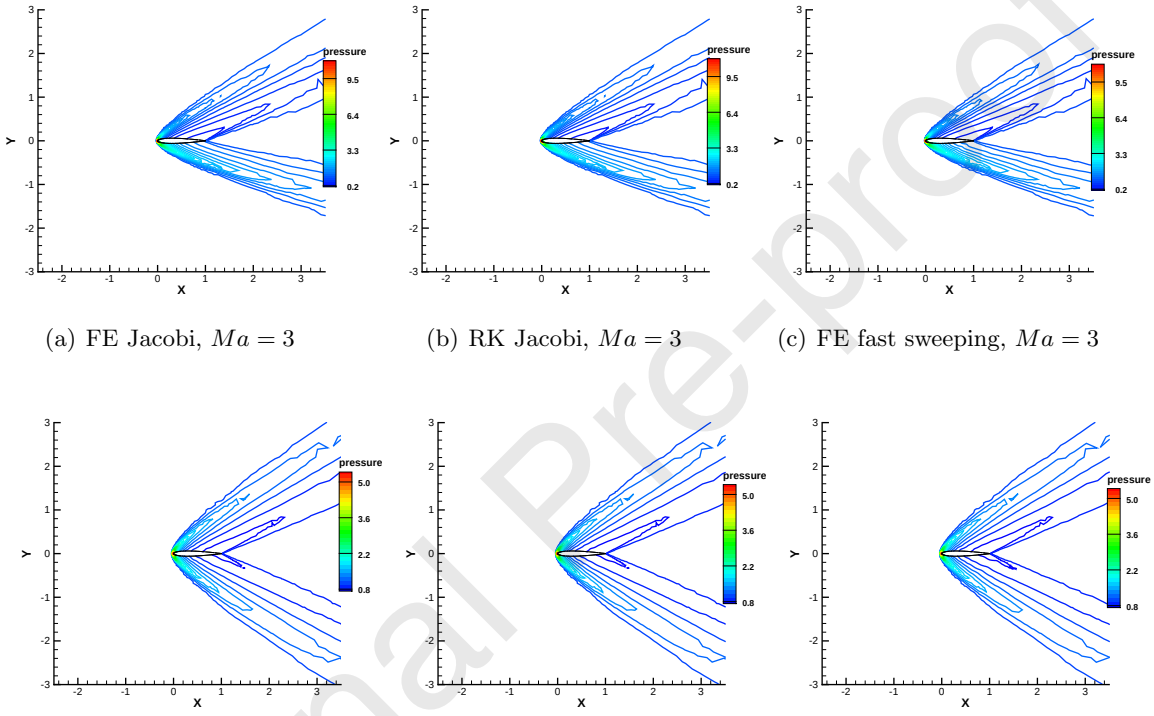
Jacobi scheme to converge is not as severe as that in the examples 1, 2 and 3. Due to the simple one-stage structure of the FE Jacobi scheme, it takes the FE Jacobi scheme less number of iterations and less CPU times to converge to steady states than the RK Jacobi scheme, for both cases with different Mach numbers. The FE fast sweeping scheme is still the most efficient iterative method among three methods. It allows much larger CFL numbers than the FE Jacobi scheme, and even slightly larger CFL numbers than the RK Jacobi scheme. It also has a simple one-stage structure. With the largest CFL number permitted in each method to reach steady state solution, the FE fast sweeping method on unstructured triangular meshes saves $50\% \sim 55\%$ CPU time cost of that by the RK Jacobi scheme (the TVD-RK3 scheme) for both supersonic flow cases in this example. In Figure 15, the pressure contours of the converged steady state solutions of these three schemes are presented for both supersonic flow cases. Again, we observe comparable numerical steady states of these schemes. Figure 16 shows the residue history of these three schemes with different CFL numbers. It is observed that the residue of iterations can settle down to tiny values at the level of round off errors, which verifies the absolute convergence of the developed high order fast sweeping method on triangular meshes here.

Example 7. Subsonic flows past an NACA0012 airfoil

As a continuation of Example 6, the problem of inviscid Euler subsonic flows past a single NACA0012 airfoil configuration in [32] is solved in this example. Two cases of subsonic flows are considered, i.e., a flow with Mach number $Ma = 0.8$, angle of attack $\alpha = 1.25°$; and a flow with Mach number $Ma = 0.2$, angle of attack $\alpha = 1°$. The computational domain is $[-15, 15] \times [-15, 15]$. The unstructured mesh used is the same as that for Example 6, which is shown in Figure 14. Four corners of the computational domain are chosen as the reference points to form the alternating sweeping directions in the FE fast sweeping scheme. In Table 7, number of iterations required to reach the convergence criterion threshold value 10^{-11}, and total CPU time when the schemes converge under various CFL numbers are reported for the FE Jacobi scheme, the RK Jacobi scheme, and the FE fast sweeping scheme. In this example, we also notice that the CFL number constraint for the FE Jacobi scheme to converge is not as severe as that in the examples 1, 2 and 3, and its simple one-stage structure makes it be more efficient to converge to steady states than the RK Jacobi

$Ma = 3, \alpha = 10°$		
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.6	75193	27432.75
0.7	64349	23534.16
0.8	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
1.0	134485	49427.91
1.2	112165	42557.77
1.3	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
1.0	44688	28805.77
1.2	37168	23178.72
1.4	31816	19563.90
1.5	Not convergent	-
$Ma = 2, \alpha = 1°$		
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.6	116201	47048.46
0.7	99598	40294.13
0.8	87149	36696.27
0.9	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
1.0	209200	82761.36
1.2	174319	69020.84
1.4	149425	61582.79
1.5	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
1.0	68696	47259.59
1.2	57208	38592.23
1.4	49000	32435.17
1.6	42848	29783.56
1.7	Not convergent	-

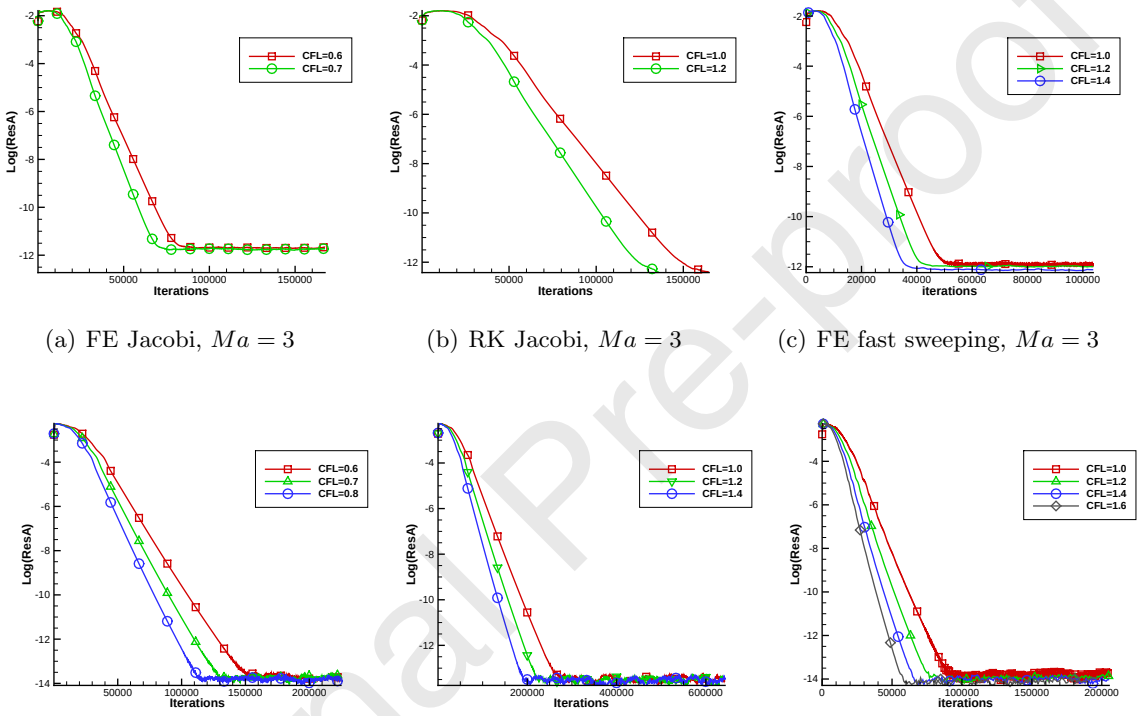
Table 6: Example 6, supersonic flows past an NACA0012 airfoil. Number of iterations and total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11}. CPU time unit: second.



(a) FE Jacobi, $Ma = 3$ (b) RK Jacobi, $Ma = 3$ (c) FE fast sweeping, $Ma = 3$

(d) FE Jacobi, $Ma = 2$ (e) RK Jacobi, $Ma = 2$ (f) FE fast sweeping, $Ma = 2$

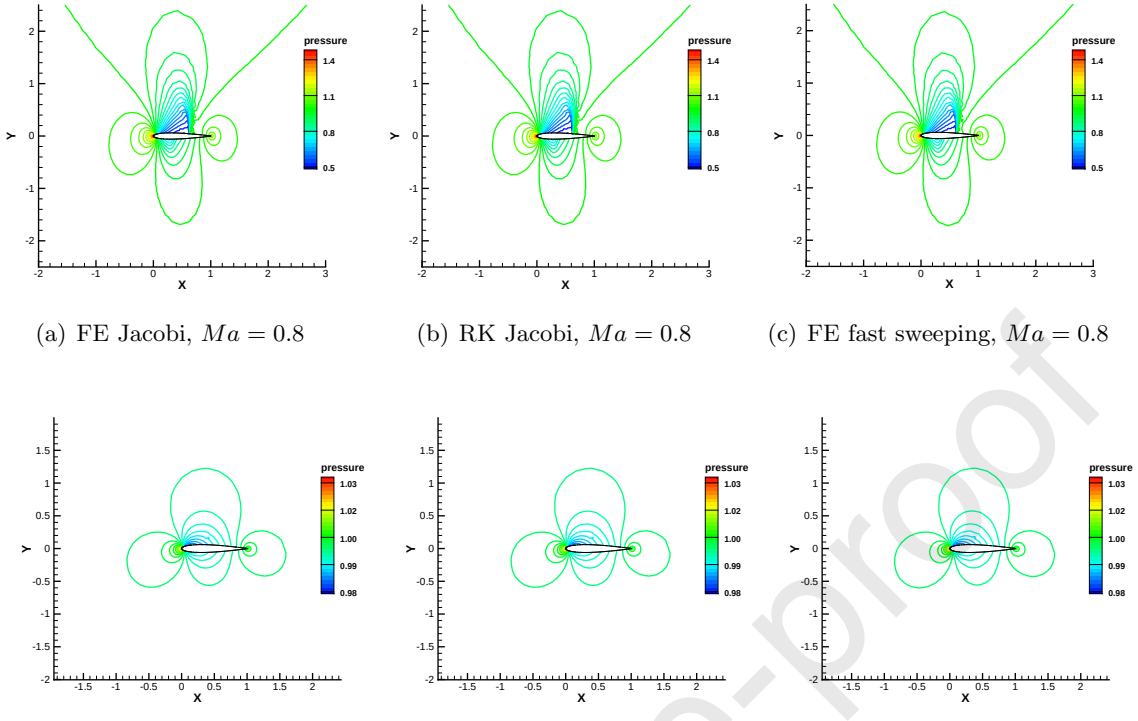
Figure 15: Example 6, supersonic flows past an NACA0012 airfoil. The converged steady states of numerical solutions by three different iterative schemes. (a) (b) (c): 30 equally spaced pressure contours from 0.2 to 11 for the case of $Ma = 3$, angle of attack $\alpha = 10°$; (d) (e) (f): 30 equally spaced pressure contours from 0.8 to 5.2 for the case of $Ma = 2$, angle of attack $\alpha = 1°$.



(a) FE Jacobi, $Ma = 3$ (b) RK Jacobi, $Ma = 3$ (c) FE fast sweeping, $Ma = 3$

(d) FE Jacobi, $Ma = 2$ (e) RK Jacobi, $Ma = 2$ (f) FE fast sweeping, $Ma = 2$

Figure 16: Example 6, supersonic flows past an NACA0012 airfoil. The convergence history of the residue as a function of number of iterations for three schemes with different CFL numbers. (a), (b), (c): for the case of $Ma = 3$, angle of attack $\alpha = 10°$; (d) (e) (f): for the case of $Ma = 2$, angle of attack $\alpha = 1°$.



(a) FE Jacobi, $Ma = 0.8$ (b) RK Jacobi, $Ma = 0.8$ (c) FE fast sweeping, $Ma = 0.8$

(d) FE Jacobi, $Ma = 0.2$ (e) RK Jacobi, $Ma = 0.2$ (f) FE fast sweeping, $Ma = 0.2$

Figure 17: Example 7, subsonic flows past an NACA0012 airfoil. The converged steady states of numerical solutions by three different iterative schemes. (a) (b) (c): 30 equally spaced pressure contours from 0.5 to 1.46 for the case of $Ma = 0.8$, angle of attack $\alpha = 1.25°$; (d) (e) (f): 30 equally spaced pressure contours from 0.98 to 1.03 for the case of $Ma = 0.2$, angle of attack $\alpha = 1°$.

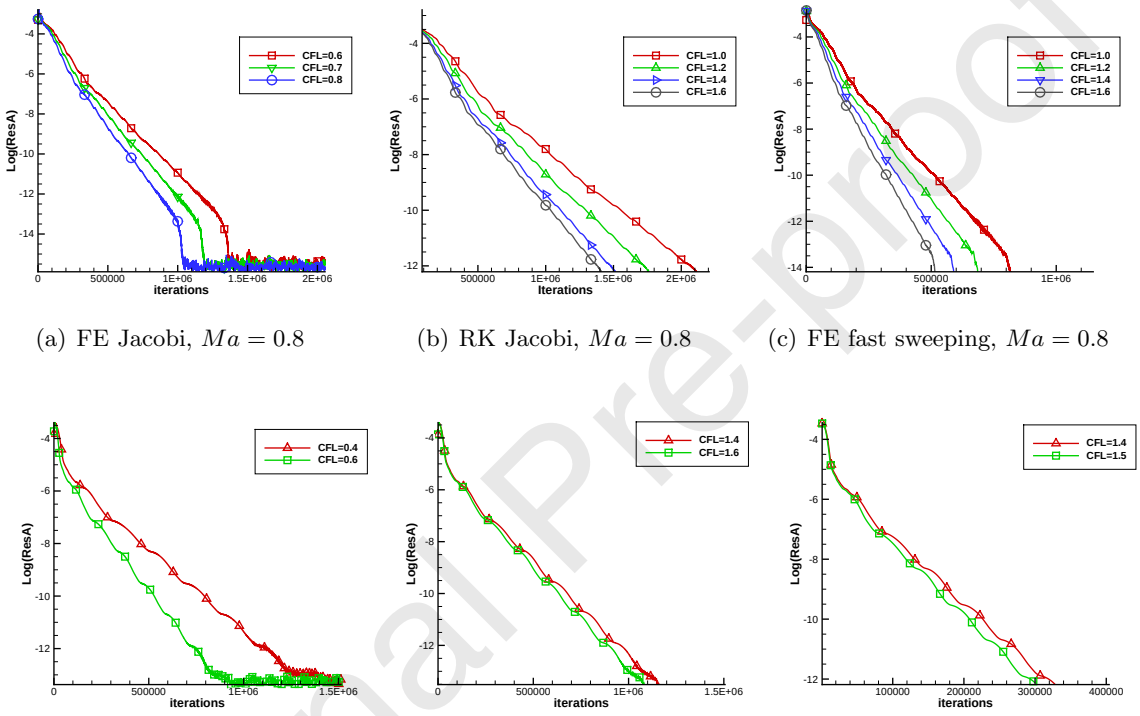
scheme for both cases with different Mach numbers. Again, the FE fast sweeping scheme is the most efficient iterative method among three methods. It allows much larger CFL numbers than the FE Jacobi scheme, and comparable CFL number sizes with the RK Jacobi scheme. The FE fast sweeping scheme also has a simple one-stage structure. With the largest CFL number permitted in each method to reach steady state solution, the FE fast sweeping method on unstructured triangular meshes saves more than 40% CPU time cost of that by the RK Jacobi scheme (the TVD-RK3 scheme) for both subsonic flow cases in this example. The contour plots of the pressure variable of the converged steady state solutions for these three schemes are shown in Figure 17. Comparable numerical steady states for these different iterative schemes are observed. Figure 18 presents the residue history of these three schemes with different CFL numbers. It is observed that the residue of iterations can settle down to tiny values at the level of round off errors, which again verifies the absolute convergence of the developed high order fast sweeping method on triangular meshes.

4 Concluding remarks

High order accuracy fast sweeping methods have been well developed on rectangular meshes to efficiently solve steady state solutions of hyperbolic PDEs. However, it was still a open problem how to design high order accuracy fast sweeping methods on unstructured meshes. In this paper,

$Ma = 0.8, \alpha = 1.25°$		
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.6	1007029	425623.98
0.7	863213	365297.07
0.8	755334	316719.83
0.9	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
1.0	1812610	742083.62
1.2	1510567	623277.44
1.4	1294810	533356.95
1.6	1208500	500951.51
1.7	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
1.0	598216	477581.42
1.2	498168	398820.98
1.4	426688	338455.70
1.6	373080	295754.15
1.7	Not convergent	-
$Ma = 0.2, \alpha = 1°$		
FE Jacobi scheme		
CFL number	iteration number	CPU time
0.4	959981	392068.22
0.6	639947	273666.77
0.7	Not convergent	-
RK Jacobi scheme		
CFL number	iteration number	CPU time
1.4	822721	331972.06
1.6	767878	317268.26
1.7	Not convergent	-
FE fast sweeping scheme		
CFL number	iteration number	CPU time
1.4	274184	203245.83
1.5	252120	186856.24
1.6	Not convergent	-

Table 7: Example 7, subsonic flows past an NACA0012 airfoil. Number of iterations and total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11}. CPU time unit: second.



(a) FE Jacobi, $Ma = 0.8$ (b) RK Jacobi, $Ma = 0.8$ (c) FE fast sweeping, $Ma = 0.8$

(d) FE Jacobi, $Ma = 0.2$ (e) RK Jacobi, $Ma = 0.2$ (f) FE fast sweeping, $Ma = 0.2$

Figure 18: Example 7, subsonic flows past an NACA0012 airfoil. The convergence history of the residue as a function of number of iterations for three schemes with different CFL numbers. (a), (b), (c): for the case of $Ma = 0.8$, angle of attack $\alpha = 1.25°$; (d) (e) (f): for the case of $Ma = 0.2$, angle of attack $\alpha = 1°$.

we develop a high order fixed-point fast sweeping method on unstructured triangular meshes for solving steady state solutions of hyperbolic conservation laws. Multiple reference points on the computational domain are introduced to order all the cells and form alternating sweeping directions on unstructured meshes. The local solver of the proposed fixed-point fast sweeping method is based on a fifth-order finite volume unstructured WENO scheme with unequal-sized sub-stencils, to achieve the absolute convergence of the iterations. Extensive numerical experiments, including solving difficult problems which are defined on complex domains or challenging for high order schemes to converge to steady states, show that the designed fixed-point fast sweeping scheme on unstructured meshes can significantly enlarge the CFL number of the forward Euler scheme with a high order WENO spatial discretization (e.g., the fifth order WENO scheme here) to the level of the TVD-RK3 scheme. As a result, up to 70% CPU time can be saved by using the proposed fast sweeping method rather than the TVD-RK3 scheme for iterations to converge to steady states of the WENO scheme. The iteration residues of the new absolutely convergent fast sweeping method on unstructured meshes converge to round off errors for all benchmark problems tested in this paper.

We focus on the Euler systems in this paper. However, the high order fixed-point fast sweeping WENO method on unstructured meshes developed here, along with the high order fixed-point fast sweeping WENO methods on structured meshes developed in the previous work, can be naturally extended to solve steady state problems of other hyperbolic PDEs. Since one of the key factors that fast sweeping methods can accelerate convergence of numerical simulations to steady state solutions is that they utilize alternating sweeping strategy to cover a family of characteristics of the PDEs in a certain direction simultaneously in each sweeping order, the methods are especially effective for solving steady state solutions of hyperbolic PDEs. For example, we will apply the high order fixed-point fast sweeping WENO methods in solving steady state problems of the shallow water equations, which arise often in many applications such as simulation of free surface flows in rivers and coastal areas, prediction of tides, storm surge levels and coastline changes from hurricanes and ocean currents, etc. Since the shallow water equations often admit steady state solutions in which the flux gradients of the PDEs are exactly balanced by the source terms, it is desirable that the numerical schemes for the shallow water equations have the well-balanced property, i.e., they can preserve exactly the steady state solutions up to machine error with relatively coarse meshes. Therefore, it will be important to develop fast sweeping WENO methods with the well-balanced property for the shallow water equations. We will use these effective approaches for developing well-balanced numerical methods in the literature, e.g. [4, 37, 6, 48, 38, 9]. Another interesting application is to simulate steady state flow of blood in blood vessels. Although blood is usually described as an incompressible fluid modeled by the three dimensional incompressible Navier-Stokes equations, for which information of solutions propagates in different ways from that for hyperbolic PDEs and this may lead to the loss of advantage of fast sweeping methods in accelerating iteration convergence [47, 27], in recent years interesting hyperbolic PDE systems have been developed and studied for blood flow. These hyperbolic PDE models were derived by averaging the full three dimensional incompressible Navier-Stokes equations over the vessel cross section, assuming that the characteristic axial length-scale is much longer than the radial one and that the flow is axisymmetric. See e.g. [16, 39, 1]. It is expected that the fast sweeping methods developed for these hyperbolic PDE models of blood flow will show efficient convergence to steady state solutions as that for the Euler systems in this paper. Also the well-balanced property should be preserved in the schemes [18]. These future work will be carried out in the next research.

References

[1] S.C. Anco, T.M. Garrido, A.P. Márquez, M.L. Gandarias, Exact solutions and conservation laws of a one-dimensional PDE model for a blood vessel, Chaos, Solitons and Fractals, 170 (2023), 113360.

[2] D. S. Balsara, S. Garain, V. Florinski, W. Boscheri, An efficient class of WENO schemes with adaptive order for unstructured meshes, J. Comput. Phys., 404 (2020), 109062.

[3] D.S. Balsara, S. Garain, C.-W. Shu, An efficient class of WENO schemes with adaptive order, J. Comput. Phys., 326 (2016), 780-804.

[4] A. Bermudez and M. E. Vazquez, Upwind methods for hyperbolic conservation laws with source terms, Computers and Fluids, 23 (1994), 1049-1071.

[5] R. Borges, M. Carmona, B. Costa and W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, J. Comput. Phys., 227 (2008), 3191-3211.

[6] S. Bryson, Y. Epshteyn, A. Kurganov and G. Petrova, Well-balanced positivity preserving central-upwind scheme on triangular grids for the Saint-Venant system, ESAIM: Mathematical Modelling and Numerical Analysis, 45 (2011), 423-446.

[7] M. Castro, B. Costa and W.S. Don, High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws, J. Comput. Phys., 230 (2011), 1766-1792.

[8] W. Chen, C.-S. Chou and C.-Y. Kao, Lax-Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws, J. Comput. Phys., 234 (2013), 452-471.

[9] M. Ciallella, D. Torlo and M. Ricchiuto, Arbitrary high order WENO finite volume scheme with flux globalization for moving equilibria preservation, J. Sci. Comput., 96 (2023), 53.

[10] T. H. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, (1990).

[11] R.M. Cummings, W.H. Mason, S.A. Morton and D.R. McDaniel, Applied Computational Aerodynamics: A Modern Engineering Approach, Cambridge University Press, Cambridge (2015).

[12] W.S. Don and R. Borges, Accuracy of the weighted essentially non-oscillatory conservative finite difference schemes, J. Comput. Phys., 250 (2013), 347-372.

[13] M. Dumbser and M. Käser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, J. Comput. Phys., 221 (2007), 693-723.

[14] M. Dumbser, M. Käser, V.A. Titarev and E.F. Toro, Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems, J. Comput. Phys., 226 (2007), 204-243.

[15] S. Fomel, S. Luo and H. Zhao, Fast sweeping method for the factored Eikonal equation, J. Comput. Phys., 228 (2009), 6440-6455.

[16] L. Formaggia, A. Quarteroni and A. Veneziani, Cardiovascular Mathematics: Modeling and Simulation of the Circulatory System, Modeling, Simulation and Applications, Springer, 2009.

[17] O. Friedrichs, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys., 144 (1998), 194-212.

[18] B. Ghitti, C. Berthon, M.H. Le and E.F. Toro, A fully well-balanced scheme for the 1D blood flow equations with friction source term, J. Comput. Phys., 421 (2020), 109750.

[19] S. Gottlieb, C.-W. Shu and E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Review, 43(1) (2001), 89-112.

[20] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes, III, J. Comput. Phys., 71 (1987), 231-303.

[21] C. Hu and C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys., 150 (1999), 97-127.

[22] G. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys., 126(1) (1996), 202-228.

[23] D. Levy, S. Nayak, C.-W. Shu and Y.-T. Zhang, *Central WENO schemes for Hamilton-Jacobi equations on triangular meshes*, SIAM Journal on Scientific Computing, 28 (2006), 2229-2247.

[24] D. Levy, G. Puppo and G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, Math. Model. Numer. Anal., 33 (1999), 547-571.

[25] D. Levy, G. Puppo and G. Russo, Compact central WENO schemes for multidimensional conservation laws, SIAM J. Sci. Comput., 22 (2000), 656-672.

[26] F. Li, C.-W. Shu, Y.-T. Zhang and H.-K. Zhao, A second order discontinuous Galerkin fast sweeping method for Eikonal equations, J. Comput. Phys., 227 (2008), 8191-8208.

[27] L. Li, J. Zhu and Y.-T. Zhang, Absolutely convergent fixed-point fast sweeping WENO methods for steady state of hyperbolic conservation laws, J. Comput. Phys., 443 (2021), 110516.

[28] L. Li, J. Zhu, C.-W. Shu and Y.-T. Zhang, A fixed-point fast sweeping WENO method with inverse Lax-Wendroff boundary treatment for steady state of hyperbolic conservation laws, Commun. Appl. Math. Comput., 5 (2023), 403-427. https://doi.org/10.1007/s42967-021-00179-6

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

[30] Y. Liu and Y.-T. Zhang, *A robust reconstruction for unstructured WENO schemes*, Journal of Scientific Computing, 54 (2013), 603-621.

[31] H. Luo, J.D. Baum and R. Löhner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, J. Comput. Phys., 225 (2007), 686-713.

[32] H. Luo, J.D. Baum and R. Löhner, On the computation of steady-state compressible flows using a discontinuous Galerkin method, Int. J. Numer. Methods. Eng., 73 (2008), 597-623.

[33] Z. M. Miksis and Y.-T. Zhang, Sparse-grid implementation of fixed-point fast sweeping WENO schemes for Eikonal equations, Commun. Appl. Math. Comput., 6 (2024), 3-29. https://doi.org/10.1007/s42967-022-00209-x

[34] J. Qian, Y.-T. Zhang and H.-K. Zhao, Fast sweeping methods for Eikonal equations on triangular meshes, SIAM J. Numer. Anal., 45 (2007), 83-107.

[35] J. Qian, Y.-T. Zhang and H.-K. Zhao, A fast sweeping method for static convex Hamilton-Jacobi equations, J. Sci. Comput., 31 (2007), 237-271.

[36] F. Qin, Y. Luo, K. B. Olsen, W. Cai and G. T. Schuster, Finite difference solution of the eikonal equation along expanding wavefronts, Geophys., 57 (1992), 478-487.

[37] M. Ricchiuto, R. Abgrall and H. Deconinck, Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes, J. Comput. Phys., 222 (2007), 287-331.

[38] K.A. Schneider, J.M. Gallardo, D.S. Balsara, B. Nkonga and C. Parés, Multidimensional approximate Riemann solvers for hyperbolic nonconservative systems. Applications to shallow water systems, J. Comput. Phys., 444 (2021), 110547.

[39] W. Sheng, Q. Zhang, Y. Zheng, The Riemann problem for a blood flow model in arteries, Commun. Comput. Phys., 27 (2020), 227-250.

[40] J. Shi, Y.-T. Zhang and C.-W. Shu, Resolution of high order WENO schemes for complicated flow structures, J. Comput. Phys., 186(2) (2003), 690-696.

[41] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, Berlin, 1998, 325-432.

[42] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Rev., 51(1) (2009), 82-126.

[43] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys., 77(2) (1988), 439-471.

[44] V.A. Titarev, P. Tsoutsanis and D. Drikakis, WENO schemes for mixed-element unstructured meshes, Commun. Comput. Phys., 8 (2010), 585-609.

[45] J. van Trier and W. W. Symes, Upwind finite-difference calculation of traveltimes, Geophys., 56 (1991), 812-821.

[46] L. Wu and Y.-T. Zhang, A third order fast sweeping method with linear computational complexity for Eikonal equations, J. Sci. Comput. 62 (2015), 198-229.

[47] L. Wu, Y.-T. Zhang, S. Zhang and C.-W. Shu, High order fixed-point sweeping WENO methods for steady state of hyperbolic conservation laws and its convergence study, Commun. Comput. Phys., 20(4) (2016), 835-869.

[48] Y. Xing and C.-W. Shu, A survey of high order schemes for the shallow water equations, J. Math. Study, 47 (2014), 221-249.

[49] T. Xiong, M. Zhang, Y.-T. Zhang and C.-W. Shu, Fast sweeping fifth order WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment, J. Sci. Comput., 45(1) (2010), 514-536.

[50] S. Zhang, S. Jiang and C.-W. Shu, *Improvement of convergence to steady state solutions of Euler equations with the WENO schemes*, Journal of Scientific Computing, 47 (2011), 216-238.

[51] S. Zhang and C.-W. Shu, *A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions*, Journal of Scientific Computing, 31 (2007), 273-305.

[52] Y.-T. Zhang, S. Chen, F. Li, H. Zhao and C.-W. Shu, Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations, SIAM J. Sci. Comput. 33 (2011), 1873-1896.

[53] Y.-T. Zhang and C.-W. Shu, High order WENO schemes for Hamilton-Jacobi equations on triangular meshes, SIAM Journal on Scientific Computing, 24 (2003), 1005-1030.

[54] Y.-T. Zhang and C.-W. Shu, Third order WENO scheme on three dimensional tetrahedral meshes, Commun. Comput. Phys., 5 (2009), 836-848.

[55] Y.-T. Zhang, H.-K. Zhao and S. Chen, Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations, Methods. Appl. Anal., 13(3) (2006), 299-320.

[56] Y.-T. Zhang, H.-K. Zhao and J. Qian, High order fast sweeping methods for static Hamilton-Jacobi equations, J. Sci. Comput., 29(1) (2006), 25-56.

[57] H.-K. Zhao, A fast sweeping method for Eikonal equations, Math. Comput., 74 (2005), 603-627.

[58] J. Zhu and J. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, J. Comput. Phys., 318 (2016), 110-121.

[59] J. Zhu and J. Qiu, New finite volume weighted essentially non-oscillatory schemes on triangular meshes, SIAM J. Sci. Comput., 40 (2018), 903-928.

[60] J. Zhu and C.-W. Shu, Numerical study on the convergence to steady state solutions of a new class of high order WENO schemes, Journal of Computational Physics, 349 (2017), 80-96.

[61] J. Zhu and C.-W. Shu, Numerical study on the convergence to steady-state solutions of a new class of finite volume WENO schemes: triangular meshes, Shock Waves, 29 (2019), 3-25.

[62] J. Zhu, X. Zhong, C.-W. Shu and J. Qiu, Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes, J. Comput. Phys., 248 (2013), 200-220.

Declaration of interests

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

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: