



A Fixed-Point Fast Sweeping WENO Method with Inverse Lax-Wendroff Boundary Treatment for Steady State of Hyperbolic Conservation Laws

Liang Li¹ · Jun Zhu² · Chi-Wang Shu³ · Yong-Tao Zhang⁴ 

Received: 25 July 2021 / Revised: 14 November 2021 / Accepted: 23 November 2021 /

Published online: 21 March 2022

© The Author(s) 2022

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. 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 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 Runge-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 to converge to steady-state solutions.

Keywords Fixed-point fast sweeping methods · Multi-resolution WENO schemes · Steady state · ILW procedure · Convergence

Mathematics Subject Classification 65N06 · 65M06 · 65N22 · 35L65

Research was supported by the NSFC Grant 11872210.

Research was supported by the NSFC Grant 11872210 and Grant No. MCMS-I-0120G01.

Research supported in part by the AFOSR Grant FA9550-20-1-0055 and NSF Grant DMS-2010107.

✉ Yong-Tao Zhang
yztang10@nd.edu

Extended author information available on the last page of the article

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 [1, 7, 15, 17–19]. 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 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 [5, 19] to time-march numerical solution to steady state. To improve computational efficiency, explicit fast sweeping WENO methods (e.g., [24, 25, 29, 30]) 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., [4, 13, 14, 31]); however, the algorithms are much more complicated than explicit methods when they are coupled with high-order schemes [9, 23, 28]. 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 stencils [32, 33] have very nice convergence property in solving steady-state problems, e.g., the high-order multi-resolution WENO schemes [35], the fixed-point fast sweeping methods were combined with the 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 [11, 21, 22].

In this paper, we combine the absolutely convergent fixed-point fast sweeping WENO method [10] with the ILW 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 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 Sect. 2. In Sect. 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 Sect. 4.

2 Numerical Methods

We consider steady-state problems of hyperbolic conservation laws

$$f(u)_x + g(u)_y = R, \quad (x, y) \in \Omega \tag{1}$$

on a bounded domain Ω 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. In the following, we first describe the absolutely convergent fixed-point fast sweeping WENO method [10] to solve (1).

2.1 A Fifth-Order Absolutely Convergent Fixed-Point Fast Sweeping WENO Method

A spatial discretization of (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 Ω by a uniform Cartesian mesh $\{(x_i, y_j)\}$ with the grid size Δx in the x -direction and Δy in the y -direction. Since high-order WENO schemes with unequal-sized substencils [34, 35] 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 the 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

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

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 ε 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 β_{l_1} , the quantity τ , and reconstruction polynomials $p_{l_1}(x)$ can be found in [10, 33]. 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

$$0 = -(\hat{f}_{i+1/2,j}^- - \hat{f}_{i-1/2,j}^-) / \Delta x - (\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2}) / \Delta y + R(u_{ij}, x_i, y_j), \tag{3}$$

$$i = 1, \dots, N; j = 1, \dots, M,$$

where N and M are the numbers 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 (3) is a nonlinear function of the numerical values at the grid points of the computational stencils. We denote it by L . To solve (3), the forward Euler type fixed-point fast sweeping method [10, 24, 29] is used. It has the following form:

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

$$i = i_1, \dots, i_N; j = j_1, \dots, j_M. \tag{4}$$

u_{ij}^n is the n th step iteration value at the grid point (x_i, y_j) . γ 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 Δt_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:

- (i) $i = 1 : N, j = 1 : M$;
- (ii) $i = N : 1, j = 1 : M$;
- (iii) $i = N : 1, j = M : 1$;
- (iv) $i = 1 : N, j = M : 1$.

This is marked as “ $i = i_1, \dots, i_N; j = j_1, \dots, j_M$ ” in the scheme (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 the 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 (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.

Remark 1 The similar fast sweeping techniques can also be applied to a TVD-RK scheme as that in [24]. However, it was found in [24] that the scheme resulted by applying these fast sweeping techniques to the TVD-RK type scheme is less efficient than the forward Euler type fixed-point fast sweeping method. Hence, in this paper, we only focus on the forward Euler type fixed-point fast sweeping method coupled with the fifth-order multi-resolution WENO scheme.

2.2 The ILW Boundary Treatment

In this section, we outline the ILW 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 1D Case

First we consider 1D compressible steady-state Euler equations

$$F(U)_x = R(U, x, y), \tag{5}$$

where the conservative variables $U \triangleq (U_1, U_2, U_3)^T = (\rho, \rho u, E)^T$, and the flux $F(U) = (\rho u, \rho u^2 + p, u(E + p))^T = \left(U_2, (\gamma' - 1)U_3 + \frac{3-\gamma'}{2} \frac{U_2^2}{U_1}, \left(\gamma' U_3 - \frac{\gamma'-1}{2} \frac{U_2^2}{U_1} \right) \frac{U_2}{U_1} \right)^T$. ρ, 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 γ' is the specific heat ratio and $\gamma' = 1.4$ for the case of air. 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

$$U_m(x_{N+1}) = \sum_{k=0}^4 \frac{(x_{N+1} - 1)^k}{k!} U_m^{*(k)}, \quad m = 1, 2, 3, \tag{6}$$

where $U_m^{*(k)}$ is a $(5 - k)$ th-order approximation of the spatial derivative $\frac{\partial^k U_m}{\partial x^k} \Big|_{x=1}$. 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 $A_{\perp}(U(x_N)) = \frac{\partial F(U)}{\partial U} \Big|_{U=U(x_N)}$, 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 (6). The procedure is summarized as follows. (i) Compute the eigenvalues λ_m and left eigenvectors l_m of the Jacobian matrix $A_{\perp}(U(x_N))$ described above, for $m = 1, 2, 3$. Determine the inflow boundary conditions by the signs of the eigenvalues λ_m . The inflow boundary conditions are prescribed and given in the problem. (ii) Compute the outgoing local characteristic variables $V_m(x_j), j = N - 4, \dots, N$ as described above. Extrapolate $V_m(x_j)$ to the boundary to find $V_m^{*(k)}, k = 0, \dots, 4$ with the multi-resolution WENO extrapolation described below. (iii) Solve for $U_m^{*(0)}, m = 1, 2, 3$, using the prescribed inflow boundary conditions and the extrapolated values $V_m^{*(0)}$. (iv) For $k = 1, \dots, 4$, use the ILW procedure of the PDEs to write the k th-order derivatives of the prescribed inflow boundary conditions as a linear combination of k th-order spatial derivatives and other terms involving lower order spatial derivatives, and together with the extrapolated values, we form a linear system for $U_m^{*(k)}, m = 1, 2, 3$. Solve the linear system to obtain $U_m^{*(k)}, m = 1, 2, 3$. (v) Use the Taylor approximation (6) to find the values of the ghost points.

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 β_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 β_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 τ which depends on the absolute differences between the smoothness indicators is calculated as $\tau = \left(\frac{|\beta_3 - \beta_1| + |\beta_3 - \beta_2|}{2} \right)^2$. The nonlinear weights are

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

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

$$V^{*(k)} = \sum_{r=1}^3 \omega_r \frac{d^k p_r(x)}{dx^k} \Big|_{x=1}. \tag{7}$$

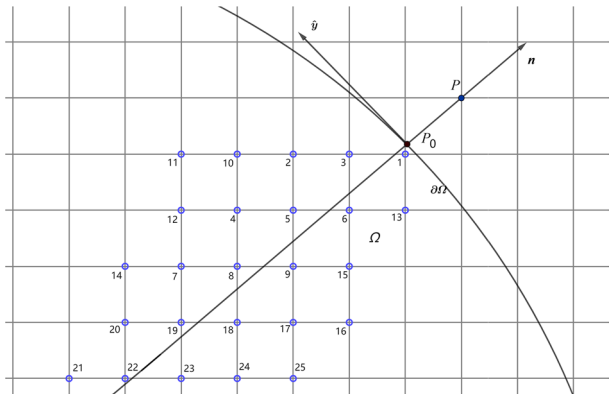
2.2.2 2D Case

We consider 2D compressible steady-state Euler equations

$$F(U)_x + G(U)_y = 0, \quad (x, y) \in \Omega \tag{8}$$

with appropriate boundary conditions, where $U = (U_1, U_2, U_3, U_4)^T = (\rho, \rho u, \rho v, E)^T$, $F(U) = (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T$, $G(U) = (\rho v, \rho uv, \rho v^2 + p, v(E + p))^T$. ρ, 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.

Fig. 1 The stencil S for 2D fifth-order multi-resolution WENO extrapolation



We follow the procedure of the ILW 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 Fig. 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 Γ . 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 θ 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 (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

$$(\hat{U}_m)_P = \sum_{k=0}^4 \frac{\Delta^k}{k!} \hat{U}_m^{*(k)}, \quad m = 1, \dots, 4. \tag{9}$$

Here, Δ 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 $\frac{\partial^k \hat{U}_m}{\partial \hat{x}^k} |_{(x,y)=P_0}$. 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 Ω , to form the Jacobian matrix of the normal flux by $A_{\perp}(\hat{U}_{(0)}) = \frac{\partial F(\hat{U})}{\partial(\hat{U})} |_{\hat{U}=\hat{U}_{(0)}}$. $A_{\perp}(\hat{U}_{(0)})$ 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

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

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 $\frac{\partial^k V_m}{\partial \hat{x}^k} |_{P_0}$. 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 (9).

2D Multi-resolution WENO Extrapolation

Here, we describe the 2D fifth-order multi-resolution WENO extrapolation. Following [11, 22], 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 25 points to guarantee the fifth-order accuracy for the extrapolation function. See Fig. 1 for an illustration of the stencil, where S_1 contains the point 1, and S_2 contains the points 1–9, while S_3 includes the points 1–25. 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 multi-resolution WENO reconstruction idea in [33], we obtain equivalent expressions for these polynomials, that is $p_1(x, y) = q_1(x, y)$, $p_2(x, y) = \frac{1}{\gamma_{2,2}}q_2(x, y) - \frac{\gamma_{1,2}}{\gamma_{2,2}}p_1(x, y)$, 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

$$V_m^{*(k)} = \sum_{r=1}^3 \omega_r \frac{\partial^k p_r(x, y)}{\partial \hat{x}^k} \Big|_{(x,y)=(x_0,y_0)}, \quad k = 0, \dots, 4. \tag{10}$$

Here, the nonlinear weights are computed as follows:

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

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, α is the multi-index notation for partial derivative operator D^α , 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 β_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 ILW boundary treatment for steady-state computations. A popular approach to solve steady-state problems is to apply time-marching approach to the system (3). For example, the TVD-RK3 scheme in [5, 19] 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 ILW 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., [10, 24, 29]). 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

$$\text{Res}A = \sum_{i=1}^N \frac{|R_i|}{N}, \tag{11}$$

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 $\text{Res } A = \sum_{i=1}^{\mathcal{N}} \frac{|R1_i| + |R2_i| + |R3_i| + |R4_i|}{4\mathcal{N}}$. Here, $R * _i$'s are the local residuals of the conservative variables, namely, $R1_i = \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t_n}$, $R2_i = \frac{(\rho u)_i^{n+1} - (\rho u)_i^n}{\Delta t_n}$, $R3_i = \frac{(\rho v)_i^{n+1} - (\rho v)_i^n}{\Delta t_n}$, $R4_i = \frac{E_i^{n+1} - E_i^n}{\Delta t_n}$. The convergence criterion is set to be $\text{Res } A \leq \delta$, where the threshold value δ is taken to be at round-off error level $10^{-11} - 10^{-13}$. In the numerical examples, the CPU times which are costed by the TVD-RK3 scheme and the fast sweeping scheme to reach their numerical steady states are recorded and compared. Also, we record the final time which is the accumulation of time step sizes Δt_n , i.e., the total time that numerical schemes are time-marched from the initial condition at $t = 0$ to the time step that the numerical steady states are reached. In this paper, the number of iterations reported in the numerical simulations counts a complete update of numerical values in all grid points once as one iteration. Hence, these four alternating directions in the FE fast sweeping scheme are counted as four iterations. In the numerical experiments, for various examples, we compare the computational efficiency of the RK Jacobi scheme and the FE fast sweeping scheme using the largest possible CFL numbers γ 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 γ from an initial value.

Example 1 A 1D Burgers' Equation

We solve the following 1D 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_∞ 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 2D steady-state Euler system:

Table 1 Example 1: a 1D Burgers’ equation

RK Jacobi, $\gamma=1.0$						
N	L_1 error	L_1 order	L_∞ error	L_∞ 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	1 158	0.18
320	1.88E-14	5.18	8.57E-14	4.69	1 866	0.66
FE fast sweeping, $\gamma=1.0$						
N	L_1 error	L_1 order	L_∞ error	L_∞ 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

Accuracy, iteration numbers, and CPU times of two different iterative schemes. CPU time unit: second

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

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_∞ 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 2D 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 [10, 35], 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,

Table 2 Example 2: a 2D Euler system

RK Jacobi, $\gamma=1.0$						
$N \times N$	L_1 error	L_1 order	L_∞ error	L_∞ order	Iter#	CPU time
20x20	1.83E-05	–	8.34E-05	–	423	2.81
30x30	2.84E-06	4.59	1.25E-05	4.68	525	5.84
40x40	7.30E-07	4.72	3.17E-06	4.76	636	11.09
50x50	2.51E-07	4.79	1.08E-06	4.83	753	19.13
60x60	1.04E-07	4.83	4.46E-07	4.85	870	31.55
70x70	4.93E-08	4.85	2.10E-07	4.88	993	46.59
80x80	2.57E-08	4.87	1.09E-07	4.89	1 122	68.63
FE fast sweeping, $\gamma=1.0$						
$N \times N$	L_1 error	L_1 order	L_∞ error	L_∞ order	Iter#	CPU time
20x20	2.43E-05	–	8.30E-05	–	134	0.91
30x30	3.60E-06	4.71	1.09E-05	5.00	163	2.45
40x40	9.03E-07	4.81	2.78E-06	4.76	192	4.50
50x50	3.05E-07	4.86	9.43E-07	4.85	224	8.16
60x60	1.26E-07	4.87	3.86E-07	4.90	256	13.20
70x70	5.90E-08	4.90	1.80E-07	4.94	288	20.31
80x80	3.06E-08	4.91	9.39E-08	4.89	320	29.84

Accuracy, iteration numbers, and CPU times of two different iterative schemes. CPU time unit: second

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

$$(\rho, u, v, p)^T = \begin{cases} (1.0, 2.9, 0, 5/7)^T |_{(0,y)}, \\ (1.699\ 97, 2.619\ 34, -0.506\ 32, 1.528\ 19)^T |_{(x,1)}. \end{cases}$$

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., [10, 24, 26, 27, 35]), a reflection condition is applied along the bottom boundary. Here, the fifth-order ILW boundary treatment is applied to the bottom boundary 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 120×30 . Number of iterations, the final time, and the 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.

Table 3 Example 3: regular shock reflection

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
0.6	5 661	4.68	52.03
0.7	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
0.6	1 811	4.48	27.34
0.7	Not convergent	–	–

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

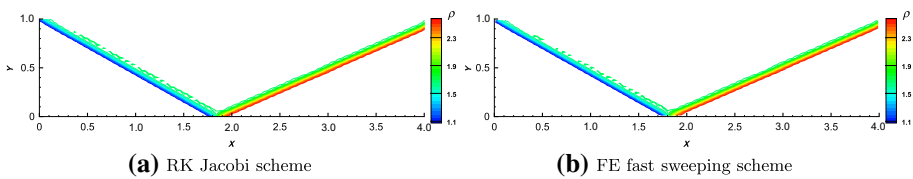


Fig. 2 Example 3: regular shock reflection. Thirty equally spaced density contours from 1.1 to 2.6 of the converged steady states of numerical solutions by two different iterative schemes

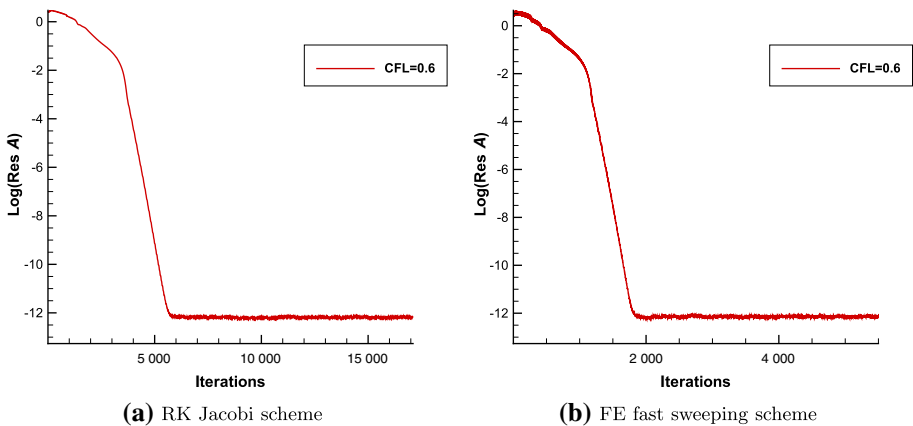


Fig. 3 Example 3: regular shock reflection. The convergence history of the residue as a function of number of iterations for two schemes

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 [11, 21]. A curved solid wall that is a circular cylinder of unit diameter is positioned at the

origin $(0, 0)$ of the computational 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 are 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 the 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 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 are shown in Fig. 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 Fig. 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

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 200×200 . 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 the total CPU time when convergence is obtained for the RK Jacobi and the FE fast sweeping

Table 4 Example 4: supersonic cylinder problem

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
0.4	59 154	37.07	3 447.44
0.5	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
0.6	11 100	31.34	1 417.70
0.7	Not convergent	–	–

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

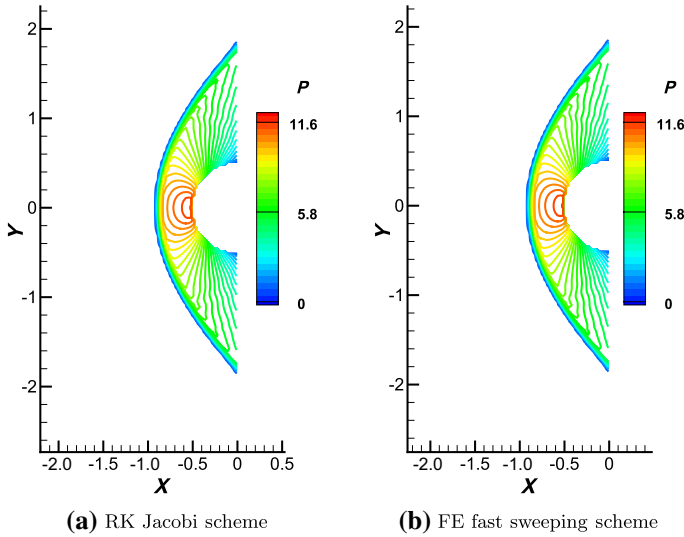


Fig. 4 Example 4: supersonic cylinder problem. Thirty equally spaced pressure contours from 0 to 12 of the converged steady states of numerical solutions by two different iterative schemes

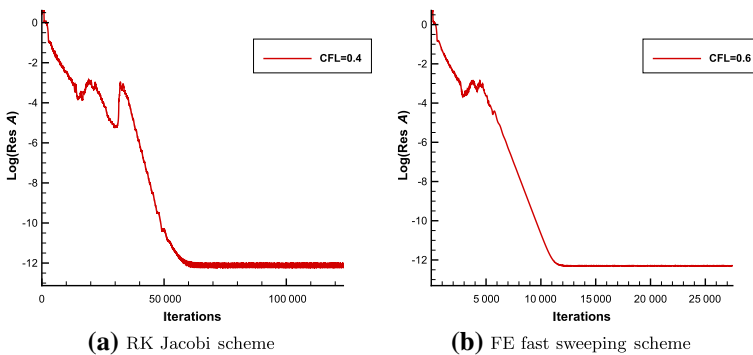


Fig. 5 Example 4: supersonic cylinder problem. The convergence history of the residue as a function of number of iterations for two schemes

schemes with the largest possible CFL numbers γ 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 Figs. 6 and 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 Figs. 7 and 9, which show that the iteration residues settle down to the round-off error $10^{-11} - 10^{-12}$ 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% – 70% CPU time of that by the RK Jacobi scheme, while similar numerical steady-state solutions are obtained by them.

Table 5 Example 5: supersonic flow past a single NACA0012 airfoil

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.0	11 742	11.55	1 408.97
1.1	10 707	11.59	1 281.77
1.2	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.1	3 561	11.73	721.17
1.2	3 201	11.51	659.97
1.3	Not convergent	–	–

$Ma_\infty = 2$. Number of iterations, the final time, and the total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-12} . CPU time unit: second

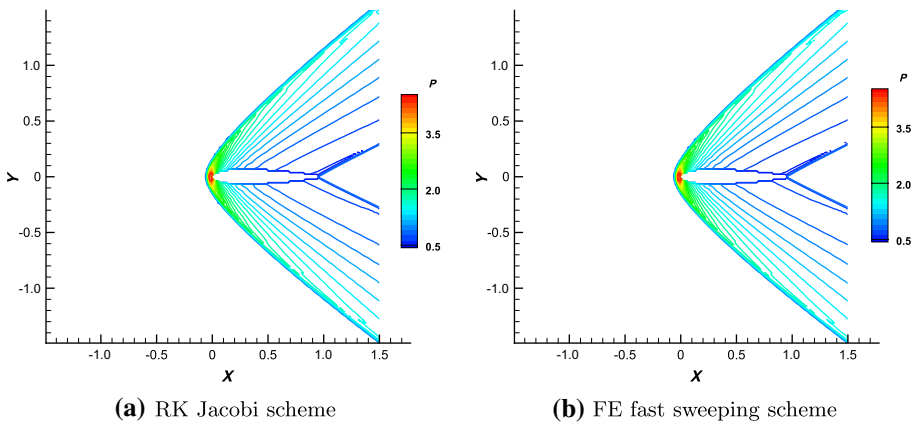


Fig. 6 Example 5: supersonic flow past a single NACA0012 airfoil. $Ma_\infty = 2$. Thirty equally spaced pressure contours from 0.4 to 4.2 of the converged steady states of numerical solutions by two different iterative schemes

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 200×200 . 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 the total CPU time when convergence is obtained for the RK Jacobi and the FE fast sweeping schemes with the largest possible CFL numbers γ 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 Figs. 10 and 12

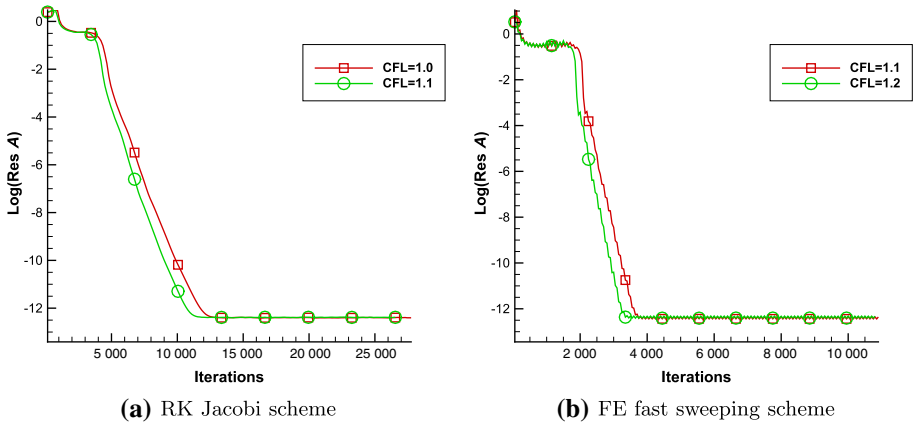


Fig. 7 Example 5: supersonic flow past a single NACA0012 airfoil. $Ma_\infty = 2$. The convergence history of the residue as a function of number of iterations for two schemes

Table 6 Example 5: supersonic flow past a single NACA0012 airfoil

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.0	6 498	4.86	1 122.81
1.1	5 910	4.86	965.31
1.2	5 418	4.86	909.52
1.3	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.0	1 573	3.56	324.41
1.1	1 413	3.52	285.92
1.2	Not convergent	–	–

$Ma_\infty = 3$. Number of iterations, the final time, and the total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11} . CPU time unit: second

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 Figs. 11 and 13, residue history in terms of iterations is presented. Again, the iteration residues settle down to the round-off error $10^{-11} - 10^{-12}$ level for both 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.

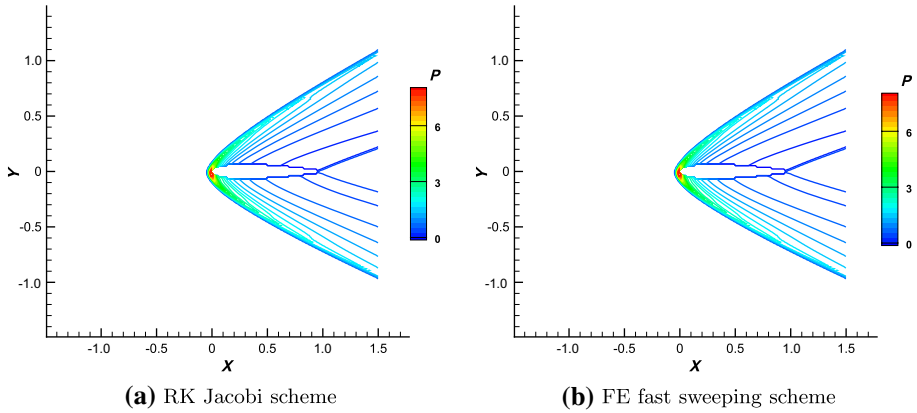


Fig. 8 Example 5: supersonic flow past a single NACA0012 airfoil, $Ma_\infty = 3$. Thirty equally spaced pressure contours from 0.4 to 8.0 of the converged steady states of numerical solutions by two different iterative schemes

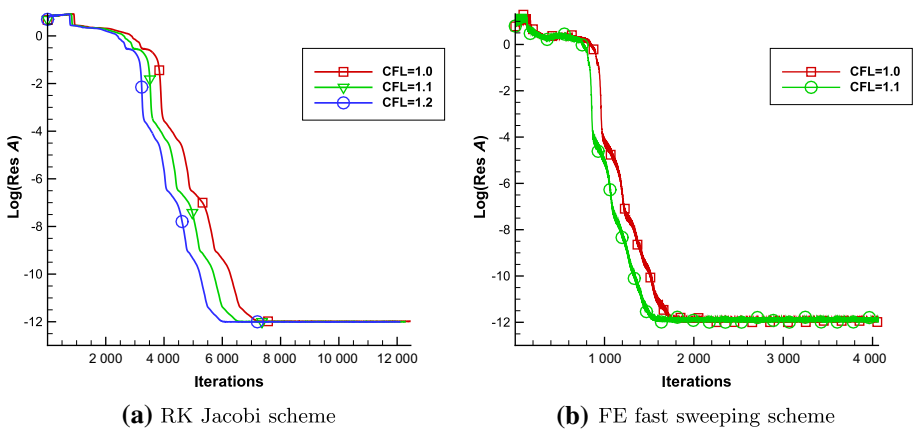


Fig. 9 Example 5: supersonic flow past a single NACA0012 airfoil, $Ma_\infty = 3$. The convergence history of the residue as a function of number of iterations for two schemes

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 $Ma_\infty = 0.38$ in [12, 34]. 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 320×320 . 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 the total CPU time when convergence is obtained for the RK Jacobi and the FE fast sweeping schemes with the largest possible

Table 7 Example 6: subsonic flow past a single NACA0012 airfoil

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.8	34 479	83.93	4 081.89
1.9	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.0	18 798	76.25	3 128.41
1.1	Not convergent	–	–

$Ma_\infty = 0.8$, Number of iterations, the final time, and the total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11} . CPU time unit: second

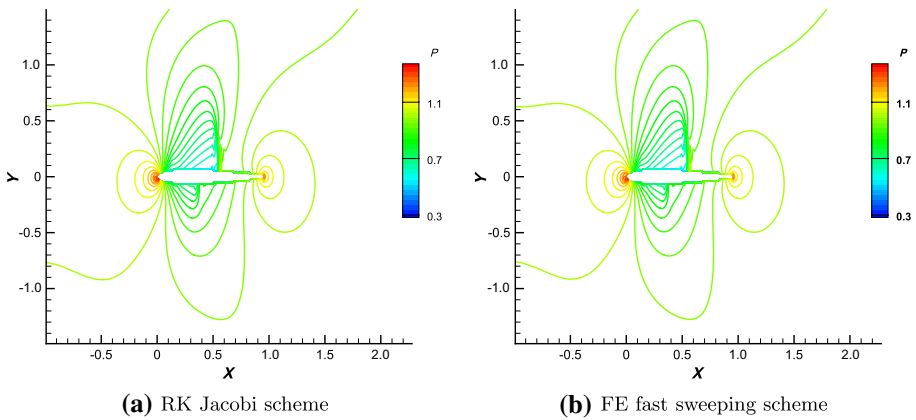


Fig. 10 Example 6: subsonic flow past a single NACA0012 airfoil. $Ma_\infty = 0.8$. Thirty equally spaced pressure contours from 0.3 to 1.35 of the converged steady states of numerical solutions by two different iterative schemes

CFL numbers γ are reported in Table 9. Contour plots of the Mach number variable of the converged steady-state solutions for both schemes are shown in Fig. 14. We observe that comparable numerical steady-state solutions are obtained by these two schemes. However, it 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 Fig. 15 shows that the iteration residues settle down to the round-off error $10^{-13} - 10^{-14}$ 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 using the FE fast sweeping scheme than the supersonic cases of the previous examples. Table 9 shows that the number of iterations

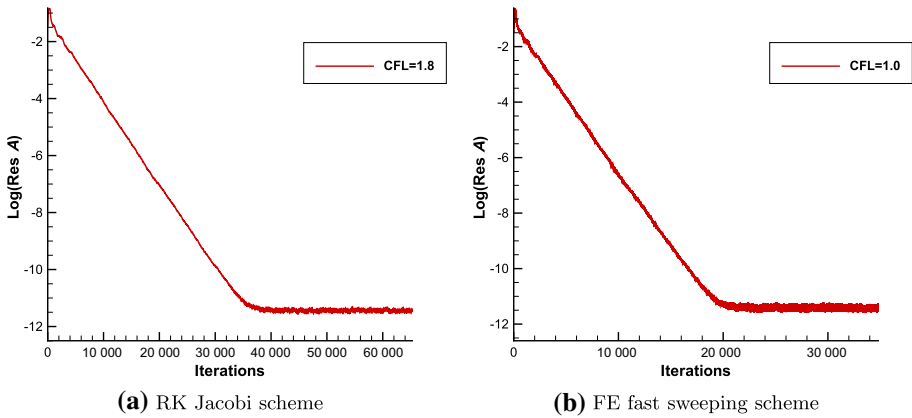


Fig. 11 Example 6: subsonic flow past a single NACA0012 airfoil. $Ma_\infty = 0.8$. The convergence history of the residue as a function of number of iterations for two schemes

Table 8 Example 6: subsonic flow past a single NACA0012 airfoil

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
2.0	19 008	58.63	2 680.56
2.1	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.4	7 202	46.63	1 449.30
1.5	Not convergent	–	–

$Ma_\infty = 0.5$. Number of iterations, the final time, and the total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11} . CPU time unit: second

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.

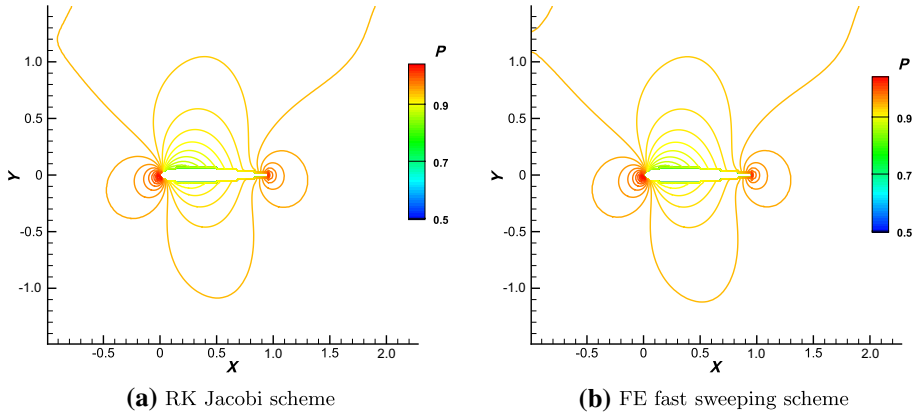


Fig. 12 Example 6: subsonic flow past a single NACA0012 airfoil. $Ma_\infty = 0.5$. Sixty equally spaced pressure contours from 0.5 to 1.1 of the converged steady states of numerical solutions by two different iterative schemes

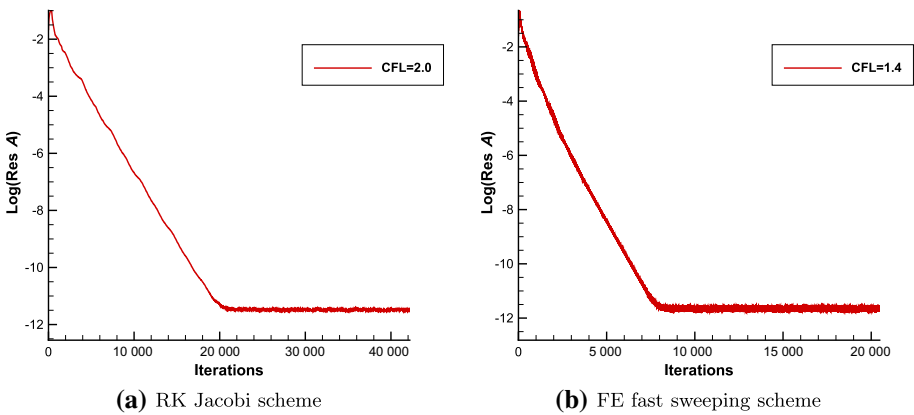


Fig. 13 Example 6: subsonic flow past a single NACA0012 airfoil. $Ma_\infty = 0.5$. The convergence history of the residue as a function of number of iterations for two schemes

4 Concluding Remarks

In this paper, the absolutely convergent fixed-point fast sweeping WENO method in [10] is combined with the ILW boundary treatment procedure to solve the 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

Table 9 Example 7: subsonic flow past a circular cylinder

RK Jacobi scheme			
γ : CFL number	Iteration number	Final time	CPU time
1.9	33 090	190.79	7 383.91
2.0	Not convergent	–	–
FE fast sweeping scheme			
γ : CFL number	Iteration number	Final time	CPU time
0.9	18 004	146.46	7 223.11
1.0	Not convergent	–	–

Number of iterations, the final time, and the total CPU time when convergence is obtained. Convergence criterion threshold value is 10^{-11} . CPU time unit: second

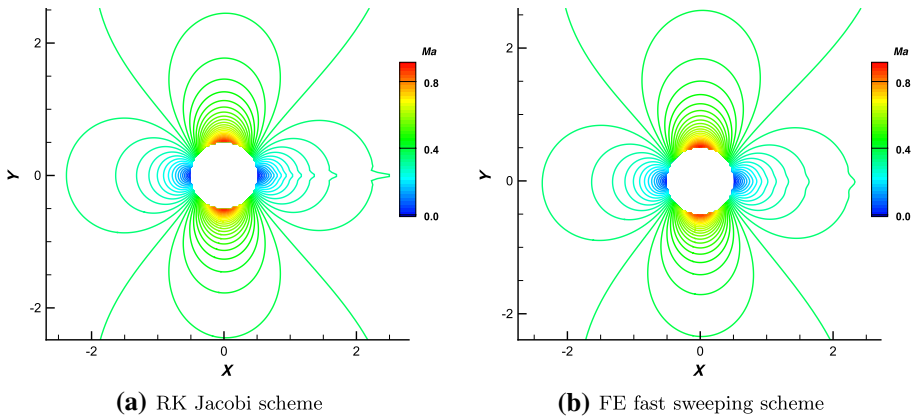


Fig. 14 Example 7: subsonic flow past a circular cylinder. Forty-four equally spaced Mach number contours from 0.0 to 0.92 of the converged steady states of numerical solutions by two different iterative schemes

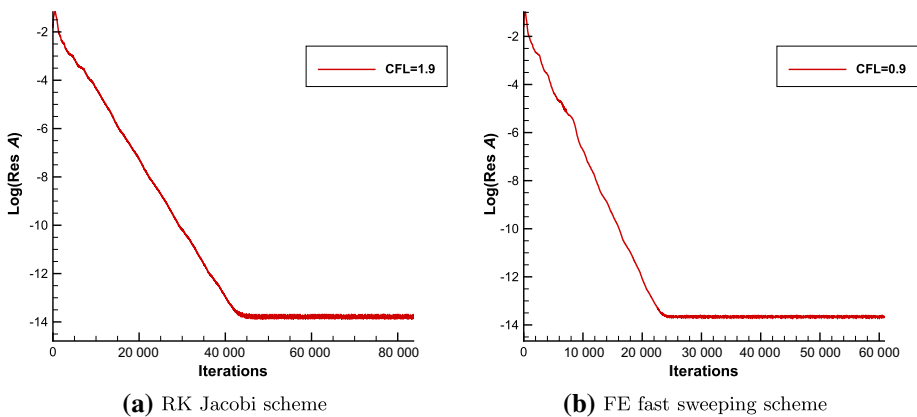


Fig. 15 Example 7: subsonic flow past a circular cylinder. The convergence history of the residue as a function of number of iterations for two schemes

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.

Compliance with Ethical Standards

Conflict of Interest The authors declare that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Borges, R., Carmona, M., Costa, B., Don, W.S.: An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J. Comput. Phys.* **227**(6), 3191–3211 (2008)
2. Castro, M., Costa, B., Don, W.S.: High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *J. Comput. Phys.* **230**(5), 1766–1792 (2011)
3. Chen, S.: Fixed-point fast sweeping WENO methods for steady state solution of scalar hyperbolic conservation laws. *Int. J. Numer. Anal. Mod.* **11**(1), 117–130 (2014)
4. Fomel, S., Luo, S., Zhao, H.: Fast sweeping method for the factored Eikonal equation. *J. Comput. Phys.* **228**(17), 6440–6455 (2009)
5. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **43**(1), 89–112 (2001)
6. Huang, L., Shu, C.-W., Zhang, M.: Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation. *J. Comput. Math.* **26**(3), 336–346 (2008)
7. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**(1), 202–228 (1996)
8. Krivodonova, L., Berger, M.: High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys.* **211**(2), 492–512 (2006)
9. Li, F., Shu, C.-W., Zhang, Y.-T., Zhao, H.-K.: A second order discontinuous Galerkin fast sweeping method for Eikonal equations. *J. Comput. Phys.* **227**(17), 8191–8208 (2008)
10. Li, L., Zhu, J., Zhang, Y.-T.: Absolutely convergent fixed-point fast sweeping WENO methods for steady state of hyperbolic conservation laws. *J. Comput. Phys.* **443**, 110516 (2021)
11. Lu, J., Shu, C.-W., Tan, S., Zhang, M.: An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary. *J. Comput. Phys.* **426**, 109940 (2021)
12. Luo, H., Baum, J.D., Löhner, R.: A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. *J. Comput. Phys.* **225**(1), 686–713 (2007)
13. Qian, J., Zhang, Y.-T., Zhao, H.-K.: Fast sweeping methods for Eikonal equations on triangular meshes. *SIAM J. Numer. Anal.* **45**(1), 83–107 (2007)
14. Qian, J., Zhang, Y.-T., Zhao, H.-K.: A fast sweeping method for static convex Hamilton-Jacobi equations. *J. Sci. Comput.* **31**(1), 237–271 (2007)
15. Shi, J., Zhang, Y.-T., Shu, C.-W.: Resolution of high order WENO schemes for complicated flow structures. *J. Comput. Phys.* **186**(2), 690–696 (2003)
16. Shida, Y., Kuwahara, K., Ono, K., Takami, H.: Computation of dynamic stall of a NACA-0012 airfoil. *AIAA J.* **25**(3), 408–413 (1987)

17. Shu, C.-W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Cockburn, B., Johnson, C., Shu, C.-W., Tadmor, E. (eds) *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*. Lecture Notes in Mathematics, vol. 1697, pp. 325–432. Springer, Berlin (1998)
18. Shu, C.-W.: High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Rev.* **51**(1), 82–126 (2009)
19. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.* **77**(2), 439–471 (1988)
20. Sjögreen, B., Petersson, N.A.: A Cartesian embedded boundary method for hyperbolic conservation laws. *Commun. Comput. Phys.* **2**(6), 1199–1219 (2007)
21. Tan, S., Shu, C.-W.: Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. *J. Comput. Phys.* **229**(21), 8144–8166 (2010)
22. Tan, S., Wang, C., Shu, C.-W., Ning, J.: Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws. *J. Comput. Phys.* **231**(6), 2510–2527 (2012)
23. Wu, L., Zhang, Y.-T.: A third order fast sweeping method with linear computational complexity for Eikonal equations. *J. Sci. Comput.* **62**(1), 198–229 (2015)
24. Wu, L., Zhang, Y.-T., Zhang, S., Shu, C.-W.: High order fixed-point sweeping WENO methods for steady state of hyperbolic conservation laws and its convergence study. *Commun. Comput. Phys.* **20**(4), 835–869 (2016)
25. Xiong, T., Zhang, M., Zhang, Y.-T., Shu, C.-W.: Fast sweeping fifth order WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment. *J. Sci. Comput.* **45**(1), 514–536 (2010)
26. Zhang, S., Jiang, S., Shu, C.-W.: Improvement of convergence to steady state solutions of Euler equations with the WENO schemes. *J. Sci. Comput.* **47**(2), 216–238 (2011)
27. Zhang, S., Shu, C.-W.: A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions. *J. Sci. Comput.* **31**(1), 273–305 (2007)
28. Zhang, Y.-T., Chen, S., Li, F., Zhao, H., Shu, C.-W.: Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations. *SIAM J. Sci. Comput.* **33**(4), 1873–1896 (2011)
29. Zhang, Y.-T., Zhao, H.-K., Chen, S.: Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations. *Meth. Appl. Anal.* **13**(3), 299–320 (2006)
30. Zhang, Y.-T., Zhao, H.-K., Qian, J.: High order fast sweeping methods for static Hamilton-Jacobi equations. *J. Sci. Comput.* **29**(1), 25–56 (2006)
31. Zhao, H.-K.: A fast sweeping method for Eikonal equations. *Math. Comput.* **74**(250), 603–627 (2005)
32. Zhu, J., Qiu, J.: A new type of finite volume WENO schemes for hyperbolic conservation laws. *J. Sci. Comput.* **73**(5), 1338–1359 (2017)
33. Zhu, J., Shu, C.-W.: A new type of multi-resolution WENO schemes with increasingly higher order of accuracy. *J. Comput. Phys.* **375**(3), 659–683 (2018)
34. Zhu, J., Shu, C.-W.: Numerical study on the convergence to steady-state solutions of a new class of finite volume WENO schemes: triangular meshes. *Shock Wav.* **29**(1), 3–25 (2019)
35. Zhu, J., Shu, C.-W.: Convergence to steady-state solutions of the new type of high-order multi-resolution WENO schemes: a numerical study. *Commun. Appl. Math. Comput.* **2**(6), 429–460 (2020)

Authors and Affiliations

Liang Li¹ · Jun Zhu² · Chi-Wang Shu³ · Yong-Tao Zhang⁴ 

Liang Li
liliangnuaa@163.com

Jun Zhu
zhujun@nuaa.edu.cn

Chi-Wang Shu
chi-wang_shu@brown.edu

¹ Key Laboratory of Mathematical Modelling and High Performance Computing of Air Vehicles (NUAA), MIIT, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China

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 210016, Jiangsu, China

³ Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

⁴ Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA