

# A fixed-point fast sweeping WENO method with inverse Lax-Wendroff boundary treatment for steady state of hyperbolic conservation laws

Liang Li<sup>1</sup>, Jun Zhu<sup>2</sup>, Chi-Wang Shu<sup>3</sup>, Yong-Tao Zhang<sup>4</sup>

## Abstract

Fixed-point fast sweeping WENO methods are a class of efficient high order numerical methods to solve steady state solutions of hyperbolic partial differential equations (PDEs). The Gauss-Seidel iterations and alternating sweeping strategy are used to cover characteristics of hyperbolic PDEs in each sweeping order to achieve fast convergence rate to steady state solutions. A nice property of fixed-point fast sweeping WENO methods which distinguishes them from other fast sweeping methods is that they are explicit and do not require inverse operation of nonlinear local systems. Hence they are easy to be applied to a general hyperbolic system. In order to deal with the difficulties associated with numerical boundary treatment when high order finite difference methods on a Cartesian mesh are used to solve hyperbolic PDEs on complex domains, inverse Lax-Wendroff (ILW) procedures were developed as a very effective approach in the literature. In this paper, we combine a fifth order fixed-point fast sweeping WENO method with an inverse Lax-Wendroff (ILW) procedure to solve steady state solution of hyperbolic conservation laws on complex computing regions. Numerical experiments are performed to test the method in solving various problems including the cases with the physical boundary not aligned with the grids. Numerical results show high order accuracy and good performance of the method. Furthermore, the method is compared with the popular third order total variation diminishing Rung-Kutta (TVD-RK3) time-marching method for steady state computations. Numerical examples show that for most of examples, the fixed-point fast sweeping method saves more than half CPU time costs than TVD-RK3 in order to converge to steady state solutions.

**Key Words:** Fixed-point fast sweeping methods, multi-resolution WENO schemes, steady state, inverse Lax-Wendroff procedure, convergence.

## 1 Introduction

Steady state problems of hyperbolic conservation laws are important mathematical models arising in many applications of compressible fluid dynamics. High order accuracy WENO schemes are a class of popular numerical methods for spatial discretizations of nonlinear hyperbolic conservation laws, e.g., see [19, 17, 18, 7, 15, 1]. After spatial discretizations of a steady state problem by a high order WENO scheme, a large nonlinear algebraic system is formed. One of the important factors which determines efficiency of the numerical simulations is how to design fast algorithms for

---

<sup>1</sup>College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China. E-mail: liliangnuaa@163.com

<sup>2</sup>College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China. E-mail: zhujun@nuaa.edu.cn

<sup>3</sup>Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: chi-wang\_shu@brown.edu. Research supported in part by AFOSR grant FA9550-20-1-0055 and NSF grant DMS-2010107.

<sup>4</sup>Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: yzhang10@nd.edu

solving the large nonlinear system. For both linear and nonlinear stability of high order schemes, a common approach is to use the third order total variation diminishing Runge-Kutta (TVD-RK3) scheme [19, 5] to time-march numerical solution to steady state. In order to improve computational efficiency, explicit fast sweeping WENO methods (e.g., [30, 29, 25, 24]) were developed to accelerate the convergence to steady state solutions of high order schemes. Fast sweeping methods are a class of efficient iterative methods developed to 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. Coupled with the Gauss-Seidel iterations, these methods can achieve a fast convergence speed. Implicit fast sweeping methods are very efficient and often have linear computational complexity (see e.g. [31, 13, 14, 4]), however the algorithms are much more complicated than explicit methods when they are coupled with high order schemes [9, 28, 23]. In order to solve steady state problems of the nonlinear hyperbolic conservation laws by high order WENO fast sweeping schemes, the fixed-point fast sweeping methods were applied [3, 24] which were based on the fixed-point fast sweeping methods for static Hamilton-Jacobi equations [29]. This kind of fast sweeping methods have nice properties such as that they have explicit forms and do not involve inverse operation of nonlinear local systems, and they can be adopted to solve general hyperbolic systems with any monotone numerical fluxes and high order WENO approximations easily. Numerical experiments show that the methods converge to steady state solutions much faster than regular time-marching approach. Since it shows that the WENO schemes with unequal-sized substencils [33, 32] have very nice convergence property in solving steady state problems, e.g., the high order multi-resolution WENO schemes [34], the fixed-point fast sweeping methods were combined with multi-resolution WENO scheme in [10] to overcome the issue that fast sweeping iteration residue does not fully converge to values of round-off error level for difficult steady state problems, and a fully convergent fast sweeping WENO method (called absolutely convergent fixed-point fast sweeping WENO method) was obtained.

When the computational domain has complex geometries, the efficient finite difference method is difficult to be applied, because the Cartesian grid points are usually not located on the physical boundary and the boundary may intersect the grids in an arbitrary fashion. There are many excellent approaches to deal with this. For example, an accurate implementation of solid wall boundary conditions in curved geometries was developed in [8], and a Cartesian embedded boundary method based on the finite difference formulation was designed in [20]. In [6, 25], the inverse Lax-Wendroff (ILW) numerical boundary condition procedure was developed for solving static Hamilton-Jacobi equations. The effective ILW methods were successfully developed to solve hyperbolic conservation laws on complex geometry domains in [21, 22, 11].

In this paper, we combine the absolutely convergent fixed-point fast sweeping WENO method [10] with the inverse Lax-Wendroff procedure to solve steady state solution of hyperbolic conservation laws on complex domains. Numerical experiments are performed to test the method in solving various problems including the cases with the physical boundary not aligned with the grids. It is shown that the absolutely convergent fixed-point fast sweeping method with the ILW procedure is more efficient than other popular approach such as the third order total variation diminishing Runge-Kutta (TVD-RK3) time-marching method to converge to steady state solutions of the fifth order multi-resolution WENO scheme. Furthermore, it can drive the residue of iterations to converge to machine zero / round off errors for all benchmark problems tested. The rest of the paper is organized as follows. The detailed numerical algorithm is described in Section 2. In Section 3 we provide extensive numerical experiments to test and study the proposed method, and to perform comparisons of different methods. Concluding remarks are given in Section 4.

## 2 Numerical methods

We consider steady state problems of hyperbolic conservation laws

$$f(u)_x + g(u)_y = R, \quad (x, y) \in \Omega, \quad (2.1)$$

on a bounded domain  $\Omega$  with appropriate boundary conditions prescribed on  $\partial\Omega$ . Here  $u$  is the vector of the unknown conservative variables,  $f(u)$  and  $g(u)$  are the vectors of flux functions, and  $R$  is the source term. For example, the steady Euler system of equations in compressible fluid dynamics has that  $u = (\rho, \rho u, \rho v, E)^T$ ,  $f(u) = (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T$ , and  $g(u) = (\rho v, \rho uv, \rho v^2 + p, v(E + p))^T$ . Here  $\rho$  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. In the following we first describe the absolutely convergent fixed-point fast sweeping WENO method [10] to solve (2.1).

### 2.1 A fifth order absolutely convergent fixed-point fast sweeping WENO method

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 grid points. We partition  $\Omega$  by a uniform Cartesian mesh  $\{(x_i, y_j)\}$  with grid size  $\Delta x$  in the  $x$ -direction and  $\Delta y$  in the  $y$ -direction. Since high order WENO schemes with unequal-sized substencils [35, 34] show nice convergence property in solving steady state problems, a fifth order multi-resolution WENO scheme [33] is used in [10] for the spatial discretization. The procedure is outlined here. For upwinding and linear stability of the scheme, we perform a flux splitting. For example the  $x$ -direction flux  $f(u)$  is split into  $f(u) = f^+(u) + f^-(u)$ , which satisfies the condition that  $\frac{df^+(u)}{du} \geq 0$  and  $\frac{df^-(u)}{du} \leq 0$  for the scalar case, or the corresponding eigenvalues of Jacobian matrix are positive / negative for the system case with a local characteristic decomposition. Here a simple Lax-Friedrichs flux splitting  $f^\pm(u) = \frac{1}{2}(f(u) \pm \alpha u)$  is used, where  $\alpha = \max_u |f'(u)|$ . Then each of them is approximated separately by the fifth order multi-resolution WENO scheme to obtain numerical fluxes  $\hat{f}_{i+1/2,j}^+$  and  $\hat{f}_{i+1/2,j}^-$  at the point  $(x_{i+1/2}, y_j)$  respectively, where  $x_{i+1/2} = (x_i + x_{i+1})/2$ . The final numerical flux is  $\hat{f}_{i+1/2,j} = \hat{f}_{i+1/2,j}^+ + \hat{f}_{i+1/2,j}^-$ . Specifically, the reconstructed numerical flux  $\hat{f}_{i+1/2,j}^+$  is

$$\hat{f}_{i+1/2,j}^+ = \sum_{l_1=1}^3 \omega_{l_1} p_{l_1}(x_{i+1/2}). \quad (2.2)$$

The nonlinear weights are computed as

$$\omega_{l_1} = \frac{\bar{\omega}_{l_1}}{\sum_{l_2=1}^3 \bar{\omega}_{l_2}}, \quad \bar{\omega}_{l_1} = \gamma_{l_1,3} \left(1 + \frac{\tau}{\varepsilon + \beta_{l_1}}\right), \quad l_1 = 1, 2, 3,$$

where  $\varepsilon$  is taken as  $\varepsilon = 10^{-6}$  to avoid the case that the denominator becomes zero. The details on the linear weights  $\gamma_{l_1,3}$ , computations of smoothness indicators  $\beta_{l_1}$ , the quantity  $\tau$  and reconstruction polynomials  $p_{l_1}(x)$  can be found in [33, 10]. The reconstruction of the numerical flux  $\hat{f}_{i+1/2,j}^-$  is mirror-symmetric with respect to  $x_{i+1/2}$ , and the similar procedure is applied to the  $y$ -direction flux  $g(u)$ .

After the fifth order multi-resolution WENO finite difference spatial discretization, we obtain a nonlinear algebraic system

$$0 = -(\hat{f}_{i+1/2,j} - \hat{f}_{i-1/2,j})/\Delta x - (\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2})/\Delta y + R(u_{ij}, x_i, y_j), \quad i = 1, \dots, N; j = 1, \dots, M, \quad (2.3)$$

where  $N$  and  $M$  are the number of grid points in the  $x$  and  $y$  directions respectively.  $\hat{f}_{i+1/2,j}$  and  $\hat{g}_{i,j+1/2}$  are the numerical fluxes. The right-hand-side (RHS) of (2.3) is a nonlinear function of the numerical values at the grid points of the computational stencils. We denote it by  $L$ . To solve (2.3), the forward Euler type fixed-point fast sweeping method [29, 24, 10] is used. It has the following form

$$u_{ij}^{n+1} = u_{ij}^n + \frac{\gamma}{\alpha_x/\Delta x + \alpha_y/\Delta y} L(u_{i-r,j}^*, \dots, u_{i+s,j}^*; u_{i,j}^n; u_{i,j-r}^*, \dots, u_{i,j+s}^*),$$

$$i = i_1, \dots, i_N; j = j_1, \dots, j_M. \quad (2.4)$$

$u_{ij}^n$  is the  $n$ -th step iteration value at the grid point  $(x_i, y_j)$ .  $\gamma$  is the Courant-Friedrichs-Lewy (CFL) number and  $\frac{\gamma}{\alpha_x/\Delta x + \alpha_y/\Delta y}$  is corresponding to the time step size  $\Delta t_n$ .  $\alpha_x = \max_u |f'(u)|$  and  $\alpha_y = \max_u |g'(u)|$  represent the maximum characteristic speeds in each spatial direction.  $r, s$  are values which depend on the stencil of the WENO reconstruction. For the fifth order multi-resolution WENO scheme used here, we have  $r = s = 3$ . In the fast sweeping method, the Gauss-Seidel iterations and alternating direction sweepings are used. The Gauss-Seidel philosophy requires that the newest numerical values of  $u$  are used in the fifth order multi-resolution WENO reconstruction stencils whenever they are available. The iterations do not just proceed in only one direction  $i = 1 : N, j = 1 : M$ , but in the following four alternating directions repeatedly,

- (1)  $i = 1 : N, j = 1 : M$ ;
- (2)  $i = N : 1, j = 1 : M$ ;
- (3)  $i = N : 1, j = M : 1$ ;
- (4)  $i = 1 : N, j = M : 1$ .

This is marked as “ $i = i_1, \dots, i_N; j = j_1, \dots, j_M$ ” in the scheme (2.4) to show the alternating iteration directions. The purpose of using alternating direction sweepings is to utilize the characteristics property of hyperbolic PDEs. By combining with the Gauss-Seidel philosophy, a fast convergence to steady state solution is achieved, as shown in the following numerical experiments. In the Gauss-Seidel iterations, the newest numerical values on the stencil of the WENO scheme are used if they are available. This fact is shown by the notation  $u^*$  in the scheme (2.4) to represent the numerical values in the WENO stencil, with the understanding that  $u_{k,l}^*$  could be  $u_{k,l}^n$  or  $u_{k,l}^{n+1}$ , depending on the current sweeping direction.

## 2.2 The inverse Lax-Wendroff boundary treatment

In this section, we outline the inverse Lax-Wendroff numerical boundary treatment in both one-dimensional (1D) and two-dimensional (2D) problems. The ILW methods have been developed and studied extensively in e.g. [21, 22], so here we just give a very brief description of the ILW methods, and the details are omitted and referred to [21, 22]. The only difference from [21, 22] is that in this paper, the fifth order multi-resolution WENO method is used for the extrapolation step in the ILW methods, and we provide detailed description about that.

### 2.2.1 One-dimensional case

First we consider 1D compressible steady state Euler equations

$$F(U)_x = R(U, x, y), \quad (2.5)$$

where the conservative variables  $U \triangleq (U_1, U_2, U_3)^T = (\rho, \rho u, E)^T$ , and the flux  $F(U) = (\rho u, \rho u^2 + p, u(E + p))^T = (U_2, (\gamma' - 1)U_3 + \frac{3-\gamma'}{2}\frac{U_2^2}{U_1}, (\gamma'U_3 - \frac{\gamma'-1}{2}\frac{U_2^2}{U_1})\frac{U_2}{U_1})^T$ .  $\rho, u, p$  and  $E$  denote the density, velocity, pressure and total energy, respectively.  $R(U, x, y) = (R_1(U, x, y), R_2(U, x, y), R_3(U, x, y))^T$  are the source terms. The equation of state has the form  $E = \frac{p}{\gamma'-1} + \frac{1}{2}\rho u^2$  where  $\gamma'$  is the specific heat ratio. The sound speed is  $c = \sqrt{\gamma'p/\rho}$ . Without loss of generality, the computational domain is taken as  $x \in (-1, 1)$ .

The computational domain is partitioned by a uniform mesh

$$-1 + h/2 = x_0 < x_1 < \dots < x_N = 1 - h/2,$$

where  $h$  is the size of the grid. We proceed as in [21] to compute numerical values on ghost points. Here the computation of  $U(x_{N+1})$  is taken as an example. Other values such as  $U(x_{-1}), U(x_{-2}), \dots$  and  $U(x_{N+2}), U(x_{N+3}), \dots$  are treated in a similar way. By a fifth order Taylor approximation,

$$U_m(x_{N+1}) = \sum_{k=0}^4 \frac{(x_{N+1} - 1)^k}{k!} U_m^{*(k)}, \quad m = 1, 2, 3, \quad (2.6)$$

where  $U_m^{*(k)}$  is a  $(5 - k)$  th order approximation of the spatial derivative  $\frac{\partial^k U_m}{\partial x^k} |_{x=1}$ . Then a local characteristic decomposition is performed to take care of the inflow and outflow boundary conditions. At the boundary, the Jacobian matrix of the flux is denoted by  $A_\perp(U(x_N)) = \frac{\partial F(U)}{\partial U} |_{U=U(x_N)}$ , and  $A_\perp(U(x_N))$  has three eigenvalues  $\lambda_1 = u(x_N) - c(x_N), \lambda_2 = u(x_N), \lambda_3 = u(x_N) + c(x_N)$  and a complete set of left eigenvectors  $l_1(U(x_N)), l_2(U(x_N)), l_3(U(x_N))$ . The local characteristic variables  $V_m$  at interior grid points near the boundary are defined by

$$V_m(x_j) = l_m^T(U(x_N))U(x_j), \quad j = N, N - 1, \dots, N - 4; \quad m = 1, 2, 3.$$

Using the numerical values on these interior points, we can find fourth-order polynomials to compute extrapolation values for both functions and their derivatives at the boundary  $x = 1$ , i.e.,  $V_m^{*(k)}, m = 1, 2, 3; k = 0, 1, 2, 3, 4$ . For a more robust algorithm to deal with possible discontinuities near the boundary, we use a multi-resolution WENO type extrapolation which is described in details below. Then, following the procedure in [21], these extrapolation values  $V_m^{*(k)}$  are coupled with the inflow boundary conditions and the PDEs itself to compute these values  $U_m^{*(k)}$  which are needed in (2.6).

### 1D multi-resolution WENO extrapolation

Here we describe the multi-resolution WENO extrapolations for the 1D case, i.e., construct a  $(5 - k)$ th order approximation  $V_m^{*(k)}$  at the boundary  $x = 1$ , using the numerical values at the interior grid points of the domain  $V_m(x_j), j = N - 4, \dots, N$ . For the simplicity of notation, we omit the subscript  $m$  and use  $V_j$  to denote  $V_m(x_j)$  for a specific  $m$ . Three substencils are used to construct WENO extrapolations, i.e., the substencils  $S_1 = \{x_N\}, S_2 = \{x_{N-2}, x_{N-1}, x_N\}$  and  $S_3 = \{x_{N-4}, \dots, x_N\}$ . The extrapolation procedure is given in the following.

**Step 1.** On each substencil, a Lagrange polynomial  $q_r(x)$  of degree  $2r - 2$  is constructed such that  $q_r(x_j) = V_j, j = N - 2r + 2, \dots, N; r = 1, 2, 3$ .

**Step 2.** Obtain equivalent expressions for these polynomials as in multi-resolution WENO reconstruction procedure [33]. Namely, take  $p_1(x) = q_1(x)$ , and construct

$$p_2(x) = \frac{1}{\gamma_{2,2}} q_2(x) - \frac{\gamma_{1,2}}{\gamma_{2,2}} p_1(x),$$

$$p_3(x) = \frac{1}{\gamma_{3,3}} q_3(x) - \frac{\gamma_{2,3}}{\gamma_{3,3}} p_2(x) - \frac{\gamma_{1,3}}{\gamma_{3,3}} p_1(x),$$

with  $\sum_{l=1}^2 \gamma_{l,2} = 1$ ,  $\sum_{l=1}^3 \gamma_{l,3} = 1$  and  $\gamma_{2,2} \neq 0, \gamma_{3,3} \neq 0$ . We take  $\gamma_{1,2} = 1/11$ ,  $\gamma_{2,2} = 10/11$ ,  $\gamma_{1,3} = 1/111$ ,  $\gamma_{2,3} = 10/111$ ,  $\gamma_{3,3} = 100/111$  as in [33].

**Step 3.** Compute the smoothness indicators  $\beta_l$  which measure how smooth the functions  $p_l(x)$  for  $l = 1, 2, 3$  are in the interval  $[x_N, x_{N+1}]$ . We use the same recipe for the smoothness indicators as in [7, 17]:

$$\beta_l = \sum_{\alpha=1}^{2l-2} \int_{x_N}^{x_{N+1}} h^{2\alpha-1} \left( \frac{d^\alpha p_l(x)}{dx^\alpha} \right)^2 dx, \quad l = 1, 2, 3.$$

Notice here  $\beta_1 = 0$  which is a simplification of that in [33]. Numerical experiments in this paper show that high order accuracy and resolution, and nonlinear stability of the computations are still maintained by taking this simplified value of  $\beta_1$ .

**Step 4.** Compute the nonlinear weights based on the linear weights and the smoothness indicators. The WENO-Z type nonlinear weights as in [1, 2] are used. First a quantity  $\tau$  which depends on the absolute differences between the smoothness indicators is calculated:  $\tau = \left( \frac{|\beta_3 - \beta_1| + |\beta_3 - \beta_2|}{2} \right)^2$ . The nonlinear weights are

$$\omega_{l_1} = \frac{\bar{\omega}_{l_1}}{\sum_{l_2=1}^3 \bar{\omega}_{l_2}}, \quad \bar{\omega}_{l_1} = \gamma_{l_1,3} \left( 1 + \frac{\tau}{\varepsilon + \beta_{l_1}} \right), \quad l_1 = 1, 2, 3.$$

Here  $\varepsilon = 10^{-6}$  is used to avoid the case that the denominator becomes zero. Then the final  $(5-k)$ th order WENO approximation  $V^{*(k)}$  at the boundary  $x = 1$  is

$$V^{*(k)} = \sum_{r=1}^3 \omega_r \frac{d^k p_r(x)}{dx^k} \Big|_{x=1}. \quad (2.7)$$

## 2.2.2 Two-dimensional case

We consider 2D compressible steady state Euler equations

$$F(U)_x + G(U)_y = 0, \quad (x, y) \in \Omega, \quad (2.8)$$

with appropriate boundary conditions, where  $U = (U_1, U_2, U_3, U_4)^T = (\rho, \rho u, \rho v, E)^T$ ,  $F(U) = (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T$ ,  $G(U) = (\rho v, \rho uv, \rho v^2 + p, v(E + p))^T$ .  $\rho, u, v, p$  and  $E$  denote the density,  $x$ -velocity,  $y$ -velocity, pressure, and total energy, respectively. The equation of state has the form  $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2)$ , where  $\gamma=1.4$  for the case of air.

We follow the procedure of the Inverse Lax-Wendroff boundary treatment in [21]. For example a ghost point  $P = (x_i, y_j)$ , we first identify the point  $P_0 = (x_0, y_0)$  on the boundary  $\Gamma = \partial\Omega$  such that the outward normal  $n(P_0)$  at  $P_0$  goes through  $P$ . See Figure 1 for an illustration. The coordinates of the point  $P_0$  and the outward normal  $n(P_0)$  are obtained from the given geometry of  $\Gamma$ . We set up a local coordinate system  $(\hat{x}, \hat{y})$  at  $P_0$  by

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix},$$

where  $\theta$  is the angle between the outward normal  $n(P_0)$  and the  $x$ -axis and  $T$  is the corresponding rotation matrix. The  $\hat{x}$ -axis has the same direction as  $n(P_0)$  and the  $\hat{y}$ -axis is the tangential direction. Under this local coordinate system, the Euler equations (2.8) are transformed to  $F(\hat{U})_{\hat{x}} + G(\hat{U})_{\hat{y}} = 0$ , where  $\hat{U} = (\hat{U}_1, \hat{U}_2, \hat{U}_3, \hat{U}_4)^T$ . For the fifth order boundary treatment, the numerical value at the ghost point P is approximated by the Taylor expansion

$$(\hat{U}_m)_P = \sum_{k=0}^4 \frac{\Delta^k}{k!} \hat{U}_m^{*(k)}, \quad m = 1, \dots, 4. \quad (2.9)$$

Here  $\Delta$  is the  $\hat{x}$ -coordinate of the ghost point P and  $\hat{U}_m^{*(k)}$  is a  $(5 - k)$ th order approximation of the normal derivative  $\frac{\partial^k \hat{U}_m}{\partial \hat{x}^k} \Big|_{(x,y)=P_0}$ . To decide the inflow or outflow boundary conditions to apply in the ILW procedure, a local characteristic decomposition is performed. We use the numerical values  $\hat{U}_{(0)}$  of a grid point which is the closest to  $P_0$  among all the grid points inside the domain  $\Omega$ , to form the Jacobian matrix of the normal flux by  $A_{\perp}(\hat{U}_{(0)}) = \frac{\partial F(\hat{U})}{\partial(\hat{U})} \Big|_{\hat{U}=\hat{U}_{(0)}}$ .  $A_{\perp}(\hat{U}_{(0)})$  has four eigenvalues  $\lambda_1 = \hat{u}_0 - c_0$ ,  $\lambda_2 = \lambda_3 = \hat{u}_0$ ,  $\lambda_4 = \hat{u}_0 + c_0$  and a complete set of left eigenvectors  $l_m(\hat{U}_{(0)})$ ,  $m = 1, \dots, 4$ , where  $\hat{u}_0$  and  $c_0$  are the corresponding  $x$ -velocity and the sound speed. The local characteristic variables  $V_m$  at interior grid points near  $P_0$  are given by

$$(V_m)_{\mu,\nu} = l_m(\hat{U}_{(0)})\hat{U}_{\mu,\nu}, \quad m = 1, \dots, 4, \quad (x_{\mu}, y_{\nu}) \in S,$$

where  $S$  is the stencil for the fifth order multi-resolution WENO extrapolation, which is used to compute  $V_m^{*(k)}$ , the  $(5 - k)$ th order approximation of  $\frac{\partial^k V_m}{\partial \hat{x}^k} \Big|_{P_0}$ . The fifth order multi-resolution WENO extrapolation will be described in the following. Then, following the procedure in [21, 22], these extrapolation values  $V_m^{*(k)}$  are coupled with the inflow boundary conditions and the PDEs itself to compute these values  $\hat{U}_m^{*(k)}$  which are needed in the Taylor expansion (2.9).

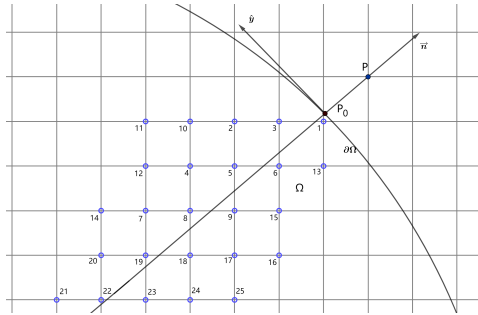


Figure 1: The stencil  $S$  for two-dimensional fifth order multi-resolution WENO extrapolation.

## 2D multi-resolution WENO extrapolation

Here we describe the two dimensional fifth order multi-resolution WENO extrapolation. Following [22, 11], interior grid points in the neighborhood of the point  $P_0$  which are distributed around the normal  $n(P_0)$  ( $\hat{x}$ -axis) are selected as the stencil  $S$  of the WENO extrapolation. For the fifth order multi-resolution WENO extrapolation used here,  $S$  contains three substencils  $S_1, S_2, S_3$  where  $S_3$  is the same as  $S$ , and  $S_1$  has just one point while  $S_2$  has nine points and  $S_3$  is composed of twenty five points to guarantee the fifth order accuracy for the extrapolation function. See Figure 1 for an illustration of the stencil, where  $S_1$  contains the point 1, and  $S_2$  contains the points 1 to 9 while  $S_3$  includes the points 1 to 25. Given the numerical values of the variables  $V_m$  on the stencil  $S$ , we construct a constant polynomial  $q_1(x, y)$  based on  $S_1$ , a degree 2 polynomial  $q_2(x, y)$  based on  $S_2$ , and a degree 4 polynomial  $q_3(x, y)$  based on  $S_3$ . Since there are more given values than the degrees of freedom, the least square method is used to find the polynomials  $q_2$  and  $q_3$ . Following the multiresolution WENO reconstruction idea in [33], we obtain equivalent expressions for these polynomials, i.e.,  $p_1(x, y) = q_1(x, y)$ ,  $p_2(x, y) = \frac{1}{\gamma_{2,2}}q_2(x, y) - \frac{\gamma_{1,2}}{\gamma_{2,2}}p_1(x, y)$ , and  $p_3(x, y) = \frac{1}{\gamma_{3,3}}q_3(x, y) - \frac{\gamma_{1,3}}{\gamma_{3,3}}p_1(x, y) - \frac{\gamma_{2,3}}{\gamma_{3,3}}p_2(x, y)$ . The linear weights are taken as  $\gamma_{1,2} = 1/11$ ,  $\gamma_{2,2} = 10/11$ , and  $\gamma_{1,3} = 1/111$ ,  $\gamma_{2,3} = 10/111$ ,  $\gamma_{3,3} = 100/111$ . Then the multi-resolution WENO

extrapolation gives

$$V_m^{*(k)} = \sum_{r=1}^3 \omega_r \frac{\partial^k p_r(x, y)}{\partial \hat{x}^k} \Big|_{(x, y) = (x_0, y_0)}, \quad k = 0, \dots, 4. \quad (2.10)$$

Here the nonlinear weights are computed as the following:

$$\omega_{l_1} = \frac{\bar{\omega}_{l_1}}{\sum_{l_2=1}^3 \bar{\omega}_{l_2}}, \quad \bar{\omega}_{l_1} = \gamma_{l_1, 3} \left(1 + \frac{\tau}{\varepsilon + \beta_{l_1}}\right), \quad l_1 = 1, 2, 3.$$

The quantity  $\tau = \left(\frac{|\beta_3 - \beta_1| + |\beta_3 - \beta_2|}{2}\right)^2$ , and the smoothness indicators

$$\beta_l = \sum_{|\alpha|=1}^{2l-2} \int_{x_0 - \frac{\Delta x}{2}}^{x_0 + \frac{\Delta x}{2}} \int_{y_0 - \frac{\Delta y}{2}}^{y_0 + \frac{\Delta y}{2}} (\Delta x \Delta y)^{|\alpha|-1} (D^\alpha p_l(x, y))^2 dx dy, \quad l = 1, 2, 3.$$

Here  $\alpha$  is the multi-index notation for partial derivative operator  $D^\alpha$ , and  $\varepsilon = 10^{-6}$  is used to avoid the case that the denominator becomes zero. As that in the 1D case, here the smoothness indicator  $\beta_1 = 0$  which is a simplification of that in [33]. Numerical experiments in the following section verify that high order accuracy and resolution, and nonlinear stability of the simulations are maintained well by taking this simplified value of  $\beta_1$ .

### 3 Numerical experiments

In this section we perform numerical experiments to test the performance of the fixed-point fast sweeping multi-resolution WENO method with the inverse Lax-Wendroff boundary treatment for steady state computations. A popular approach to solve steady state problems is to apply time marching approach to the system (2.3). For example, the TVD-RK3 scheme in [19, 5] has been broadly used for time marching high order WENO schemes to steady states, and both linear and nonlinear stability of high order WENO schemes are maintained. To show the advantage of the fixed-point fast sweeping multi-resolution WENO method with the inverse Lax-Wendroff boundary treatment, the computational efficiency of the TVD-RK3 scheme and this fast sweeping scheme is compared. Time marching methods are actually Jacobi type fixed-point iterations (see e.g. [29, 24, 10]). For the presentation purpose, the fixed-point fast sweeping method is called ‘‘FE fast sweeping scheme’’, and the TVD-RK3 time marching scheme ‘‘RK Jacobi scheme’’ in this section. The convergence of the iterations is measured by the average residue which is defined as

$$ResA = \sum_{i=1}^{\mathcal{N}} \frac{|R_i|}{\mathcal{N}}, \quad (3.1)$$

where the local residue at the grid point  $i$  is

$$R_i = \frac{u_i^{n+1} - u_i^n}{\Delta t_n}.$$

$\mathcal{N}$  is the total number of grid points and  $n$  is the iteration step.  $\Delta t_n = \frac{\gamma}{\alpha_x / \Delta x + \alpha_y / \Delta y}$ . For the 2D Euler system, the average residue is defined as  $ResA = \sum_{i=1}^{\mathcal{N}} \frac{|R1_i| + |R2_i| + |R3_i| + |R4_i|}{4\mathcal{N}}$ . Here  $R*_i$ ’s are the local residuals of the conservative variables, namely,  $R1_i = \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t_n}$ ,  $R2_i = \frac{(\rho u)_i^{n+1} - (\rho u)_i^n}{\Delta t_n}$ ,  $R3_i =$



RK Jacobi, $\gamma=1.0$						
N	L <sub>1</sub> error	L <sub>1</sub> order	L <sub>∞</sub> error	L <sub>∞</sub> order	iter#	CPU time
20	3.54E-08	-	5.88E-08	-	273	7.0E-03
40	8.07E-10	5.46	2.29E-09	4.68	345	1.7E-02
80	2.28E-11	5.15	7.46E-11	4.94	618	6.3E-02
160	6.81E-13	5.06	2.21E-12	5.08	1158	0.18
320	1.88E-14	5.18	8.57E-14	4.69	1866	0.66
FE fast sweeping, $\gamma=1.0$						
N	L <sub>1</sub> error	L <sub>1</sub> order	L <sub>∞</sub> error	L <sub>∞</sub> order	iter#	CPU time
20	3.50E-08	-	5.90E-08	-	140	4.0E-03
40	8.06E-10	5.44	2.29E-09	4.69	155	9.0E-03
80	2.28E-11	5.14	7.46E-11	4.94	212	2.6E-02
160	6.88E-13	5.05	2.36E-12	4.98	318	8.3E-02
320	2.13E-14	5.01	7.69E-14	4.94	548	0.28

Table 1: Example 1. A 1D Burgers’ equation. Accuracy, iteration numbers and CPU times of two different iterative schemes. CPU time unit: second.

$\frac{(\rho v)_i^{n+1} - (\rho v)_i^n}{\Delta t_n}$ ,  $RA_i = \frac{E_i^{n+1} - E_i^n}{\Delta t_n}$ . The convergence criterion is set to be  $ResA \leq \delta$ , where the threshold value  $\delta$  is taken to be at round off error level  $10^{-11} \sim 10^{-13}$ .

### Example 1. A 1D Burgers’ equation

We solve the following one-dimensional steady state Burgers’ equation with a source term

$$\left(\frac{u^2}{2}\right)_x = \sin x \cos x, \quad x \in \left[\frac{1}{4}\pi, \frac{3}{4}\pi\right].$$

The initial condition function  $u_0(x) = 2\sin x$  is used as the initial guess in the iterations, in which case the unique steady state solution for this problem is  $u(x) = \sin x$ . An inflow boundary condition is imposed at the left boundary  $x = (1/4)\pi$  with  $u((1/4)\pi) = \sqrt{2}/2$ . The right boundary  $x = (3/4)\pi$  is the outflow boundary. Both iterative schemes with the fifth order ILW procedure and fifth order multi-resolution WENO extrapolation are used to compute the steady state solution.  $10^{-13}$  is taken as the iteration convergence criterion threshold value. Mesh refinement study is performed.  $L_1$  and  $L_\infty$  numerical errors and accuracy orders, iteration numbers and CPU costs of each iterative scheme to converge are presented in Table 1. The results show that both schemes achieve comparable numerical errors and the fifth order accuracy when they converge. However, the FE fast sweeping scheme is more efficient than the TVD-RK3 scheme in terms of both the iteration numbers and CPU times. In this example, we can see that the FE fast sweeping scheme saves about 60% CPU time cost of that of the RK Jacobi (TVD-RK3) scheme on the most refined grid.

### Example 2. A 2D Euler system

The second accuracy test example we consider is the following two-dimensional steady state Euler system

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

RK Jacobi, $\gamma=1.0$						
$N \times N$	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order	iter‡	CPU time
$20 \times 20$	1.83E-05	-	8.34E-05	-	423	2.81
$30 \times 30$	2.84E-06	4.59	1.25E-05	4.68	525	5.84
$40 \times 40$	7.30E-07	4.72	3.17E-06	4.76	636	11.09
$50 \times 50$	2.51E-07	4.79	1.08E-06	4.83	753	19.13
$60 \times 60$	1.04E-07	4.83	4.46E-07	4.85	870	31.55
$70 \times 70$	4.93E-08	4.85	2.10E-07	4.88	993	46.59
$80 \times 80$	2.57E-08	4.87	1.09E-07	4.89	1122	68.63
FE fast sweeping, $\gamma=1.0$						
$N \times N$	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order	iter‡	CPU time
$20 \times 20$	2.43E-05	-	8.30E-05	-	134	0.91
$30 \times 30$	3.60E-06	4.71	1.09E-05	5.00	163	2.45
$40 \times 40$	9.03E-07	4.81	2.78E-06	4.76	192	4.50
$50 \times 50$	3.05E-07	4.86	9.43E-07	4.85	224	8.16
$60 \times 60$	1.26E-07	4.87	3.86E-07	4.90	256	13.20
$70 \times 70$	5.90E-08	4.90	1.80E-07	4.94	288	20.31
$80 \times 80$	3.06E-08	4.91	9.39E-08	4.89	320	29.84

Table 2: Example 2. A 2D Euler system. Accuracy, iteration numbers and CPU times of two different iterative schemes. CPU time unit: second.

The exact steady state solution is  $\rho(x, y) = 1 + 0.2 \sin(x - y)$ ,  $u(x, y) = 1$ ,  $v(x, y) = 1$ , and  $p(x, y) = 1$ . The computational domain is  $(x, y) \in [-\pi, \pi] \times [-\pi, \pi]$ . To start the iterations in the schemes, the exact steady state solution is used as the initial guess. Since the exact steady state solution does not satisfy the numerical schemes, it will be driven by the iterative schemes to their numerical steady states. Both iterative schemes with the fifth order ILW numerical boundary treatment and fifth order multi-resolution WENO extrapolation are used to compute the steady state solution. We take  $10^{-12}$  as the iteration convergence criterion threshold value and perform mesh refinement study.  $L_1$  and  $L_\infty$  numerical errors and accuracy orders, iteration numbers and CPU costs of each iterative scheme to converge are reported in Table 2. Similar as the last example, we observe that both iterative schemes achieve comparable numerical errors and the fifth order accuracy when they converge. Again, the FE fast sweeping scheme is more efficient than the RK Jacobi (TVD-RK3) scheme in both the iteration numbers and CPU times, and it saves about 60% CPU time cost of that of the TVD-RK3 scheme in this 2D Euler system example.

### Example 3. Regular shock reflection

We consider the two dimensional regular shock reflection problem. Because the residues of high order WENO schemes in solving this problem are often difficult to settle down to machine zero or the round-off error level (see e.g. [24, 26, 27]), this is a typical benchmark problem to test the convergence of high order WENO methods for solving steady flow problems. In recent work [34, 10], it is found that the iterative schemes based on multi-resolution WENO schemes can drive the iteration residues to the round-off error level. Here we use this example to test the absolutely convergent fixed-point fast sweeping method with ILW boundary treatment proposed here, and verify its efficiency. The computational domain is a rectangle with length 4 and height 1. The flow

RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
0.6	5661	4.68	52.03
0.7	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
0.6	1811	4.48	27.34
0.7	Not convergent	-	-

Table 3: Example 3. Regular shock reflection. Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-12}$ . CPU time unit: second.

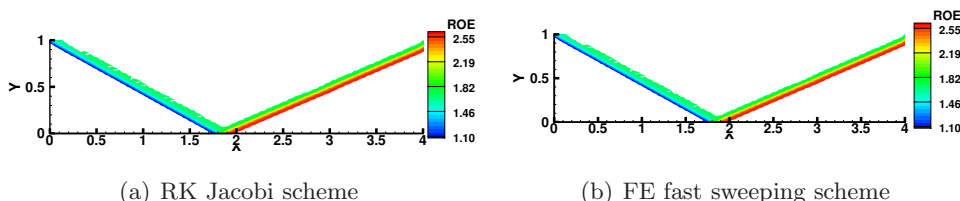


Figure 2: Example 3. Regular shock reflection. 30 equally spaced density contours from 1.1 to 2.6 of the converged steady states of numerical solutions by two different iterative schemes.

has the following Dirichlet conditions along the left and the top boundaries:

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

The initial values in the entire domain to start the iterations are set to be those values of the left boundary. The right boundary is the outflow boundary. In the literature (e.g. [24, 26, 27, 34, 10]), a reflection condition is applied along the bottom boundary. Here, the fifth order ILW boundary treatment is applied to the bottom boundary in order to test the method. Both the fast sweeping method and the RK Jacobi scheme are used to compute the steady state solution, and  $10^{-12}$  is taken as the iteration convergence criterion threshold value. The computational grid is  $120 \times 30$ . Number of iterations, the final time and total CPU time when the schemes converge are reported for the RK Jacobi scheme and the FE fast sweeping scheme with the fifth order ILW boundary treatment in Table 3. It is observed that it takes the FE fast sweeping scheme a much smaller number of iterations to converge to steady state than the RK Jacobi (TVD-RK3) scheme, and about 50% CPU time cost is saved. Figure 2 presents the contour plots of the density variable of the converged steady state solutions, and the comparable results are obtained for both schemes. Figure 3 shows residue history in terms of iterations for these schemes. We see that for this difficult benchmark problem, the iteration residues converge to small values of round off error level for both schemes with the fifth order ILW boundary treatment.

#### Example 4. Supersonic flow past a cylinder

In this example we apply the methods to a problem with the physical boundary not aligned with the computational grids, i.e., supersonic flow past a cylinder problem [21, 11]. A curved solid wall that is a circular cylinder of unit diameter is positioned at the origin  $(0,0)$  of the computational

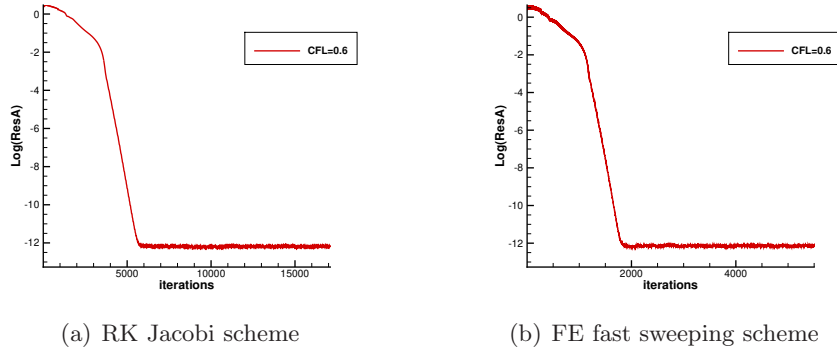


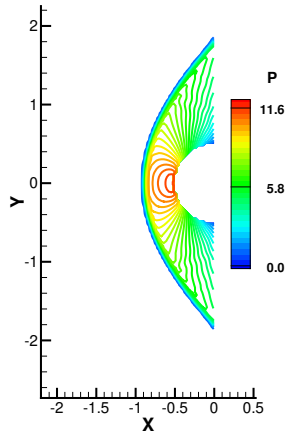
Figure 3: Example 3. Regular shock reflection. The convergence history of the residue as a function of number of iterations for two schemes.

RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
0.4	59154	37.07	3447.44
0.5	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
0.6	11100	31.34	1417.70
0.7	Not convergent	-	-

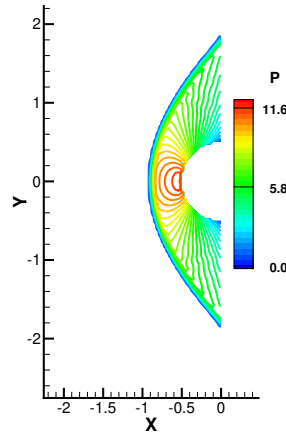
Table 4: Example 4. Supersonic cylinder problem. Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-12}$ . CPU time unit: second.

domain  $[-5.0, 0] \times [-5.0, 5.0]$ . We follow the setup as in [21]. The problem is initialized by a Mach 3 flow moving toward the cylinder wall from the left. At the left inflow boundary  $x = -5.0$ , the uniform far-field data is imposed. At the top boundary  $y = 5.0$  and the right boundary  $x = 0$ , we use constant extrapolation as in [21]. The simulation is carried out at the upper half of the whole domain by the symmetry of the problem. Hence at  $y = 0$ , the reflection technique is applied. The fifth order ILW procedure is used to deal with the surface of the solid wall cylinder. The FE fast sweeping scheme and the RK Jacobi scheme are used to compute the steady state solution, and  $10^{-12}$  is taken as the iteration convergence criterion threshold value. The computational grid sizes are  $\Delta x = \Delta y = \frac{1}{32}$ . Number of iterations, the final time and total CPU time when the iterations converge are reported for both schemes with the fifth order ILW boundary treatment in Table 4. Again, it is observed that the FE fast sweeping scheme is much more efficient than the RK Jacobi scheme to converge to steady state, in terms of both the number of iterations and CPU time costs. About 60% CPU time cost is saved by using the FE fast sweeping scheme with the fifth order ILW boundary treatment. The contour plots of the pressure variable of the converged steady state solutions is shown in Figure 4. We observe similar results for both iterative schemes and the well-captured bow shock. Residue history in terms of iterations for these two schemes is presented in Figure 5. As in the last example, the iteration residues settle down to the  $10^{-12}$  level, the round off error level, for both iterative schemes with the fifth order ILW boundary method.

### Example 5. Supersonic flow past a NACA0012 airfoil

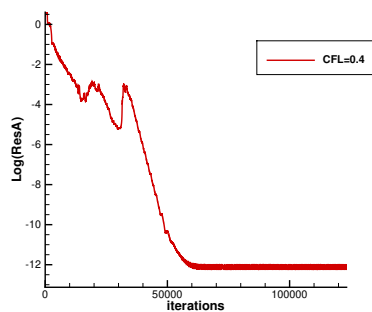


(a) RK Jacobi Scheme

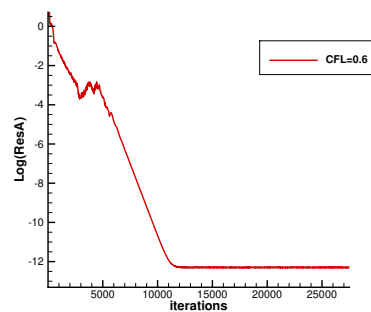


(b) FE fast sweeping Scheme

Figure 4: Example 4. Supersonic cylinder problem. 30 equally spaced pressure contours from 0 to 12 of the converged steady states of numerical solutions by two different iterative schemes.



(a) RK Jacobi Scheme



(b) FE fast sweeping Scheme

Figure 5: Example 4. Supersonic cylinder problem. The convergence history of the residue as a function of number of iterations for two schemes.

RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.0	11742	11.55	1408.97
1.1	10707	11.59	1281.77
1.2	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.1	3561	11.73	721.17
1.2	3201	11.51	659.97
1.3	Not convergent	-	-

Table 5: Example 5. Supersonic flow past a single NACA0012 airfoil.  $Ma_\infty = 2$ . Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-12}$ . CPU time unit: second.

In this example, we consider the problem of an inviscid Euler supersonic flow past a single NACA0012 airfoil configuration [16, 35]. Two cases of supersonic flows are tested, i.e., a flow with Mach number  $Ma_\infty = 2$  and angle of attack  $\alpha = 1^\circ$ ; and a flow with Mach number  $Ma_\infty = 3$  and angle of attack  $\alpha = 3^\circ$ . The computational domain is  $[-1.5, 1.5] \times [-1.5, 1.5]$ , and the computational grid is  $200 \times 200$ . The fifth order ILW procedure is used to deal with the complex surface boundary in this problem. Number of iterations required to reach the convergence threshold value, the final time and total CPU time when convergence is obtained for the RK Jacobi and the FE fast sweeping schemes with the largest possible CFL numbers  $\gamma$  are reported in Table 5 for the  $Ma_\infty = 2$  case and Table 6 for the  $Ma_\infty = 3$  case. Contour plots of the pressure variable of the converged steady state solutions for these schemes are shown in Figure 6 and Figure 8 for two different Mach number cases. It is observed that the shock waves on the complex surface boundary are well captured. Residue history in terms of iterations is reported in Figure 7 and Figure 9, which show that the iteration residues settle down to the round off error  $10^{-11} \sim 10^{-12}$  level for both iterative schemes with the fifth order ILW boundary method. As shown by the results of this example, if the largest CFL number allowed by each scheme to converge to steady states is used, the FE fast sweeping scheme saves about 50%  $\sim$  70% CPU time of that by the RK Jacobi scheme, while similar numerical steady state solutions are obtained by them.

### Example 6. Subsonic flow past a NACA0012 airfoil

Following the last example, here we test the cases of subsonic flow past a single NACA0012 airfoil configuration. Two cases of subsonic flows are considered, i.e., a flow with Mach number  $Ma_\infty = 0.8$  and angle of attack  $\alpha = 1.25^\circ$ ; and a flow with Mach number  $Ma_\infty = 0.5$  and angle of attack  $\alpha = 1.25^\circ$ . The computational domain is  $[-1.0, 2.0] \times [-1.5, 1.5]$ , and the computational grid is  $200 \times 200$ . As in the supersonic cases of the last example, the fifth order ILW procedure is used to deal with the complex surface boundary in this NACA0012 airfoil problem. Number of iterations required to reach the convergence threshold value, the final time and total CPU time when convergence is obtained for the RK Jacobi and the FE fast sweeping schemes with the largest possible CFL numbers  $\gamma$  are reported in Table 7 for the  $Ma_\infty = 0.8$  case and Table 8 for the  $Ma_\infty = 0.5$  case. Contour plots of the pressure variable of the converged steady state solutions for these schemes are shown in Figure 10 and Figure 12 for these different Mach number cases. As in the previous example, we observe that the shock waves on the complex surface boundary are well captured. In Figure 11 and Figure 13, residue history in terms of iterations is presented. Again, the iteration residues settle down to the round off error  $10^{-11} \sim 10^{-12}$  level for both

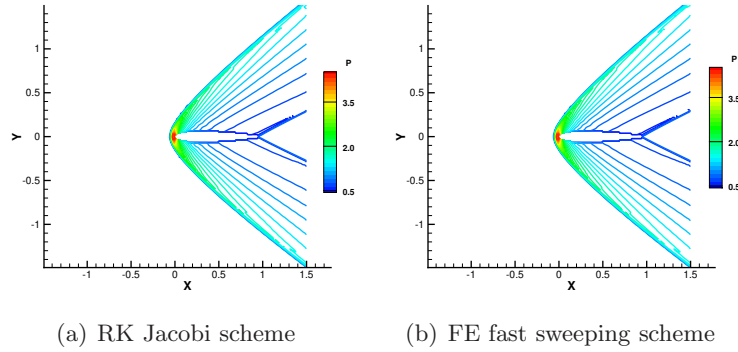


Figure 6: Example 5. Supersonic flow past a single NACA0012 airfoil.  $Ma_\infty = 2$ . 30 equally spaced pressure contours from 0.4 to 4.2 of the converged steady states of numerical solutions by two different iterative schemes.

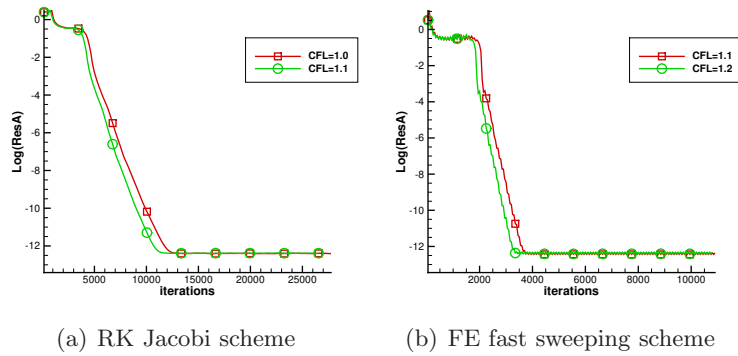


Figure 7: Example 5. Supersonic flow past a single NACA0012 airfoil.  $Ma_\infty = 2$ . The convergence history of the residue as a function of number of iterations for two schemes.

RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.0	6498	4.86	1122.81
1.1	5910	4.86	965.31
1.2	5418	4.86	909.52
1.3	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.0	1573	3.56	324.41
1.1	1413	3.52	285.92
1.2	Not convergent	-	-

Table 6: Example 5. Supersonic flow past a single NACA0012 airfoil.  $Ma_\infty = 3$ . Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-11}$ . CPU time unit: second.

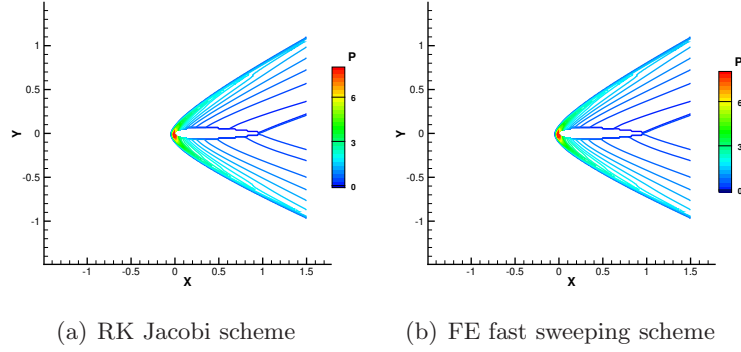


Figure 8: Example 5. Supersonic flow past a single NACA0012 airfoil.  $Ma_\infty = 3$ . 30 equally spaced pressure contours from 0.4 to 8.0 of the converged steady states of numerical solutions by two different iterative schemes.

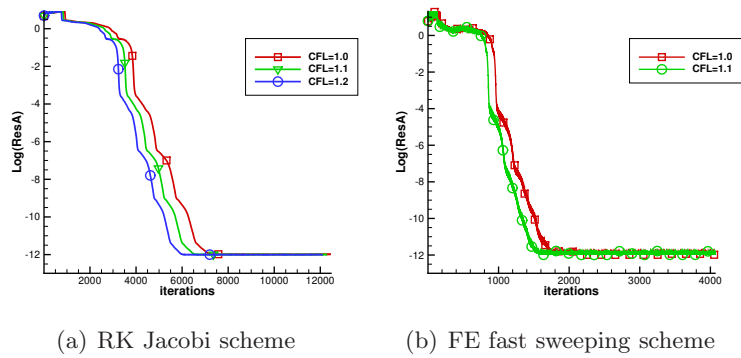


Figure 9: Example 5. Supersonic flow past a single NACA0012 airfoil.  $Ma_\infty = 3$ . The convergence history of the residue as a function of number of iterations for two schemes.



RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.8	34479	83.93	4081.89
1.9	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.0	18798	76.25	3128.41
1.1	Not convergent	-	-

Table 7: Example 6. Subsonic flow past a single NACA0012 airfoil.  $Ma_\infty = 0.8$ , Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-11}$ . CPU time unit: second.

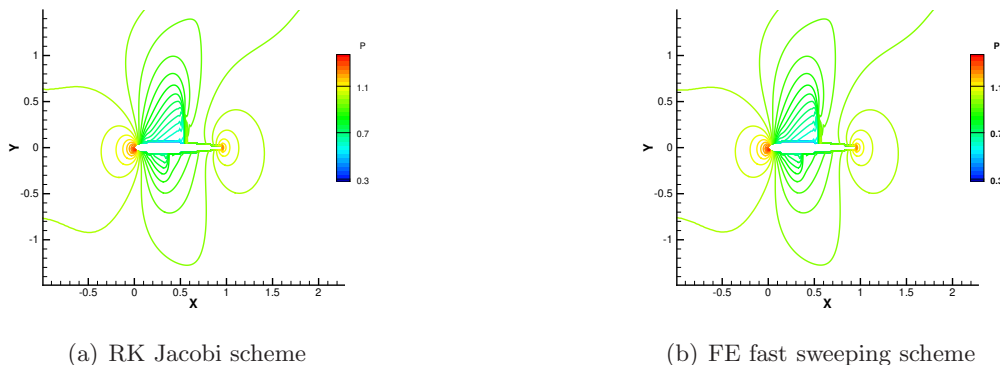


Figure 10: Example 6. Subsonic flow past a single NACA0012 airfoil.  $Ma_\infty = 0.8$ , 30 equally spaced pressure contours from 0.3 to 1.35 of the converged steady states of numerical solutions by two different iterative schemes.

iterative schemes with the fifth order ILW boundary procedure. From the simulation results of this example, if the largest CFL number allowed by each scheme to converge to steady states is used, the FE fast sweeping scheme saves about 46% CPU time of that by the RK Jacobi scheme for the  $Ma_\infty = 0.5$  case to obtain similar numerical steady state solutions, while about 23% CPU time saving is obtained for the  $Ma_\infty = 0.8$  case. Comparing with the supersonic cases of the last example, we observe less CPU time saving for the subsonic cases here. This will be further verified in the next example.

### Example 7. Subsonic flow past a cylinder

In this example, we solve the problem of a subsonic flow past a circular cylinder with the Mach number  $M_\infty = 0.38$  in [35, 12]. The radius of the circular cylinder is 0.5, which is the curved solid wall and located at the center of the computational domain  $[-5.0, 5.0] \times [-5.0, 5.0]$ . The computational grid is  $320 \times 320$ . To deal with the physical boundary which is not aligned with the computational grids, we apply the fifth order ILW procedure to the complex surface boundary. Number of iterations required to reach the convergence threshold value, the final time and total CPU time when convergence is obtained for the RK Jacobi and the FE fast sweeping schemes with the largest possible CFL numbers  $\gamma$  are reported in Table 9. Contour plots of the Mach number variable of the converged steady state solutions for both schemes are shown in Figure 14. We observe that comparable numerical steady state solutions are obtained by these two schemes. However, it

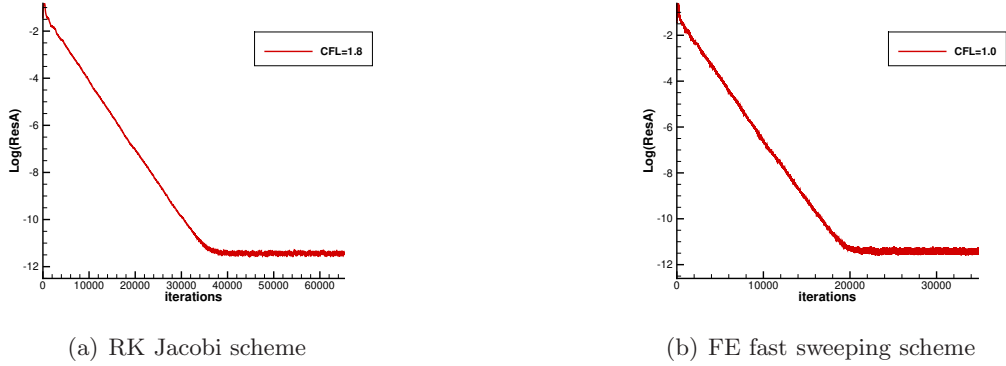


Figure 11: Example 6. Subsonic flow past a single NACA0012 airfoil.  $Ma_\infty = 0.8$ . The convergence history of the residue as a function of number of iterations for two schemes.

RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
2.0	19008	58.63	2680.56
2.1	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.4	7202	46.63	1449.30
1.5	Not convergent	-	-

Table 8: Example 6. Subsonic flow past a single NACA0012 airfoil.  $Ma_\infty = 0.5$ . Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-11}$ . CPU time unit: second.

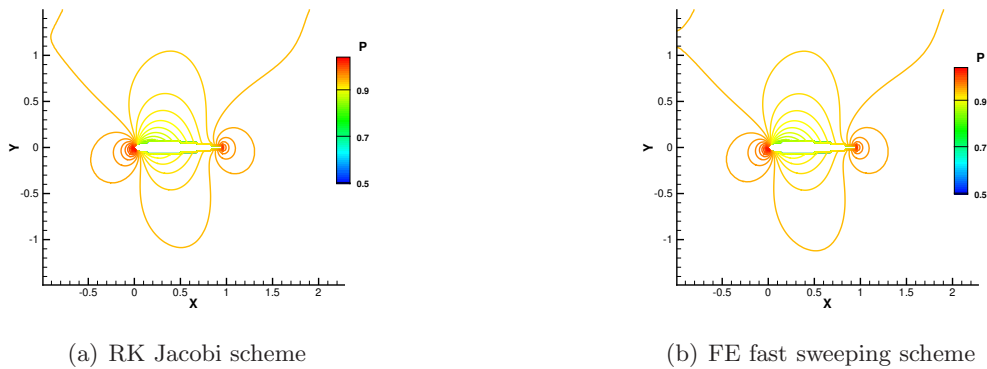


Figure 12: Example 6. Subsonic flow past a single NACA0012 airfoil.  $Ma_\infty = 0.5$ . 60 equally spaced pressure contours from 0.5 to 1.1 of the converged steady states of numerical solutions by two different iterative schemes.

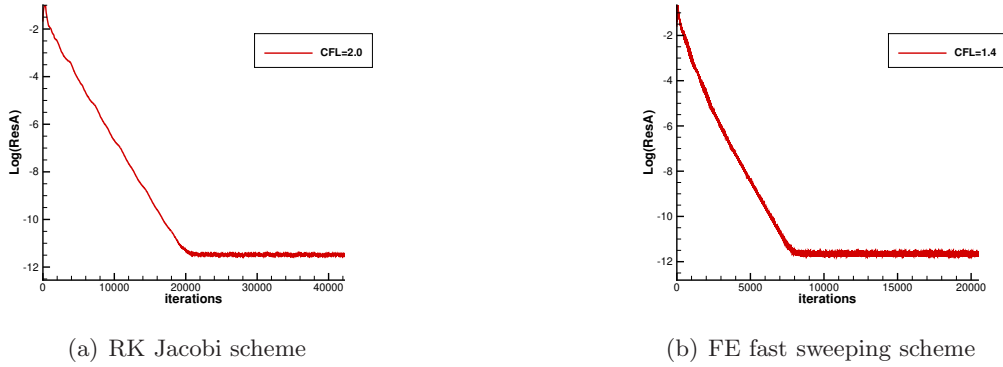
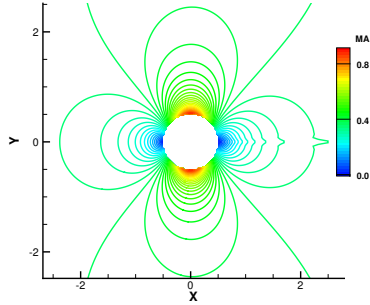


Figure 13: Example 6. Subsonic flow past a single NACA0012 airfoil.  $Ma_\infty = 0.5$ . The convergence history of the residue as a function of number of iterations for two schemes.

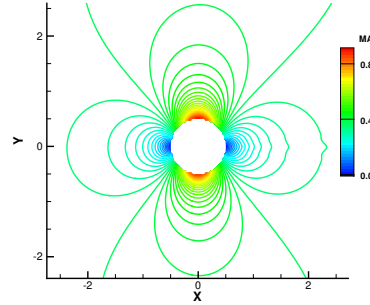
RK Jacobi scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
1.9	33090	190.79	7383.91
2.0	Not convergent	-	-
FE fast sweeping scheme			
$\gamma$ : CFL number	iteration number	final time	CPU time
0.9	18004	146.46	7223.11
1.0	Not convergent	-	-

Table 9: Example 7. Subsonic flow past a circular cylinder. Number of iterations, the final time and total CPU time when convergence is obtained. Convergence criterion threshold value is  $10^{-11}$ . CPU time unit: second.

is noticeable that the FE fast sweeping scheme gives a better symmetric numerical steady state solution around the cylinder than the RK Jacobi scheme. Residue history in terms of iterations provided in Figure 15 shows that the iteration residues settle down to the round off error  $10^{-13} \sim 10^{-14}$  level for both iterative schemes with the fifth order ILW boundary procedure. From the simulation results of this example with a low Mach number flow, we observe less CPU time saving by using the FE fast sweeping scheme than the supersonic cases of the previous examples. The Table 9 shows that the number of iterations required to reach the convergence of the FE fast sweeping scheme is much less than the RK Jacobi scheme. However, since in the fast sweeping iterations, the WENO reconstructions for numerical fluxes at the cell interfaces need to be performed individually for grid points sharing the same cell interface due to the Gauss-Seidel procedure [10], one iteration of the FE fast sweeping scheme has more operations than that of the RK Jacobi scheme. Hence the CPU time cost of the FE fast sweeping scheme in this example is just slightly less than that of the RK Jacobi scheme. It will be interesting to study the possible reason which causes the FE fast sweeping scheme less efficient for these low Mach number flow problems, for example, by investigating the key factors that fast sweeping methods can accelerate convergence of numerical simulations to steady state solutions [31]. This will be among our future work on this topic.

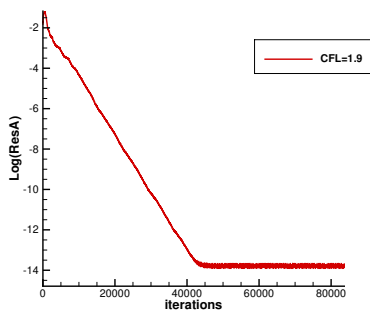


(a) RK Jacobi scheme

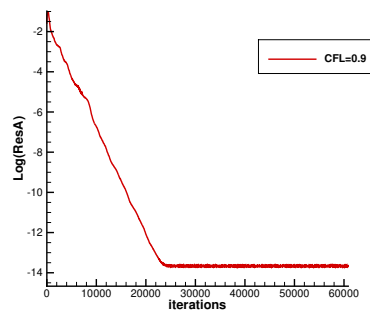


(b) FE fast sweeping scheme

Figure 14: Example 7. Subsonic flow past a circular cylinder. 44 equally spaced Mach number contours from 0.0 to 0.92 of the converged steady states of numerical solutions by two different iterative schemes.



(a) RK Jacobi scheme



(b) FE fast sweeping scheme

Figure 15: Example 7. Subsonic flow past a circular cylinder. The convergence history of the residue as a function of number of iterations for two schemes.

## 4 Concluding remarks

In this paper, the absolutely convergent fixed-point fast sweeping WENO method in [10] is combined with the inverse Lax-Wendroff boundary treatment procedure to solve steady state solution of hyperbolic conservation laws, especially for problems with the physical boundary not aligned with the computational grids. By applying WENO reconstructions with unequal-sized substencils such as the fifth order multi-resolution WENO scheme, iteration residues of the fast sweeping method with the ILW procedure converge to tiny values at the round-off error level, for all test problems including difficult cases. Via numerical experiments on various examples, we also verify that the absolutely convergent fixed-point fast sweeping method with the ILW boundary treatment presented in this paper is more efficient than the popular TVD-RK3 time-marching method to converge to steady state solutions of a high order WENO scheme, and more than 50% CPU time costs are saved for most of examples in this paper. In the future work, we will study the approach about how to further improve the computational efficiency of the absolutely convergent fixed-point fast sweeping WENO method with the ILW boundary treatment procedure, especially for some subsonic flow problems.

**Conflict of Interest:** the authors declare that there is no conflict of interest.

## References

- [1] 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(6) (2008), 3191-3211.
- [2] 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(5) (2011), 1766-1792.
- [3] S. Chen, *Fixed-point fast sweeping WENO methods for steady state solution of scalar hyperbolic conservation laws*, Int. J. Nume. Anal. Mod., 11(1) (2014), 117-130.
- [4] S. Fomel, S. Luo and H. Zhao, *Fast sweeping method for the factored eikonal equation*, J. Comput. Phys., 228(17) (2009), 6440-6455.
- [5] S. Gottlieb, C.-W. Shu and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43(1) (2001), 89-112.
- [6] L. Huang, C.-W. Shu and M. Zhang, *Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation*, J. Comput. Math., 26(3) (2008), 336-346.
- [7] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126(1) (1996), 202-228.
- [8] L. Krivodonova and M. Berger, *High-order accurate implementation of solid wall boundary conditions in curved geometries*, J. Comput. Phys., 211 (2006), 492-512.
- [9] 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(17) (2008), 8191-8208.

- [10] 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.
- [11] J. Lu, C.-W. Shu, S. Tan, and M. Zhang, *An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary*, J. Comput. Phys., 426 (2021), 109940.
- [12] 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.
- [13] J. Qian, Y.-T. Zhang and H.-K. Zhao, *Fast sweeping methods for Eikonal equations on triangular meshes*, SIAM J. Numer. Anal., 45(1) (2007), 83-107.
- [14] J. Qian, Y.-T. Zhang and H.-K. Zhao, *A fast sweeping method for static convex Hamilton-Jacobi equations*, J. Sci. Comput., 31(1) (2007), 237-271.
- [15] J. Shi, Y.-T. Zhang, C.-W. Shu, *Resolution of high order WENO schemes for complicated flow structures*, J. Comput. Phys., 186(2) (2003), 690-696.
- [16] Y. Shida, K. Kuwahara, K. Ono and H. Takami, *Computation of dynamic stall of a NACA-0012 airfoil*, AIAA Journal, 25 (1987), 408-413.
- [17] 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.
- [18] C.-W. Shu, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM Rev., 51(1) (2009), 82-126.
- [19] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, J. Comput. Phys., 77(2) (1988), 439-471.
- [20] B. Sjögren and N.A. Petersson, *A Cartesian embedded boundary method for hyperbolic conservation laws*, Commun. Comput. Phys., 2(6) (2007), 1199-1219.
- [21] S. Tan and C.-W. Shu, *Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws*, J. Comput. Phys., 229(21) (2010), 8144-8166.
- [22] S. Tan, C. Wang, C.-W. Shu. and J. Ning, *Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws*, J. Comput. Phys., 231(6) (2012), 2510-2527.
- [23] L. Wu and Y.-T. Zhang, *A third order fast sweeping method with linear computational complexity for Eikonal equations*, J. Sci. Comput., 62(1) (2015), 198-229.
- [24] 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.
- [25] 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.

- [26] S. Zhang, S. Jiang and C.-W. Shu, *Improvement of convergence to steady state solutions of Euler equations with the WENO schemes*, J. Sci. Comput., 47(2) (2011), 216-238.
- [27] S. Zhang and C.-W. Shu, *A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions*, J. Sci. Comput., 31(1) (2007), 273-305.
- [28] 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(4) (2011), 1873-1896.
- [29] Y.-T. Zhang, H.-K. Zhao and S. Chen, *Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations*, Methods and Applications of Analysis, 13(3) (2006), 299-320.
- [30] 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.
- [31] H.-K. Zhao, *A fast sweeping method for Eikonal equations*, Math. Comput., 74(250) (2005), 603-627.
- [32] J. Zhu and J. Qiu, *A new type of finite volume WENO schemes for hyperbolic conservation laws*, J. Sci. Comput., 73(5) (2017), 1338-1359.
- [33] J. Zhu and C.-W. Shu, *A new type of multi-resolution WENO schemes with increasingly higher order of accuracy*, J. Comput. Phys., 375(3) (2018), 659-683.
- [34] J. Zhu and C.-W. Shu, *Convergence to steady-state solutions of the new type of high-order multi-resolution WENO schemes: a numerical study*, Communications on Applied Mathematics and Computation, 2(6) (2020), 429-460.
- [35] 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.