Journal of Computational Physics 234 (2013) 452–471

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Lax–Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws

Weitao Chen [a], Ching-Shan Chou [a,*,1], Chiu-Yen Kao [a,b,2]

[a] Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA
[b] Department of Mathematics and Computer Science, Claremont Mckenna College, Claremont, CA 91711, USA

ARTICLE INFO

Article history:
Received 5 March 2012
Received in revised form 28 September 2012
Accepted 1 October 2012
Available online 23 October 2012

Keywords:
Hyperbolic conservation laws
Steady state problems
Fast sweeping methods
High order accuracy
WENO reconstruction

ABSTRACT

Fast sweeping methods are efficient iterative numerical schemes originally designed for solving stationary Hamilton–Jacobi equations. Their efficiency relies on Gauss–Seidel type nonlinear iterations, and a finite number of sweeping directions. In this paper, we generalize the fast sweeping methods to hyperbolic conservation laws with source terms. The algorithm is obtained through finite difference discretization, with the numerical fluxes evaluated in WENO (Weighted Essentially Non-oscillatory) fashion, coupled with Gauss–Seidel iterations. In particular, we consider mainly the Lax-Friedrichs numerical fluxes. Extensive numerical examples in both scalar and system test problems in one and two dimensions demonstrate the efficiency, high order accuracy and the capability of resolving shocks of the proposed methods.

1. Introduction

Hyperbolic conservation laws and Hamilton–Jacobi equations are first order nonlinear partial differential equations which arise in many applications such as gas dynamics [19,40], shallow water waves [41], geometrical optics [42], crystal growth [23,25], etching, photolithography [33], computer vision [32,24] and seismology [7]. The solutions of these equations can develop singularity, such as discontinuity in the solutions or their derivatives. Under these circumstances, the solutions do not satisfy the equations in the classical sense and one must resort to weak solutions. For hyperbolic conservation equations, "vanishing viscosity solution" and "entropy solution" are introduced to define the weak solution uniquely (see [20,18] and references therein). In early 1980s, Crandall and Lions [10,11] introduced "viscosity solution", among weak solutions, to study of the existence, uniqueness, and stability properties of Hamilton–Jacobi equations. Since then, many numerical methods have been proposed to approximate the viscosity solutions [34,18,9,38]. The challenges of the numerical schemes generally are to achieve high order accuracy and be able to resolve the shock or singularity well, while maintaining conservation. One key to the scheme design of hyperbolic conservation laws and Hamilton–Jacobi equations, in order to correctly capture the viscosity solution, is the use of consistent and conservative numerical fluxes and Hamiltonians.

For Hamilton–Jacobi equations,

$$\phi_t(x,t) + H(x,t,\phi,\nabla\phi(x,t)) = 0$$

* Corresponding author. Tel.: +1 614 292 9947.
E-mail addresses: chen@math.ohio-state.edu (W. Chen), chou@math.ohio-state.edu (C.-S. Chou), kao@math.ohio-state.edu, Ckao@cmc.edu (C.-Y. Kao).

[1] This author is supported by NSF DMS1020625.
[2] This author is partially supported by NSF DMS1216742.

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.10.008

after Osher proved in [22] the link between static and time dependent Hamilton–Jacobi equations, in which the zero level set of the viscosity solution of the time dependent problem at a later time t is the t-level set of the static problem, fast sweeping methods for static Hamilton–Jacobi equations became popular due to their high efficiency. A fast sweeping method mainly consists of the following three essential ingredients: (1) an efficient local solver on a given Cartesian mesh [39,46,15,17,45,26] or triangulation [26,16] based on monotone numerical Hamiltonians, (2) systematic orderings of solution nodes according to some pre-determined information-flowing directions, and (3) Gauss–Seidel type iterations based on a given order of solution nodes. Among fast sweeping schemes, the methods based on upwind Hamiltonians are most efficient for convex Hamiltonians [39,46], while the methods based on Lax–Friedrichs fluxes [15,45] are most flexible to deal with general non-convex Hamiltonians.

Although for hyperbolic conservation laws with source terms,

$$u_t + \nabla \cdot f(u) = s(u),$$

in which the Jacobian matrix $f'(u)$ is diagonalizable with all the eigenvalues being real for any u, there is no such link between time dependent problems and static problems as in Hamilton–Jacobi equations, it is still favorable to have efficient numerical methods for steady state hyperbolic problems. A class of schemes, called "residual distribution schemes" [1–3,12,29,37], were proposed for solving steady state problems with pseudo time stepping. The spirit of these methods is to distribute the residuals, defined through integrating the flux and source terms on triangular or quadrilinear cells in a conservative fashion, and march in pseudo time. Residual distribution schemes were later generalized to high order schemes by Abgrall and Roe [4] through a finite element based approach. Based on the same distribution principles, Chou and Shu [8] developed a finite difference based residual distribution scheme which works on curvilinear meshes, and their scheme achieves high order accuracy and low computational cost as in finite difference methods, but without the constraint of uniform meshes.

These residual distribution schemes, though more efficient by using only pseudo time stepping, are still greatly constrained by the CFL condition for stability, which can substantially limit the speed of the schemes. In this paper, we develop a Gauss–Seidel type iterative method to accelerate the speed to compute the steady state solutions of hyperbolic equations. Inspired by fast sweeping methods for time independent Hamilton–Jacobi equations, we propose methods which discretize the steady state hyperbolic conservation laws directly, by approximating the spatial derivatives with consistent and conservative numerical fluxes, and iterate with Gauss–Seidel type nonlinear method with a finite number of alternating sweeping directions. In particular, we use the Lax–Friedrichs fluxes evaluated in WENO (Weighted Essentially Non-oscillatory) fashion [35,36,21,14,34], to achieve high order accuracy as well as high resolution of shocks.

It is worth pointing out here that, while most steady state hyperbolic problems have unique steady states, with initial conditions reasonably perturbed from the solutions, there are some problems whose steady states are totally dependent on the initial conditions through mass conservation [30]. In those cases, Gauss–Seidel type sweeping may not conserve the mass through the iterations, and therefore an additional constraint needs to be imposed in order to select the particular steady state.

This paper is organized as follows. In Section 2, we present the high order Lax–Friedrichs sweeping methods for one-dimensional scalar and system problems. In Sections 3, the method is extended to two-dimensional equations. Section 4 describes an efficient accuracy-preserving stopping criterion for the fast sweeping iterative scheme. Section 5 contains extensive numerical simulations for one and two-dimensional scalar and system steady state problems to demonstrate high order accuracy, efficiency and robustness of our scheme. Conclusions are given in Section 6.

2. The Lax–Friedrichs sweeping schemes in one dimension

2.1. One-dimensional scalar problems

In this section, we consider the one-dimensional scalar steady state problem

$$f(u)_x = s(u,x), \quad x \in [a,b] \tag{1}$$

subject to an initial guess and appropriate boundary conditions.

To obtain a numerical scheme, the interval is first discretized uniformly into N cells, and the grid points are denoted by $\{x_j\}_{j=0}^N$, where $x_j = a + j\Delta x, j = 0,\ldots,N$ and $\Delta x = (b-a)/N$. The midpoint of a cell is defined as $x_{j+\frac{1}{2}} = (x_j + x_{j+1})/2$, $j = 0,\ldots,N-1$. The numerical approximation of u on the grid points x_j are denoted by $u_j, j = 0,\ldots,N$. A conservative finite difference type discretization of Eq. (1) can be written as

$$\frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x} = s(u_j, x_j), \tag{2}$$

in which $\hat{f}_{j\pm\frac{1}{2}}$ represent numerical fluxes approximating the fluxes at $x_{j\pm\frac{1}{2}}$. The order of the numerical scheme thus depends on the order that $\frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x}$ approximates $f_x(u_j)$.

Our goal is to design a Gauss–Seidel type iterative scheme, based on Eq. (2), which can accelerate the speed to compute steady state solutions of hyperbolic equations. In Section 2.1.1, we introduce the design of the numerical scheme

by presenting a first order sweeping method with Lax–Friedrichs numerical fluxes. This first order method can be generalized to higher order by employing high order Lax–Friedrichs WENO numerical fluxes. The construction of the numerical fluxes with WENO procedure will be reviewed in Section 2.1.2, and the high order sweeping method is introduced in Section 2.1.3. While the general analysis for the convergence of the sweeping methods is difficult, we carry out the analysis for the first order sweeping method for a simple scalar linear equation in Section 2.1.4.

2.1.1. A framework: a first order Lax–Friedrichs sweeping method

We substitute the numerical fluxes in Eq. (2) by the first order Lax–Friedrichs fluxes

$$\hat{f}_{j+\frac{1}{2}} = \frac{1}{2}(f(u_j) + f(u_{j+1})) - \frac{\sigma}{2}(u_{j+1} - u_j), \tag{3}$$

where $\sigma = \max_j\{|f'(u_j)|\},\ j = 1,\ldots,N$. This leads to the discretization formula

$$\frac{1}{2}(f(u_{j+1}) - f(u_{j-1})) - \frac{\sigma}{2}(u_{j+1} - 2u_j + u_{j-1}) - \Delta x s(u_j, x_j) = 0,$$

and equivalently,

$$u_j = \frac{1}{2}(u_{j-1} + u_{j+1}) + \frac{1}{\sigma}(\Delta x s(u_j, x_j) - \frac{1}{2}(f(u_{j+1}) - f(u_{j-1}))).$$

If we denote the iteration step by n, with $n = 0$ corresponding to the initial guess, a simple updating formula would be

$$u_j^{n+1} = \frac{1}{2}(u_{j-1} + u_{j+1}) + \frac{1}{\sigma}(\Delta x s(u_j^n, x_j) - \frac{1}{2}(f(u_{j+1}) - f(u_{j-1}))). \tag{4}$$

The superscripts of u_{j+1} and u_{j-1} on the right-hand-side of Eq. (4) are not specified because they depend on whether the numerical solutions are updated from left to right or from right to left, which we call the sweeping direction. If the sweeping direction is from left to right, then we take $u_{j-1} = u_{j-1}^{n+1}$ and $u_{j+1} = u_{j+1}^n$. The formula becomes

$$u_j^{n+1} = \frac{1}{2}(u_{j-1}^{n+1} + u_{j+1}^n) + \frac{1}{\sigma}\left(\Delta x s(u_j^n, x_j) - \frac{1}{2}\left(f(u_{j+1}^n) - f(u_{j-1}^{n+1})\right)\right).$$

If the sweeping direction is from right to left, then $u_{j-1} = u_{j-1}^n$ and $u_{j+1} = u_{j+1}^{n+1}$ will be used, and

$$u_j^{n+1} = \frac{1}{2}(u_{j-1}^n + u_{j+1}^{n+1}) + \frac{1}{\sigma}\left(\Delta x s(u_j^n, x_j) - \frac{1}{2}\left(f(u_{j+1}^{n+1}) - f(u_{j-1}^n)\right)\right).$$

These updating formulas are essentially Gauss–Seidel iterations because the point values are computed using newly updated neighboring values. Our sweeping method is simply to update point values by Eq. (4), with alternating sweeping directions to achieve a faster convergence rate.

2.1.2. A review of construction of high order finite difference WENO fluxes

To develop a higher order sweeping method, one needs high order numerical fluxes in Eq. (2). In this section, we briefly review the construction of numerical fluxes for high order finite difference WENO schemes, and the same construction will be used in our scheme, presented in the next section. More details can be found in [21,14,5,34–36].

Consider a scalar hyperbolic conservation law in one dimension

$$u_t + f(u)_x = 0$$

with a positive wind direction $f'(u) \geqslant 0$. The spatial derivative is approximated by a conservative flux difference

$$f(u)_x|_{x=x_j} \approx \frac{1}{\Delta x}\left(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}\right)$$

when a uniform discretization with mesh size Δx is used. The numerical flux $\hat{f}_{j+\frac{1}{2}}$ is computed through the neighboring point values $f_i = f(u_i)$. For a $(2k-1)$th order WENO scheme, we first compute k numerical fluxes

$$\hat{f}_{j+\frac{1}{2}}^{(r)} = \sum_{i=0}^{k-1} c_{ri} f_{j-r+i}, \quad r = 0,\ldots,k-1,$$

corresponding to k different candidate stencils $S_r(j) = \{x_{j-r},\ldots,x_{j-r+k-1}\}, r = 0,\ldots,k-1$. Each of these numerical fluxes is kth order accurate. For example, when $k = 2$ (third order WENO scheme), the two second order accurate numerical fluxes are given by

$$\hat{f}_{j+\frac{1}{2}}^{(0)} = \frac{1}{2}f_j + \frac{1}{2}f_{j+1},$$
$$\hat{f}_{j+\frac{1}{2}}^{(1)} = -\frac{1}{2}f_{j-1} + \frac{3}{2}f_j.$$

The $(2k-1)$th order WENO flux is a superposition of all these k numerical fluxes

$$\hat{f}_{j+\frac{1}{2}} = \sum_{r=0}^{k-1} w_r \hat{f}^{(r)}_{j+\frac{1}{2}}.$$

The nonlinear weights w_r satisfy $w_r \geqslant 0, \sum_{r=0}^{k-1} w_r = 1$, and are defined in the following way

$$w_r = \frac{\alpha_r}{\sum_{s=0}^{k-1}\alpha_s}, \quad \alpha_r = \frac{d_r}{(\epsilon+\beta_r)^2}.$$

Here d_r are the linear weights which yield $(2k-1)$th order accuracy, β_r are the so-called "smoothness indicators" of the stencils $S_r(j)$, which measure the smoothness of the function $f(u(x))$ in the stencils. The constant ϵ is a small number used to avoid the division by zero and is typically taken as 10^{-6}. For example, when $k=2$ (third order WENO scheme), the linear weights are given by

$$d_0 = \frac{2}{3}, \quad d_1 = \frac{1}{3},$$

and the smoothness indicators are given by

$$\beta_0 = (f_{j+1} - f_j)^2, \quad \beta_1 = (f_j - f_{j-1})^2.$$

The procedure for the case with $f'(u) \leqslant 0$ is mirror symmetric with respect to $j+\frac{1}{2}$. More details can be found in [14,34].

The Lax–Friedrichs numerical flux is constructed through splitting the fluxes into positive and negative fluxes

$$f^{\pm}(u) = \frac{1}{2}(f(u) \pm \alpha u),$$

where α is taken as $\alpha = \max_u |f'(u)|$. The WENO procedure is applied to $f^{\pm}$ individually with upwind biased stencils. Depending on whether the maximum is taken globally or locally, such schemes are referred to as the Lax–Friedrichs WENO scheme or the local Lax–Friedrichs WENO scheme. In this paper, we will take α to be a global maximum along the line of computation, that is, use the typical Lax–Friedrichs WENO fluxes.

2.1.3. High order Lax–Friedrichs WENO sweeping method

As introduced in Section 2.1.2, high order numerical fluxes $\hat{f}_{j\pm\frac{1}{2}}$ can be constructed through WENO procedure, and here we use Lax–Friedrichs WENO fluxes to obtain the high order sweeping method.

First, we define

$$\hat{\hat{f}}_{j+\frac{1}{2}} = \hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2}(u_{j+1} - u_j), \quad j = 0, \ldots, N-1, \tag{5}$$

where $\hat{f}_{j+\frac{1}{2}}$ is the original high order Lax–Friedrichs flux. The newly defined flux (5) retains the form of Eq. (3) by subtracting the diffusion term explicitly, which allows us to formulate the iterative scheme. The discretization formula then could be written, in terms of $\hat{\hat{f}}_{j+\frac{1}{2}}$, as

$$\frac{(\hat{\hat{f}}_{j+\frac{1}{2}} - \frac{\sigma}{2}(u_{j+1} - u_j)) - (\hat{\hat{f}}_{j-\frac{1}{2}} - \frac{\sigma}{2}(u_j - u_{j-1}))}{\Delta x} = s(u_j, x_j). \tag{6}$$

Thus, we obtain the iterative scheme

$$u_j^{n+1} = \frac{1}{2}(u_{j-1} + u_{j+1}) + \frac{1}{\sigma}\left(\Delta x s(u_j^n, x_j) - (\hat{\hat{f}}_{j+\frac{1}{2}} - \hat{\hat{f}}_{j-\frac{1}{2}})\right). \tag{7}$$

As in the first order sweeping method, the iterations will take alternating directions: If we sweep from left to right, then in (7), $u_{j-1} = u_{j-1}^{n+1}$ and $u_{j+1} = u_{j+1}^n$ are used; if we sweep from right to left, $u_{j-1} = u_{j-1}^n$ and $u_{j+1} = u_{j+1}^{n+1}$ are used.

Upon the convergence of the iterations, high order accuracy will be achieved due to the use of high order numerical fluxes. The WENO construction of the numerical fluxes guarantees essentially non-oscillatory properties of the numerical solutions around discontinuities.

2.1.4. Convergence of the Lax–Friedrichs fast sweeping method

Generally, it is very difficult to study the convergence properties of the high order sweeping methods proposed in Section 2.1.3 due to its nonlinear nature. The analysis of the first order linear Lax–Friedrichs sweeping method for nonlinear hyperbolic equations is also limited because the numerical operators are also fully nonlinear. To gain some insights of the convergence properties with respect to different choices of σ, we perform the analysis on the simplest linear steady state equation:

$$u_x = 1, \quad x \in [0,1], \tag{8}$$

with the boundary condition $u(0) = 0$. Although at $x = 1$, one does not need a boundary condition, numerical boundary conditions are still required, and here we use linear extrapolation at the right boundary.

Let us denote by superscript $+$ the sweep from left to right, and $-$ the sweep from right to left. We also denote the solution by a vector notation $\mathbf{u}^n = (u_1^n, \ldots, u_N^n)^T$ with superscript n representing the number of iterations. If we further denote $\frac{1}{2} - \frac{1}{2\sigma}$ by a and $\frac{1}{2} + \frac{1}{2\sigma}$ by b, then after two alternating iterations the sweeping method can be formulated by

$$A_+(\mathbf{u}^{n+1})^+ = B_+(\mathbf{u}^n)^- + C, \tag{9}$$

and

$$A_-(\mathbf{u}^{n+2})^- = B_-(\mathbf{u}^{n+1})^+ + C, \tag{10}$$

where

$$A_+ = \begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ -b & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & -b & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & -b & 1 & 0 & \cdot \\ \cdot & \cdot & \cdot & 0 & -b & 1 & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & a-b & 1-2a \end{bmatrix},$$

$$B_+ = \begin{bmatrix} 0 & a & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & a & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & a & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & a \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & 0 \end{bmatrix},$$

$$A_- = \begin{bmatrix} 1 & -a & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & -a & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & -a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & -a & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 & -a \\ 0 & \cdot & \cdot & \cdot & 0 & 0 & 1-2a \end{bmatrix},$$

$$B_- = \begin{bmatrix} 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ b & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & b & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & b & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & b & 0 & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & b-a & 0 \end{bmatrix},$$

and $C = (\frac{\Delta x}{\sigma}, \frac{\Delta x}{\sigma}, \ldots, \frac{\Delta x}{\sigma})^T$.

Thus, the updating formula becomes

$$(\mathbf{u}^{n+1})^+ = (A_+)^{-1}B_+(\mathbf{u}^n)^- + (A_+)^{-1}C \tag{11}$$

and

$$(\mathbf{u}^{n+2})^- = (A_-)^{-1}B_-(\mathbf{u}^{n+1})^+ + (A_-)^{-1}C. \tag{12}$$

Combining the above two formulas, one obtains after one complete sweeping:

$$(\mathbf{u}^{n+2})^- = (A_-)^{-1}B_-(A_+)^{-1}B_+(\mathbf{u}^n)^- + (A_-)^{-1}B_-(A_+)^{-1}C + (A_-)^{-1}C. \tag{13}$$

Convergence of the iterative scheme (13) is guaranteed for any fixed N if the spectral radius $\rho\left((A_-)^{-1}B_-(A_+)^{-1}B_+\right)$ is strictly less than 1. Since it is difficult to analyze theoretically the spectral radius of the iteration matrix, we provide the numerical evaluation of the spectral radius for $N = 100$ and $N = 200$, with the value of σ varied from 1 to 20 with increment 0.1. It can

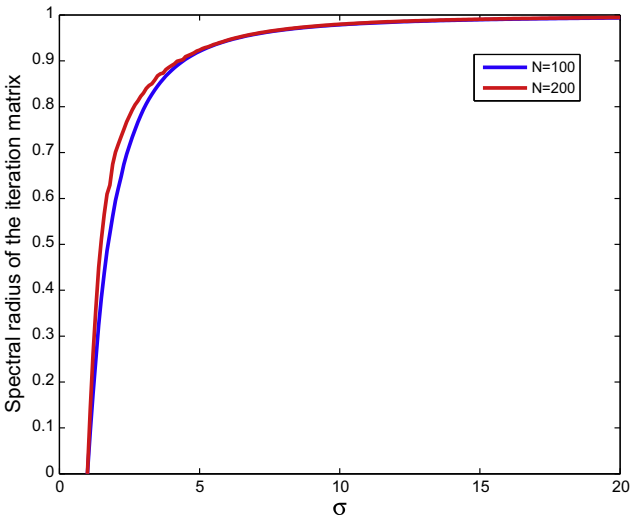

Fig. 1. Spectral radius of the iteration matrix in Eq. (13), with σ varied from 1 to 20 and incremented by 0.1.

be seen from Fig. 1 that the spectral radius is always less than 1, but increasingly approaches 1 as σ becomes larger, and it decreases rapidly as σ is close to 1. Notice that in the scheme we implemented numerically the optimal choice $\sigma = 1$ as showed in Eq. (3) to have rapid convergence. Choice of larger σ may result in more iterations.

2.2. One-dimensional system problems

For one-dimensional systems

$$\mathbf{f}(\mathbf{u})_x = \mathbf{s}(\mathbf{u}, x), \quad x \in [a, b],$$

where $\mathbf{f}$, $\mathbf{s}$ and $\mathbf{u}$ are vector-valued functions in R^m, we use the same iterative formula (7), but perform local characteristic decomposition for numerical flux construction, which is more robust than the component-by-component evaluation. The flux reconstruction procedure is described as follows. First we compute an average state $\mathbf{u}_{j+\frac{1}{2}}$ between $\mathbf{u}_j$ and $\mathbf{u}_{j+1}$, using either the simple arithmetic mean or a Roe's average [28]. The right eigenvectors $\mathbf{r}_m$ and the left eigenvectors $\mathbf{l}_m$ of the Jacobian $\mathbf{f}'(\mathbf{u}_{j+\frac{1}{2}})$ are needed for the local characteristic decomposition. The WENO procedure is used on

$$\mathbf{v}_i^{\pm} = \mathbf{R}^{-1}\mathbf{f}_i^{\pm}, \quad \text{for } i \text{ in a neighborhood of } j,$$

where $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$ is the matrix whose columns are the right eigenvectors of $\mathbf{f}'(\mathbf{u}_{j+\frac{1}{2}})$. The numerical fluxes $\hat{\mathbf{v}}^{\pm}_{j+\frac{1}{2}}$ thus computed are then projected back into the physical space by left multiplying with $\mathbf{R}$, yielding finally the numerical fluxes $\hat{\mathbf{f}}_{j\pm\frac{1}{2}}$ in the physical space. Finally, the grid values $\mathbf{u}_j$ are updated component-wisely by formula (7).

We remark here that the upwind type discretization of the source term $\mathbf{s}(\mathbf{u}, x)$ may be necessary in some equations. In [27], Roe shows that for linear systems, source terms should be upwind in the same way as the physical fluxes. Otherwise, spurious numerical results can appear when the central discretization of the source term is used. Since Lax–Friedrichs flux instead of upwind flux is used in our method, spurious numerical results are not observed frequently. This may due to the fact that Lax–Friedrichs fluxes introduce more numerical dissipation than upwind fluxes. However, in some special cases such as the nozzle flow problem which will be mentioned in Example 5.2.2 of Section 5, WENO approximation and characteristic decomposition of source term are required to maintain well-balanceness and high order accuracy [43].

3. The Lax–Friedrichs sweeping schemes in two dimensions

3.1. Two-dimensional problems

The sweeping method we described in the previous sections can be easily extended to the two-dimensional steady state problem

$$f(u)_x + g(u)_y = s(u, x, y), \quad (x, y) \in [a, b] \times [c, d], \tag{14}$$

Let $\{(x_i, y_j)\}, i = 0, \ldots, N_x, j = 0, \ldots, N_y$ denote the grid points of a uniform discretization of the computational domain, with $\Delta x = (b - a)/N_x$ and $\Delta y = (d - c)/N_y$ as the mesh sizes for x and y direction, respectively. We use $u_{i,j}$ to represent the numerical solution of u at grid point (x_i, y_j). A conservative finite difference discretization of (14) can be written as

$$\frac{\hat{f}_{i+\frac{1}{2}j}-\hat{f}_{i-\frac{1}{2}j}}{\Delta x}+\frac{\hat{g}_{ij+\frac{1}{2}}-\hat{g}_{ij-\frac{1}{2}}}{\Delta y}=s(u_{ij},x_i,y_j). \tag{15}$$

If one uses the first order Lax–Friedrichs fluxes

$$\hat{f}_{i+\frac{1}{2}j}=\frac{1}{2}\big(f(u_{ij})+f(u_{i+1j})\big)-\frac{\sigma_x}{2}\big(u_{i+1j}-u_{ij}\big),$$

$$\hat{g}_{ij+\frac{1}{2}}=\frac{1}{2}\big(g(u_{ij})+g(u_{ij+1})\big)-\frac{\sigma_y}{2}\big(u_{ij+1}-u_{ij}\big)$$

in (15), where $\sigma_x=\max_i\{|f'(u_{ij})|\}$ and $\sigma_y=\max_j\{|g'(u_{ij})|\}$. This leads to the equation

$$\frac{\Delta y}{2}\big(f(u_{i+1j})-f(u_{i-1j})\big)-\frac{\sigma_x\Delta y}{2}\big(u_{i+1j}-2u_{ij}+u_{i-1j}\big)+\frac{\Delta x}{2}\big(g(u_{ij+1})-g(u_{ij-1})\big)-\frac{\sigma_y\Delta x}{2}\big(u_{ij+1}-2u_{ij}+u_{ij-1}\big)$$
$$=\Delta x\Delta y\,s(u_{ij},x_i,y_j).$$

Thus u_{ij} can be solved in terms of neighboring grid function values

$$u_{ij}=\frac{\sigma_x\Delta y\big(u_{i+1j}+u_{i-1j}\big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}+\frac{\sigma_y\Delta x\big(u_{ij+1}+u_{ij-1}\big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}+\frac{\Delta x\Delta y\,s(u^n_{ij},x_i,y_j)}{\sigma_x\Delta y+\sigma_y\Delta x}-\frac{\Delta y\big(f(u_{i+1j})-f(u_{i-1j})\big)+\Delta x\big(g(u_{ij+1})-g(u_{ij-1})\big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}.$$

We sweep the whole domain with four alternating orderings repeatedly,

$$(1)\ \ i=1:I,j=1:J,\quad (2)\ \ i=I:1,j=1:J,\quad (3)\ \ i=1:I,j=J:1,\quad (4)\ \ i=I:1,j=J:1.$$

If the first sweeping direction is chosen, then $u_{i-1j}=u^{n+1}_{i-1j}$, $u_{ij-1}=u^{n+1}_{ij-1}$, $u_{i+1j}=u^{n}_{i+1j}$ and $u_{ij+1}=u^{n}_{ij+1}$. The update formula is

$$u^{n+1}_{ij}=\frac{\sigma_x\Delta y\Big(u^n_{i+1j}+u^{n+1}_{i-1j}\Big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}+\frac{\sigma_y\Delta x\Big(u^n_{ij+1}+u^{n+1}_{ij-1}\Big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}+\frac{\Delta x\Delta y\,s(u^n_{ij},x_i,y_j)}{\sigma_x\Delta y+\sigma_y\Delta x}-\frac{\Delta y(f(u^n_{i+1j})-f(u^{n+1}_{i-1j}))+\Delta x(g(u^n_{ij+1})-g(u^{n+1}_{ij-1}))}{2(\sigma_x\Delta y+\sigma_y\Delta x)}. \tag{16}$$

If the second sweeping direction is chosen, then $u_{i-1j}=u^{n}_{i-1j}$, $u_{ij-1}=u^{n+1}_{ij-1}$, $u_{i+1j}=u^{n+1}_{i+1j}$ and $u_{ij+1}=u^{n}_{ij+1}$. The update formula is

$$u^{n+1}_{ij}=\frac{\sigma_x\Delta y\Big(u^{n+1}_{i+1j}+u^{n}_{i-1j}\Big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}+\frac{\sigma_y\Delta x\Big(u^n_{ij+1}+u^{n+1}_{ij-1}\Big)}{2(\sigma_x\Delta y+\sigma_y\Delta x)}+\frac{\Delta x\Delta y\,s(u^n_{ij},x_i,y_j)}{\sigma_x\Delta y+\sigma_y\Delta x}-\frac{\Delta y(f(u^{n+1}_{i+1j})-f(u^{n}_{i-1j}))+\Delta x(g(u^n_{ij+1})-g(u^{n+1}_{ij-1}))}{2(\sigma_x\Delta y+\sigma_y\Delta x)}. \tag{17}$$

Notice that the indices n and $n+1$ are switched for stencils involving u_{i-1j} and u_{i+1j} in (16) and (17) because the sweeping directions 1 and 2 visit these two points in reverse order. Similarly, the updating formulas in the third and fourth sweeping directions require the switched indices n and $n+1$ for stencils involving u_{ij-1} and u_{ij+1}.

To extend this method to higher order, due to dimension-by-dimension property of the WENO procedure, we can simply replace $f(u_{i\pm1j})$ by $\hat{f}(u_{i\pm1j})$ and $g(u_{ij\pm1})$ by $\hat{g}(u_{ij\pm1})$, where $\hat{f}(u_{i\pm1j})$ and $\hat{g}(u_{ij\pm1})$ are the modified Lax–Friedrichs WENO fluxes in x and y directions defined in (5).

For two-dimensional system problems

$$\mathbf{f}(\mathbf{u})_x+\mathbf{g}(\mathbf{u})_y=\mathbf{s}(\mathbf{u},x,y),\quad x\in[a,b],$$

where $\mathbf{f}$, $\mathbf{g}$, $\mathbf{s}$ and $\mathbf{u}$ are vector-valued functions in R^m, we use the same two-dimensional iterative formula, but perform local characteristic decomposition for numerical flux construction as follows. Assume $\mathbf{f}'(\mathbf{u})$ and $\mathbf{g}'(\mathbf{u})$ can be written as $L_x\Lambda_xR_x$ and $L_y\Lambda_yR_y$, respectively, where Λ_x and Λ_y are diagonal matrices with real eigenvalues on the diagonal, and L_x,R_x and L_y,R_y are matrices of left and right eigenvectors for the corresponding Jacobians. Similar to 1-D system, we first compute an average state $\bar{u}$ in each cell $[x_i,x_{i+1}]\times[y_j,y_{j+1}]$, using either the simple arithmetic mean or a Roe's average [28]. We denote $\bar{L}_x,\bar{R}_x$ as the matrices with left and right eigenvectors L_x,R_x of $\mathbf{f}'(\mathbf{u})$ at the average state, and $\bar{\lambda}^k_x$ as the corresponding eigenvalues. The matrices $\bar{L}_y,\bar{R}_y$ and the eigenvalues $\bar{\lambda}^k_y$ are defined similarly but associated with L_y,R_y, and Λ_y of $\mathbf{g}'(\mathbf{u})$.

The WENO procedure is used on the construction of fluxes:

$$\mathbf{\Phi}^{\pm}_p=\overline{\mathbf{R}}^{-1}_{\mathbf{x}}\mathbf{f}^{\pm}_p,\quad \text{for } p \text{ in a neighborhood of } i,$$

$$\mathbf{\Psi}^{\pm}_q=\overline{\mathbf{R}}^{-1}_{\mathbf{y}}\mathbf{g}^{\pm}_q,\quad \text{for } q \text{ in a neighborhood of } j,$$

The numerical fluxes $\hat{\mathbf{\Phi}}^{\pm}_{i+\frac{1}{2}}$ and $\hat{\mathbf{\Psi}}^{\pm}_{j+\frac{1}{2}}$ are then projected back into the physical space by left multiplying with $\overline{\mathbf{R}}_{\mathbf{x}}$ and $\overline{\mathbf{R}}_{\mathbf{y}}$, yielding finally the numerical fluxes $\hat{\mathbf{f}}_{i\pm\frac{1}{2}}$ and $\hat{\mathbf{g}}_{j\pm\frac{1}{2}}$ in the physical space.

4. Stopping criteria for convergence

Iterative schemes always require stopping criteria by which the convergence of the numerical scheme is determined. Traditional Gauss–Seidel iterations adopt the stopping criteria in which the algorithm stops when a particular norm of the difference of successive iterations, called residue, is smaller than a number. This stopping criteria is effective when the residue of the scheme is monotonically decreasing, but not robust enough for schemes with oscillatory convergence history, such as high order fast sweeping methods.

In [31], it was proposed that, for first order sweeping methods for Hamilton–Jacobi equations, the algorithm stops as

$$q_n = \frac{\|u^{n+1} - u^n\|}{\|u^n - u^{n-1}\|} > 1,$$

where u^k denotes the numerical solution at k-th iteration. When the scheme defines a contraction mapping, this criterion indicates that the residue saturates due to spatial resolution and rounding error. A modified version of the criterion was proposed for high order fast sweeping methods for Hamilton–Jacobi equations:

$$q_n = \frac{\|u^{n+1} - u^n\| + h^p}{\|u^n - u^{n-1}\|} > 1, \quad n \geqslant 1, \tag{18}$$

where p is the order of the scheme. The inclusion of h^p term is to stop the algorithm when the distance between two consecutive iterations is much smaller than the truncation error of the scheme.

In this paper, we adopt the stopping criterion (18), along with

$$\|u^{n+1} - u^n\| < h^{p+1}. \tag{19}$$

This addition of criterion (19) is to exclude the possibility that the algorithm stops in early iterations where the residue fluctuates wildly, and at the same time to ensure the order of accuracy of the scheme.

5. Numerical results

Here we show the numerical results of the proposed Lax–Friedrichs WENO sweeping method for hyperbolic conservation laws with source terms in both scalar and system test problems in one and two dimensions. The efficiency and high order accuracy of the proposed scheme will be demonstrated.

All the spatial discretizations in our numerical results are uniform. The values of σ in Lax–Friedrichs fluxes and α in the flux splitting are updated at the beginning of each directional sweeping and relaxed by multiplying some constant greater than 1 to $\max_j\{|f'(u_j)|\}$. In most problems, the L^1 residues can be reduced to the level that satisfies stopping criteria (18) and (19). For cases in which the residues stagnate above the desired level, we will state in the problem. In some problems, we compare the third order sweeping method with a time marching method, the third order Lax–Friedrichs WENO finite difference method with fourth order TVD Runge–Kutta time integration [35], and we abbreviate the latter as WENO3-RK4. The CFL numbers in WENO3-RK4 are taken to be 0.6 in all problems. We remark here that the choice of the CFL number certainly affects the number of iterations to reach a steady state, but here we choose it to be sufficiently large while maintaining stability for all cases. We also test other methods of time integration such as RK2 and RK3, the performance and iteration numbers are very close to that of RK4. In WENO reconstruction, ϵ is set to be 10^{-10} in most cases, unless otherwise stated.

5.1. The one-dimensional scalar problems

Example 5.1.1. We solve the steady state solution of the one-dimensional Burgers' equation with a source term:

$$u_t + \left(\frac{u^2}{2}\right)_x = \sin x \cos x \tag{20}$$

with the initial condition

$$u(x, 0) = \beta \sin x \tag{21}$$

and boundary conditions $u(0, t) = u(\pi, t) = 0$. This problem was studied in [30] as an example of multiple steady state solutions for characteristic initial value problems. The steady state solution to this problem depends on the value of β: if $-1 < \beta < 1$, a shock will form within the domain $[0, \pi]$; otherwise, the solution will be smooth at first, followed by a shock forming at the boundary $x = \pi$ ($\beta \geqslant 1$) or $x = 0$ ($\beta \leqslant -1$), and later converge to a smooth steady state $u(x, \infty) = \sin x (\beta \geqslant 1)$ or $u(x, \infty) = -\sin x (\beta \leqslant -1)$, respectively. In order to test the order of accuracy, we take $\beta = 2$ to have a smooth stationary solution. Extended exact solutions are imposed on the ghost points of WENO reconstruction in the test of our fast sweeping method with Lax–Friedrichs WENO (LF-WENO3) fluxes. The numerical results are shown in Table 1. The convergence to third order accuracy of L^1 error can be clearly observed. The order of accuracy of L^∞ error converges to third order much more

Table 1
Errors, numerical orders of accuracy and iteration numbers of fast sweeping method with the third order finite difference LF-WENO3 fluxes for Example 5.1.1 on meshes with N cells.

N	L^1 Error	Order	L^∞ Error	Order	Iter #
40	3.19E−03	–	3.25E−03	–	134
80	4.87E−04	2.71	8.43E−04	1.95	314
160	6.90E−05	2.82	2.14E−04	1.98	754
320	1.02E−05	2.76	5.37E−05	1.99	1794
640	1.45E−06	2.81	1.21E−05	2.15	4110
1280	1.89E−07	2.94	2.13E−06	2.51	9208
2560	2.27E−08	3.06	1.28E−07	4.06	19818

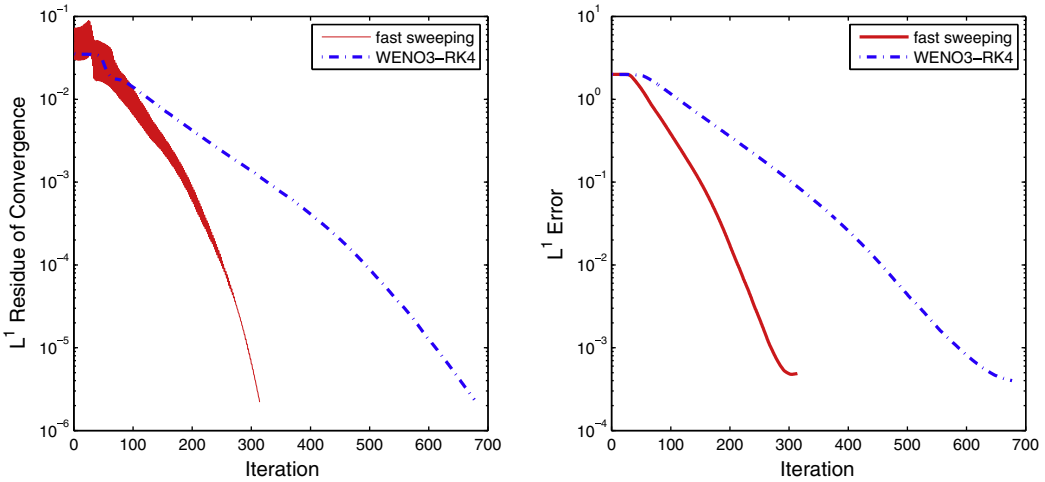

Fig. 2. Comparison of the L^1 residues (left) and L^1 errors (right) of Example 5.1.1 in fast sweeping method (red solid line) and time marching method WENO3-RK4 (blue dotted line) with $N = 80$ cells for $\beta = 2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

slowly. This slow convergence of L^∞ error has been observed in [44] because the solutions for WENO3 are more dissipative at the extrema in coarse grids.

We compare the fast sweeping method with WENO3-RK4 by plotting the residue for each iteration until the solutions for both methods reach the same stopping criteria. The comparison is displayed in Fig. 2. We can see that the sweeping method converges faster than time marching scheme to reach a sufficiently small residue, and it approaches the exact solution more quickly. In our example, fast sweeping method converges at an obviously higher rate if one requires the residue lower than 10^{-4}.

Example 5.1.2. We consider the same problem as in Example 5.1.1., but take $\beta = 0.5$ in the initial condition (21). As mentioned in the previous example, when $-1 < \beta < 1$, a shock will form within the domain, which separates two branches ($\sin x$ and $-\sin x$) of the steady state. The location of the shock is determined by the parameter β through conservation of mass ($\int_0^\pi u\,dx = 2\beta$), and can be derived to be $\pi - \arcsin\sqrt{1-\beta^2}$. For the case $\beta = 0.5$, the shock location is approximately 2.0944.

Unlike the regular conservative time evolution schemes, our Gauss–Seidel iteration does not conserve the mass over the iterations, but rather aims at solving the steady state equation directly. Therefore, the conservation of mass needs to be imposed along with the steady state equation itself to make this problem well-posed. Numerically, we impose the conservation of mass, within each iteration, by linearly projecting the numerical solution to the space in which the discrete mass $\sum_{j=0}^N u_j \Delta x$ is conserved and equal to $\sum_{j=0}^N u_0(x_j)\Delta x$, where $u_0(x)$ is the initial condition. In our test, the projection is realized by applying the operation $u_j = \frac{u_j}{\sum_{j=0}^N u_j \Delta x}\sum_{j=0}^N u_0(x_j)\Delta x$ after each direction of sweeping.

We present our numerical solution, with 160 cells, in Fig. 3. One can see that the shock is captured at the correct location and it is resolved very well.

Example 5.1.3. We consider the steady state solutions of Burgers' equation with a different source term which depends on the solution itself:

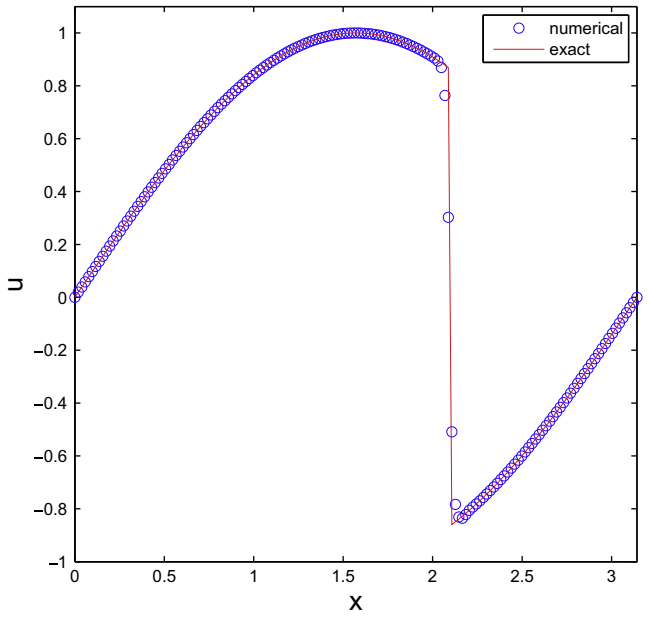

Fig. 3. The numerical solution (blue circles) versus the exact solution (red solid line) for Example 5.1.2, computed with 160 cells after 750 iterations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$u_t + \left(\frac{u^2}{2}\right)_x = -\pi \cos(\pi x) u, \quad x \in [0,1] \tag{22}$$

equipped with the boundary conditions $u(0,t) = 1$ and $u(1,t) = -0.1$. This problem has two steady states with shocks

$$u(x,\infty) = \begin{cases} u^+ = 1 - \sin(\pi x) & \text{if } 0 \leqslant x < x_s, \\ u^- = -0.1 - \sin(\pi x) & \text{if } x_s \leqslant x < 1, \end{cases}$$

where $x_s = 0.1486$ or $x_s = 0.8514$. Both solutions satisfy the Rankine–Hugoniot jump condition and the entropy conditions, but only the one with the shock at 0.1486 is stable for small perturbation. This problem was studied in [13] as an example of multiple steady states for one-dimensional transonic flows. This case is tested to demonstrate that starting with a reasonable perturbation of the stable steady state, the numerical solution converges to the stable one by the fast sweeping method.

The initial guess is given by:

$$u(x,0) = \begin{cases} 1 & \text{if } 0 \leqslant x < 0.5, \\ -0.1 & \text{if } 0.5 \leqslant x < 1, \end{cases}$$

where the initial jump is located in the middle of the positions of the shocks in the two admissible steady state solutions. Exact solutions are imposed on the ghost points of WENO reconstruction. The numerical result with 80 cells and the exact solution are displayed in Fig. 4. We can see the correct shock location captured and good resolution of the shock.

We compare the fast sweeping method with WENO3-RK4 by plotting the solutions after the same number of iterations and the residues between two consecutive iterations, as well as the L^1 error for each iteration until the solutions for both methods reach the same stopping criteria. The comparison is displayed in Fig. 5. It can be seen that the numerical solution by fast sweeping method approaches the exact solution much faster than WENO3-RK4, and the accurate shape of the solution can be obtained after very few iterations by sweeping method.

We also notice that if only one directional sweeping is applied in the iteration, the numerical solution may converge to a wrong steady state. In Fig. 6, we compare the exact solution with the numerical solution of our method with only right-to-left sweeping. The single directional sweeping fails to capture the exact shock location.

5.2. The one-dimensional systems

Example 5.2.1. We solve the steady state solutions of the one-dimensional shallow water equation:

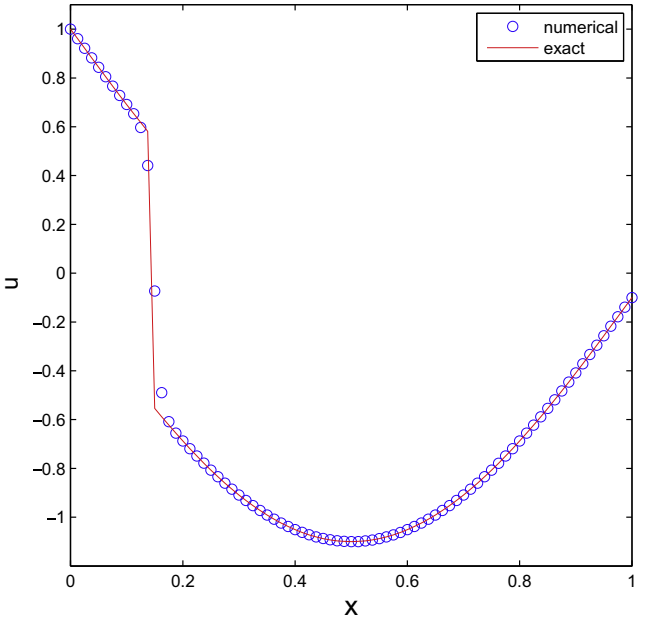

Fig. 4. The stable entropy solution (red solid line), and the numerical solution (blue circles) for Example 5.1.3, computed with 80 cells after 450 iterations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\begin{pmatrix} h \\ hu \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}_x = \begin{pmatrix} 0 \\ -ghb_x \end{pmatrix}, \tag{23}$$

where h denotes the water height, u is the velocity of the fluid, $b(x)$ represents the bottom topography and g is the gravitational constant.

Starting from a stationary initial condition, which itself is a steady state solution, we can check the order of accuracy. The smooth bottom topography is given by:

$$b(x) = 5e^{-\frac{2}{5}(x-5)^2}, \quad x \in [0, 10].$$

The initial condition is the stationary solution:

$$h + b = 10, \qquad hu = 0$$

and the exact steady state solution is imposed at the boundaries.

The numerical results are shown in Table 2. We can clearly see the third order accuracy of L^1 error, but the order of L^∞ error converges much slower. In this example we choose $\epsilon = 10^{-6}$ in the WENO3 construction.

Example 5.2.2. We test our scheme on the steady state solution of the one-dimensional nozzle flow problem:

$$\begin{pmatrix} \rho A \\ \rho uA \\ EA \end{pmatrix}_t + \begin{pmatrix} \rho uA \\ (\rho u^2 + p)A \\ uA(E+p) \end{pmatrix}_x = \begin{pmatrix} 0 \\ pA'(x) \\ 0 \end{pmatrix}, \quad x \in [0, 3] \tag{24}$$

where ρ denotes the density, u is the velocity of the fluid, E is the total energy, γ is the gas constant, which is taken as 1.4, $p = (\gamma - 1)(E - \frac{1}{2}\rho u^2)$ is the pressure, and $A(x)$ represents the area of the cross section of the nozzle.

We start with an initial condition

$$\rho(x, 0) = 1, \;\; u(x, 0) = 0 \;\text{ and }\; p(x, 0) = 1. \tag{25}$$

In our test, $u = [\rho A \;\; \rho uA \;\; EA]^T$ is updated by fast sweeping. The boundary conditions are taken as 1 for pressure fixed at the left, 0.6784 for pressure fixed at the right. For the component $u_1 = \rho A$ and $u_2 = \rho uA$, extended boundary condition is imposed. The area of the cross section $A(x)$ is given by:

$$A(x) = 1 + 2.2(x - 1.5)^2, \quad 0 \leqslant x \leqslant 3.$$

In this one-dimensional system, WENO finite difference approximation is applied to both the flux term and $A'(x)$, and the characteristic decomposition is also applied to both the flux term and the source term in order to achieve high order accuracy

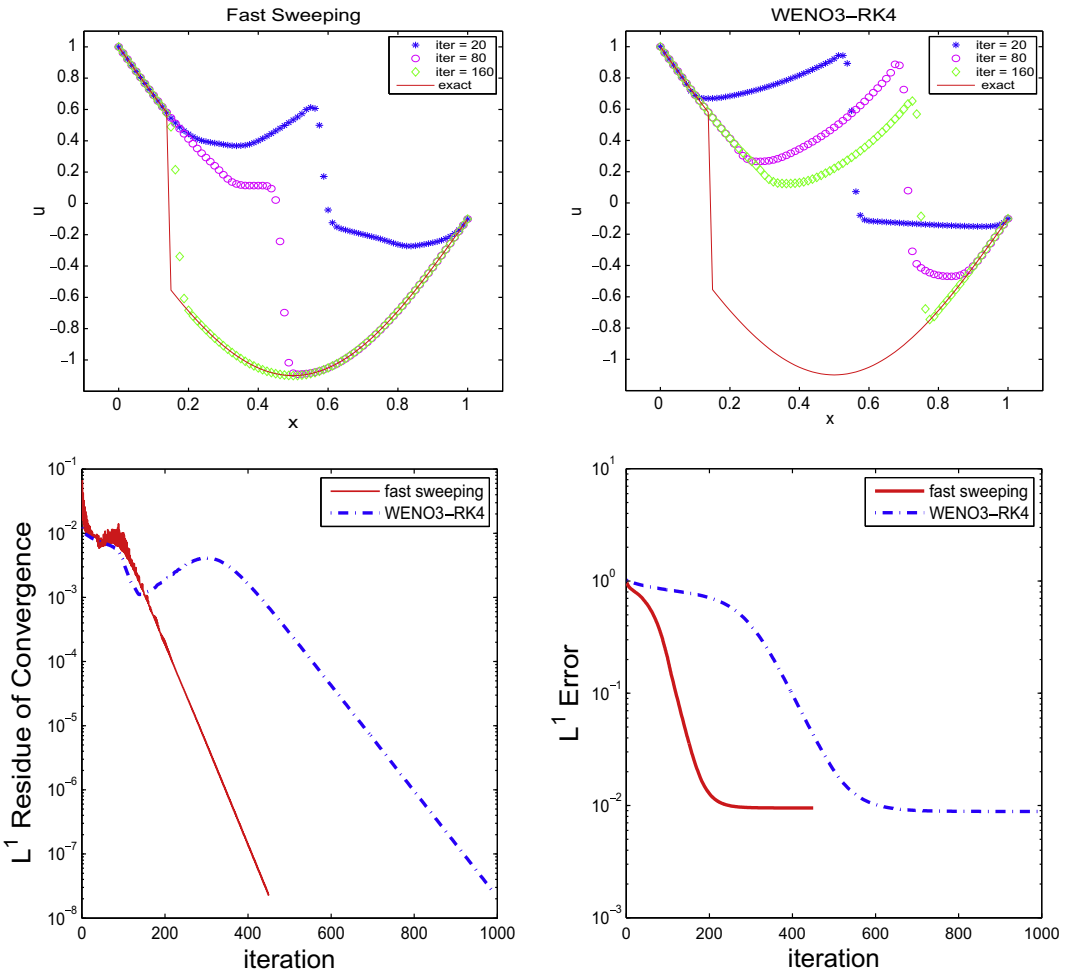

Fig. 5. Example 5.1.3 simulated with 80 cells. Top left: the solution obtained by sweeping method after 20, 80, 160 iterations. Top right: the solution obtained by WENO3-RK4 after 20, 80, 160 iterations. Bottom left: comparison of L^1 residues of convergence by fast sweeping and WENO3-RK4 methods. Bottom right: comparison of L^1 errors by these two methods.

and ensure the conservation of discretized solutions in space. In the numerical implementation, the matrix $[0\ A\ 0]^T$ is decomposed into characteristic space and the derivative $[0\ A'\ 0]^T$ is constructed by the WENO3 procedure.

In Fig. 7, the numerical solution of pressure by fast sweeping method is compared with the simulated solution with very fine mesh (1000 cells) using WENO3-RK4, which is very close to the exact solution. One can see that the shock is resolved well.

We also analyze the convergence speed by displaying the numerical solutions of fast sweeping method and WENO3-RK4 after the same number of iterations, and the history of L^1 residues and L^1 errors. The results are shown in Fig. 8. We can see that fast sweeping method approaches the exact solution much more quickly. The L^1 error by fast sweeping stabilizes around 10^{-2} after 1500 iterations, while the time marching method achieves the same level of error after 20000 iterations. However, the residue of convergence for fast sweeping method will stagnate around 10^{-4} and for time marching it can go down slowly to 10^{-7}, this may be due to the natural mechanism of fast sweeping iteration.

5.3. The two-dimensional scalar problems

Example 5.3.1. We solve the steady state problem of two-dimensional Burgers' equation with a source term:

$$u_t + \left(\frac{1}{\sqrt{2}}\frac{u^2}{2}\right)_x + \left(\frac{1}{\sqrt{2}}\frac{u^2}{2}\right)_y = \sin\left(\frac{x+y}{\sqrt{2}}\right)\cos\left(\frac{x+y}{\sqrt{2}}\right), \tag{26}$$

where $(x,y) \in \left[0,\frac{\pi}{\sqrt{2}}\right] \times \left[0,\frac{\pi}{\sqrt{2}}\right]$ with the initial condition given by

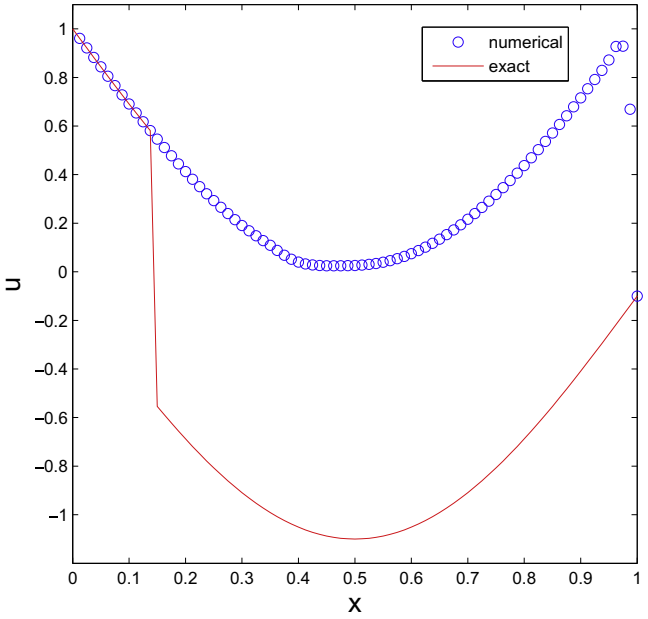

Fig. 6. The stable entropy solution (red solid line), and the numerical solution (blue circles) for Example 5.1.3 by only one directional sweeping, computed with 80 cells after 1451 iterations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Errors, numerical orders of accuracy and iteration numbers for the water height h by fast sweeping method with LF-WENO3 flux for Example 5.2.1 on meshes with N cells.

N	L^1 Error	Order	L^∞ Error	Order	Iter #
160	7.99E−04	–	2.33E−03	–	232
320	1.03E−04	2.96	5.75E−04	2.00	428
640	1.30E−05	2.98	1.40E−04	2.04	864
1280	1.13E−06	3.52	2.01E−05	2.80	1114
2560	1.19E−07	3.25	3.70E−06	2.45	2110
5120	8.41E−09	3.82	4.15E−07	3.15	3776

$$u(x,y,0) = \beta \sin\left(\frac{x+y}{\sqrt{2}}\right). \tag{27}$$

This is the one-dimensional problem studied in Example 5.1.1 along the northeast-southwest diagonal. Since our grids are not aligned with the diagonal, this is a truly two-dimensional test case. Here we take the boundary conditions to be the exact solution of the steady state problem.

For this example we take $\beta = 1.5$, which gives a smooth steady state solution $u(x,y,\infty) = \sin\left(\frac{x+y}{\sqrt{2}}\right)$. The errors and numerical orders are shown in Table 3. It can be seen clearly that third order accuracy is achieved.

Example 5.3.2. We consider the steady state solution of the following problem:

$$u_t + \left(\frac{1}{\sqrt{2}}\frac{u^2}{2}\right)_x + \left(\frac{1}{\sqrt{2}}\frac{u^2}{2}\right)_y = -\pi \cos\left(\pi \frac{x+y}{\sqrt{2}}\right) u, \tag{28}$$

where $(x,y) \in \left[0, \frac{1}{\sqrt{2}}\right] \times \left[0, \frac{1}{\sqrt{2}}\right]$. This is the one-dimensional problem in Example 5.1.3. along the northeast-southwest diagonal. Inflow boundary conditions are given by the exact solution of the steady state problem. Again, since our grids are not aligned with the diagonal line, this is a truly two-dimensional test case. As before, this problem has two steady state solutions with shocks

$$u(x,y,\infty) = \begin{cases} 1 - \sin\left(\pi \frac{x+y}{\sqrt{2}}\right) & \text{if } 0 \leqslant \frac{x+y}{\sqrt{2}} < x_s, \\ -0.1 - \sin\left(\pi \frac{x+y}{\sqrt{2}}\right) & \text{if } x_s \leqslant \frac{x+y}{\sqrt{2}} < 1, \end{cases}$$

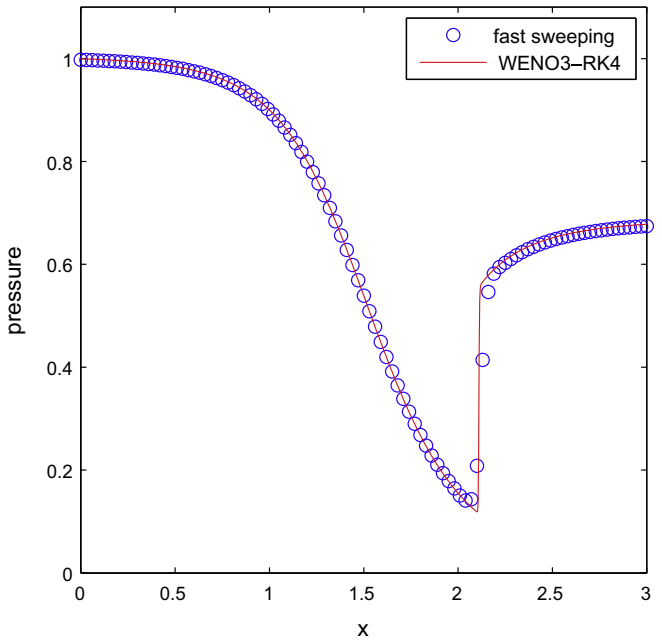

Fig. 7. Example 5.2.2. Solid red lines: exact solution approximated by WENO3-RK4 with very fine mesh (1000 cells); blue circles: numerical solution by fast sweeping method with 100 cells. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where $x_s = 0.1486$ or $x_s = 0.8514$.

The initial condition is given by:

$$u(x,y,0) = \begin{cases} 1 & \text{if } 0 \leqslant \frac{x+y}{\sqrt{2}} < 0.5, \\ -0.1 & \text{if } 0.5 \leqslant \frac{x+y}{\sqrt{2}} < 1, \end{cases}$$

where the initial jump is located in the middle of the positions of the shocks in the two admissible steady state solutions. The numerical result is displayed in Fig. 9. The shock is captured at the correct location and resolved well.

We compare the fast sweeping method with WENO3-RK4 time evolution scheme by plotting the residues and errors against the iteration numbers. The results are shown in Fig. 10. We can see that fast sweeping method reaches the same stopping criteria with much fewer iterations and approaches the exact solution more efficiently.

Example 5.3.3. We consider the two-dimensional steady state problem:

$$u_t + \left(\frac{u^2}{2}\right)_x + u_y = 0, \quad (x,y) \in [0,1] \times [0,1] \tag{29}$$

with the boundary conditions:

$$u(x,0,t) = 1.5 - 2x, \quad u(0,y,t) = 1.5, \quad u(1,y,t) = -0.5.$$

This problem was studied in [6] as a prototype example for shock boundary layer interaction. The initial condition is taken to be $u(x,y,0) = u(x,0,0) = 1.5 - 2x$. Exact solutions are imposed on the ghost points of WENO reconstruction, which means u is set to be constant for $x < 0, x > 1, y > 1$ and equals $1.5 - 2x$ for $y < 0$. The isolines of the numerical solution and the cross sections for $y = 0.25$ across the fan, for $y = 0.5$ right at the junction where the fan becomes a single shock, and at $y = 0.75$ across the shock, are displayed in Fig. 11. It can be seen from the cross section figures that both the smooth and shock regions are well-resolved.

5.4. The two-dimensional systems

Example 5.4.1. We consider a Cauchy–Riemann problem:

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} + B\frac{\partial W}{\partial y} = 0, \qquad (x,y) \in [-2,2] \times [-2,2], \quad t > 0, \tag{30}$$

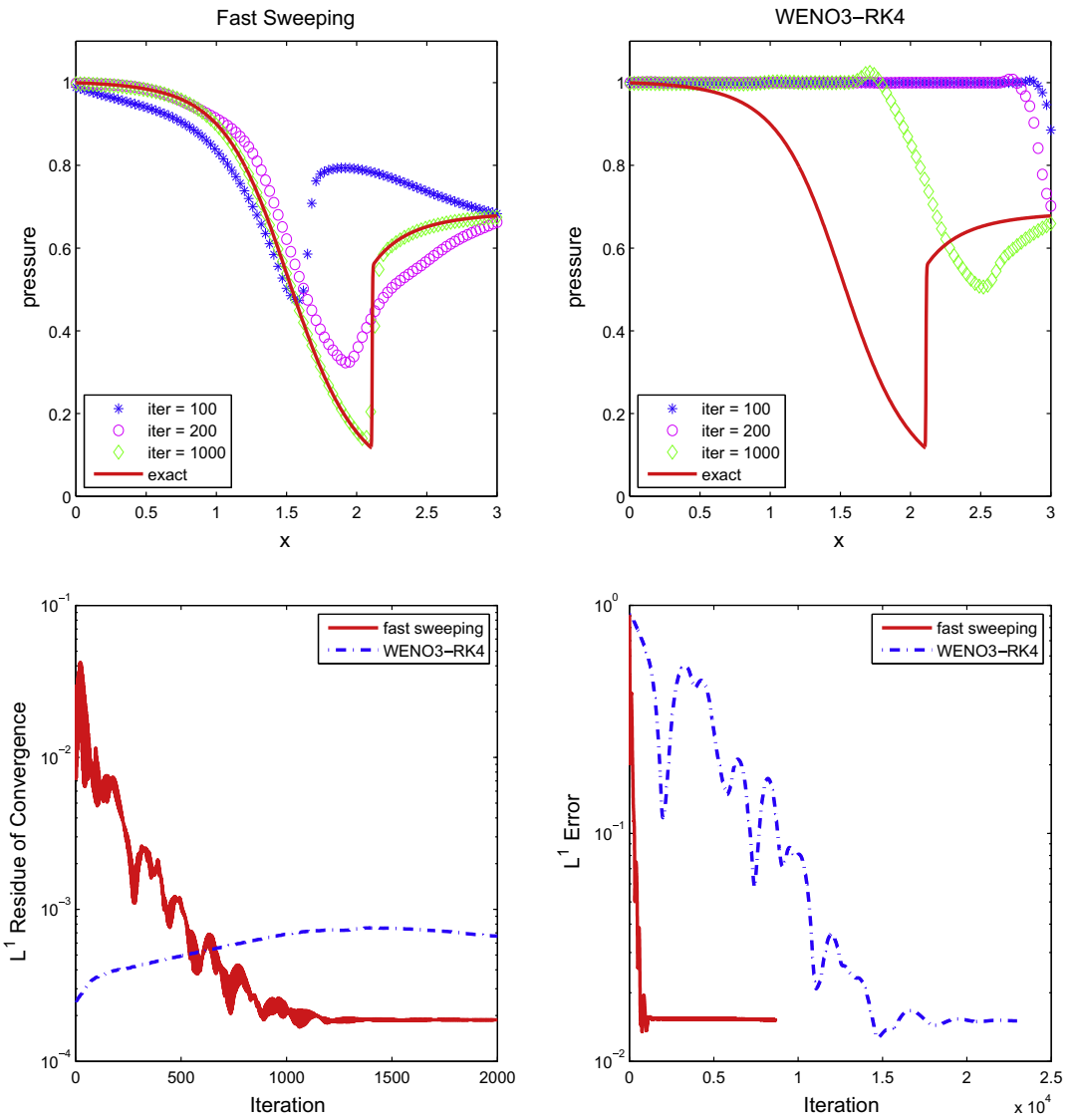

Fig. 8. Comparison of fast sweeping method with WENO3-RK4 for Example 5.2.2, with 100 cells. Top left: the solution obtained by sweeping method after 100, 200, 1000 iterations. Top right: the solution obtained by WENO3-RK4 method after 100, 200, 1000 iterations. Bottom left: comparison of L^1 residues for convergence by fast sweeping and time marching methods for first 2000 iterations. Bottom right: comparison of L^1 errors by these two methods.

Table 3
Errors, numerical orders of accuracy and iteration numbers of fast sweeping method with LF-WENO3 flux for Example 5.3.1 on meshes with $N \times N$ cells.

$N \times N$	L^1 Error	Order	L^∞ Error	Order	Iter #
20×20	4.38E−03	–	2.92E−03	–	284
40×40	6.93E−04	2.66	7.26E−04	2.00	368
80×80	1.02E−04	2.76	1.80E−04	2.01	528
160×160	1.46E−05	2.80	4.48E−05	2.01	996
320×320	1.92E−06	2.93	1.05E−05	2.09	1872
640×640	2.11E−07	3.19	1.08E−06	3.28	4588

where

$$A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

with the following Riemann data $W = (u, v)^T$,

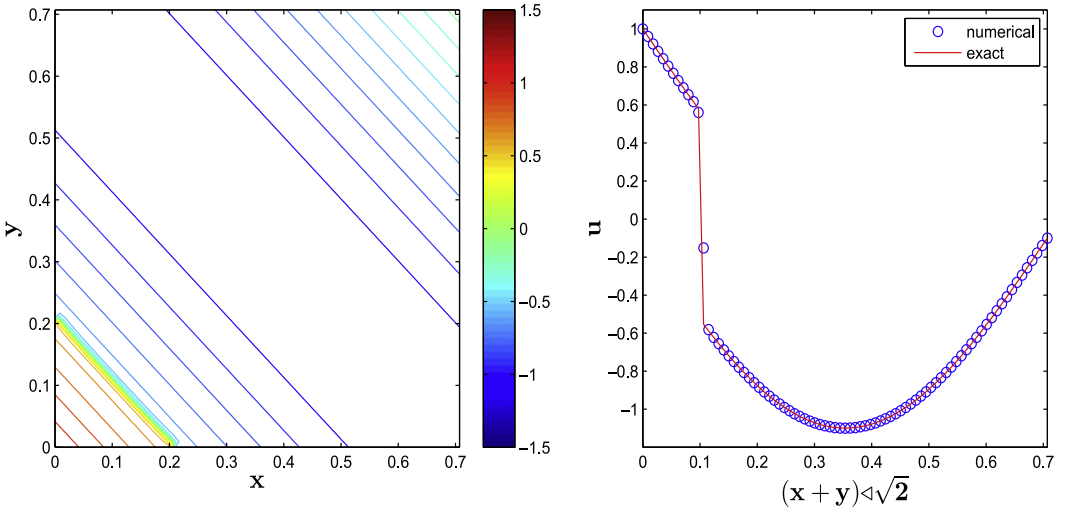

Fig. 9. Example 5.3.2 simulated on 80×80 cells. Left: 25 equally spaced contours of the solution from -1.2 to 1.1; right: the numerical solution (blue circles) versus the exact solution (red solid line) along the cross section through the northeast to southwest diagonal after 564 iterations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

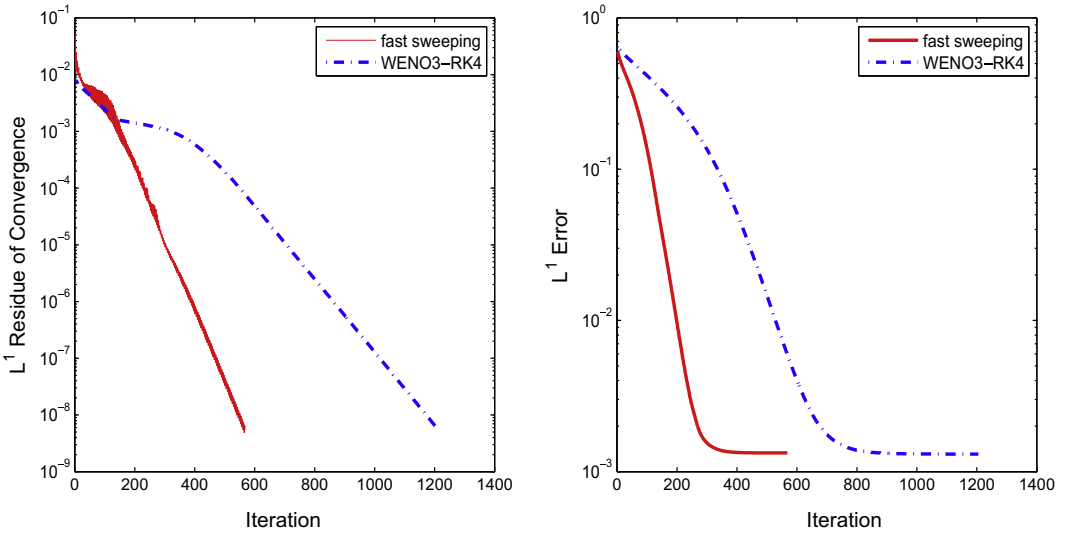

Fig. 10. Convergence history for Example 5.3.2. The L^1 residues for convergence (left) and errors (right) by fast sweeping method (red solid lines) and WENO3-RK4 time evolution method (blue dotted lines) on a mesh with 80×80 cells. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$u = \begin{cases} 1 & \text{if } x > 0 \text{ and } y > 0 \\ -1 & \text{if } x < 0 \text{ and } y > 0 \\ -1 & \text{if } x > 0 \text{ and } y < 0 \\ 1 & \text{if } x < 0 \text{ and } y < 0 \end{cases} \text{ and } v = \begin{cases} 1 & \text{if } x > 0 \text{ and } y > 0 \\ -1 & \text{if } x < 0 \text{ and } y > 0 \\ -1 & \text{if } x > 0 \text{ and } y < 0 \\ 2 & \text{if } x < 0 \text{ and } y < 0 \end{cases}. \tag{31}$$

The solution is self-similar, and therefore $W(x,y,t) = \widetilde{W}(\frac{x}{t}, \frac{y}{t})$. Let $\xi = \frac{x}{t}$, $\eta = \frac{y}{t}$, then $\widetilde{W}$ satisfies

$$(-\xi I + A)\frac{\partial \widetilde{W}}{\partial \xi} + (-\eta I + B)\frac{\partial \widetilde{W}}{\partial \eta} = 0 \tag{32}$$

which can be written as

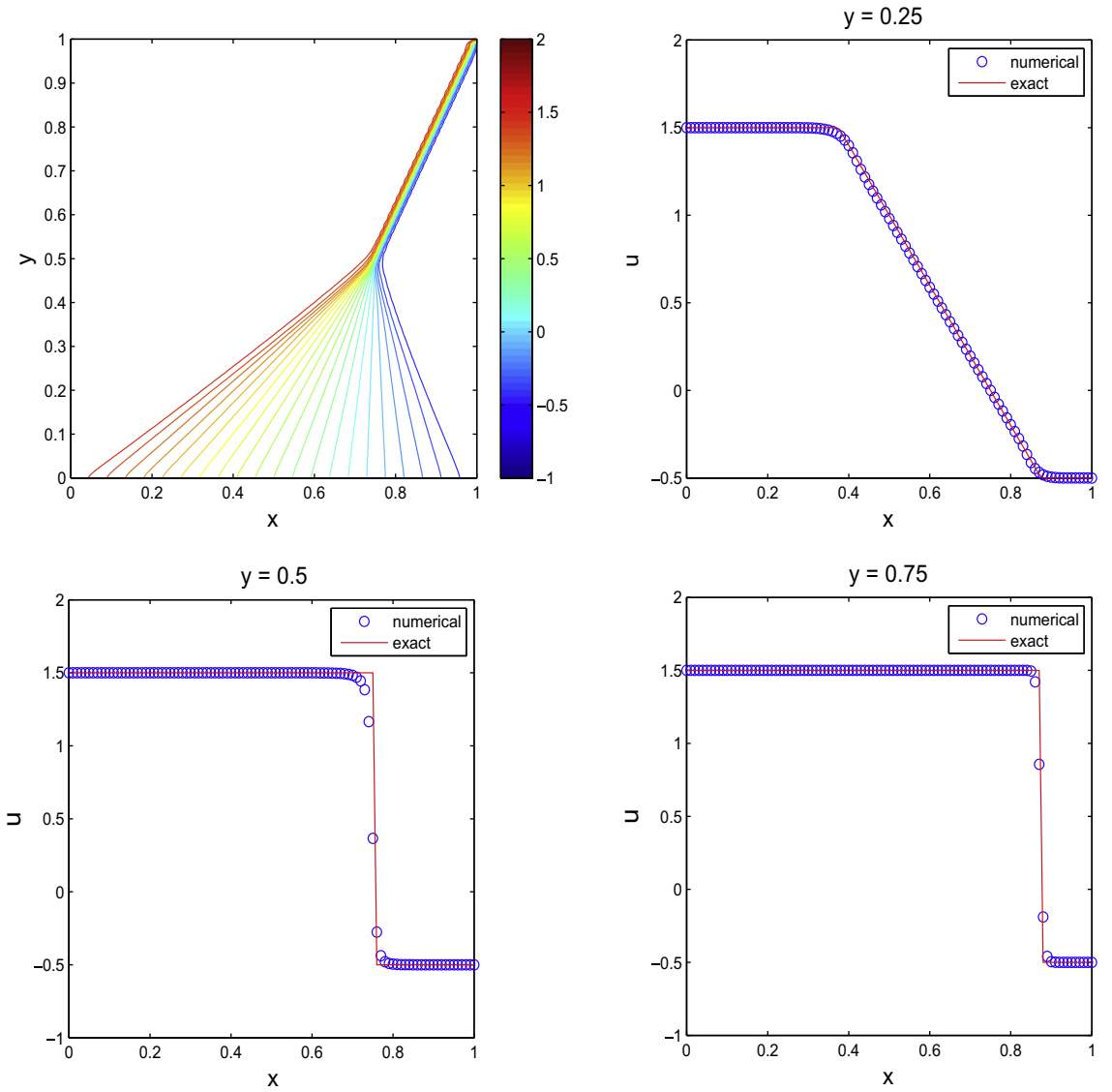

Fig. 11. Example 5.3.3 on a mesh with 100×100 cells. Top left: 25 equally spaced contour lines from -0.6 to 1.6. Top right: cross section at $y = 0.25$; bottom left: cross section at $y = 0.5$; bottom right: cross section at $y = 0.75$. For the cross sections, the solid red lines are the exact solution and blue circles are the numerical solution. Results are obtained after 176 iterations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\frac{\partial}{\partial \xi}\left[(-\xi I + A)\widetilde{W}\right] + \frac{\partial}{\partial \eta}\left[(-\eta I + B)\widetilde{W}\right] = -2\widetilde{W} \tag{33}$$

with the boundary conditions at infinity given by the Riemann data in (30) and (34) at time $t = 1$. Eq. (33) can be solved by the sweeping method with boundary conditions set as the exact solution and the same initial condition as in (34). Boundary values on the ghost points in the WENO reconstruction are set as

$$u = \begin{cases} 1 & \text{if } x > 1 \text{ and } y > 1 \\ -1 & \text{if } x > 1 \text{ and } y < 1 \\ -1 & \text{if } x < 1 \text{ and } y > 1 \\ 1.5 & \text{if } x < 1 \text{ and } -1 < y < 1 \\ 1 & \text{if } x < 1 \text{ and } y < -1 \end{cases} \quad \text{and} \quad v = \begin{cases} 1 & \text{if } x > -1 \text{ and } y > 1 \\ -1 & \text{if } x < -1 \text{ and } y > 1 \\ -1 & \text{if } x > -1 \text{ and } y < 1 \\ 1.5 & \text{if } x < -1 \text{ and } -1 < y < 1 \\ 2 & \text{if } x < -1 \text{ and } y < -1 \end{cases}. \tag{34}$$

The numerical results are shown in Fig. 12.

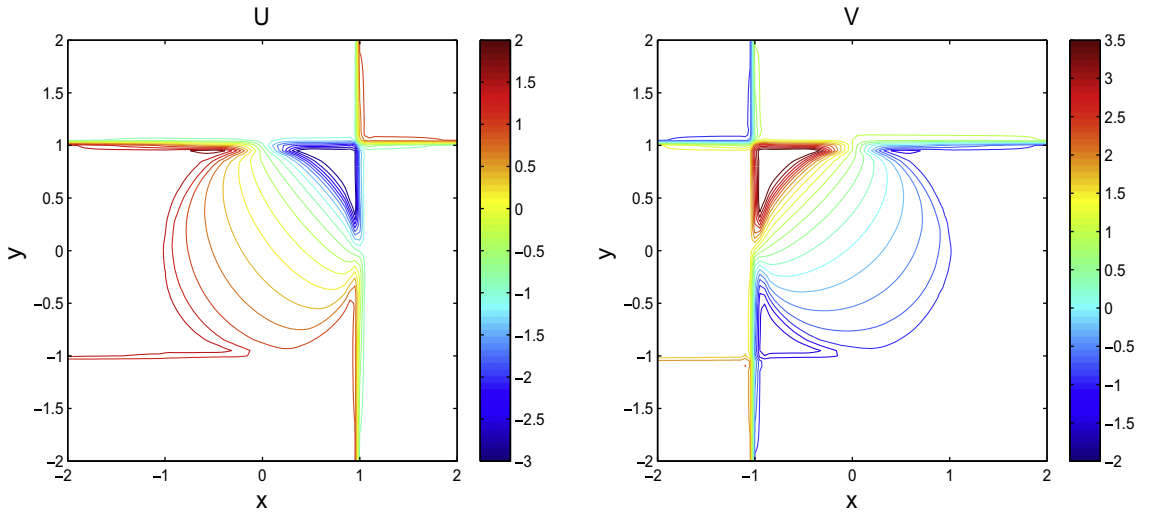

Fig. 12. Example 5.4.1 simulated on 80×80 cells. Left panel: 19 equally spaced contours for u from -2.9 to 1.7; right panel: 23 equally spaced contours for v from -1.9 to 3.37. Total iteration number is 364.

Example 5.4.2. We consider a regular shock reflection problem of the steady state solution of the two-dimensional Euler equations:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = 0, \quad (x,y) \in [0,4] \times [0,1], \tag{35}$$

where $\mathbf{u} = (\rho, \rho u, \rho v, E)^T$, $\mathbf{f}(\mathbf{u}) = (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T$, and $\mathbf{g}(\mathbf{u}) = (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T$. Here ρ is the density, (u, v) is the velocity, E is the total energy and $p = (\gamma - 1)(E - \frac{1}{2}(\rho u^2 + \rho v^2))$ is the pressure. γ is the gas constant which is taken as 1.4 in our numerical tests.

The initial condition is taken to be

$$(\rho, u, v, p) = (1.69997, 2.61934, -0.50632, 1.52819) \quad \text{on } y = 1,$$

$$(\rho, u, v, p) = \left(1, 2.9, 0, \frac{1}{\gamma}\right) \quad \text{otherwise.}$$

The boundary conditions are given by

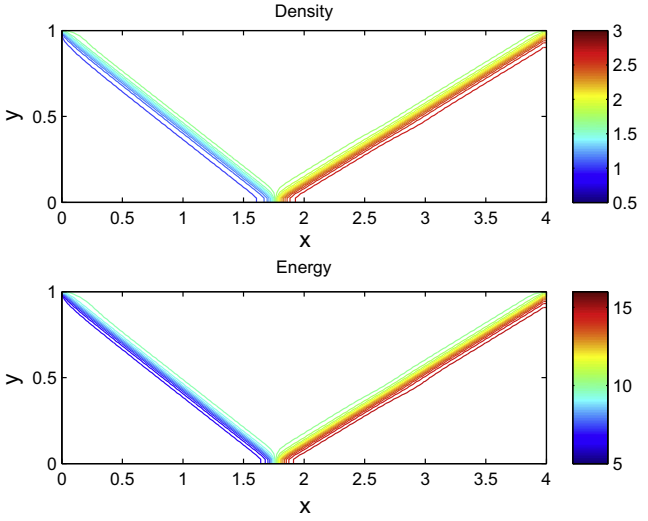

Fig. 13. Shock reflection simulated on a 160×40 uniform mesh. Top: 23 equally spaced contours from 0.94 to 2.72 for the density; bottom: 25 equally spaced contours from 5 to 15.2 for the energy. Total iteration number is 508.

$$(\rho, u, v, p) = (1.69997, 2.61934, -0.50632, 1.52819) \quad \text{on } y = 1,$$

and reflective boundary condition on $y = 0$. The left boundary at $x = 0$ is set as an inflow with $(\rho, u, v, p) = (1, 2.9, 0, \frac{1}{\gamma})$, and the right boundary at $x = 4$ is set to be an outflow with no boundary conditions prescribed. The numerical results are shown in Fig. 13. It can be clearly seen that the incident and reflected shocks are well-resolved. The L^1 residue, however, stagnates at 10^{-5} for density and 10^{-4} for energy.

6. Conclusion

In this paper, we proposed the first fast sweeping method for solving steady state hyperbolic equations with source terms. The method is based on the Lax–Friedrichs fluxes with WENO (Weighted Essentially Non-oscillatory) reconstruction to achieve high order accuracy. The alternating sweeping directions with Gauss–Seidel iterative updates are used to accelerate the speed of convergence. Furthermore, the modified stopping criteria is proposed to stop the algorithm when the residue of the scheme is sufficiently small but not monotonically decreasing.

We applied this proposed method to both scalar and system test problems including Burgers' equation, shallow water equations, nozzle flow problem and Euler equations. In all simulations, we observed that the fast sweeping LF-WENO3 method converges faster than the time evolution finite difference WENO3-RK4 methods: fast sweeping method needs fewer iterations to reach the same L_1 error of the solution. The overall computational performance of the fast sweeping method is more efficient than time evolution methods. Future work includes developing multigrid type high-order schemes based on this LF-WENO3 and other types of fast sweeping or fast marching methods based on upwind fluxes.

Acknowledgement

We thank reviewers for their constructive comments and Professor Chi-Wang Shu for the valuable discussion.

References

[1] R. Abgrall, Toward the ultimate conservative scheme: following the quest, J. Comput. Phys. 167 (2001) 277–315.
[2] R. Abgrall, T. Barth, Residual distribution schemes for conservation laws via adaptive quadrature, SIAM J. Scient. Comput. 24 (2002) 732–769.
[3] R. Abgrall, M. Mezine, Construction of second order accurate monotone and stable residual distribution schemes for steady problems, J. Comput. Phys. 195 (2004) 474–507.
[4] R. Abgrall, P.L. Roe, High order fluctuation scheme on triangular meshes, J. Scient. Comput. 19 (2003) 3–36.
[5] D.S. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, J. Comput. Phys. 160 (2000) 405–452.
[6] W. Cai, D. Gottlieb, C.-W. Shu, Essentially nonoscillatory spectral Fourier methods for shock wave calculations, Math. Comput. 52 (1989) 389–410.
[7] C. Chapman, Fundamentals of Seismic Wave Propagation, Cambridge University Press, 2010.
[8] C.-S. Chou, C.-W. Shu, High order residual distribution conservative finite difference WENO schemes for steady state problems on non-smooth meshes, J. Comput. Phys. 214 (2006) 698–724.
[9] B. Cockburn, G. Karniadakis, C.W. Shu, Discontinuous Galerkin Methods: Theory, Computation and Applications, Lecture Notes in Computational Science and Engineering, Springer, 2000.
[10] M.G. Crandall, P.L. Lions, Viscosity solutions of Hamilton–Jacobi equations, Trans. Amer. Math. Soc. 277 (1) (1983) 1–42.
[11] M.G. Crandall, P.L. Lions, Two approximations of solutions of Hamilton–Jacobi equations, Math. Comput. 43 (1984) 1–19.
[12] H. Deconinck, R. Struijs, G. Bourgeois, P, Roe, Compact advection schemes on unstructured meshes computational fluid dynamics, VKI Lecture Series, 1993.
[13] P. Embid, J. Goodman, A. Majda, Multiple steady states for 1-d transonic flow, SIAM J. Scient. Stat. Comput. 5 (1984) 21–41.
[14] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[15] C.-Y. Kao, S. Osher, J. Qian, Lax–Friedrichs sweeping scheme for static Hamilton–Jacobi equations, J. Comput. Phys. 196 (1) (2004) 367–391.
[16] C.-Y. Kao, S. Osher, J. Qian, Legendre-transform-based fast sweeping methods for static Hamilton–Jacobi equations on triangulated meshes, J. Comput. Phys. 227 (24) (2008) 10209–10225.
[17] C.-Y. Kao, S. Osher, Y.-H. Tsai, Fast sweeping methods for static Hamilton–Jacobi equations, SIAM J. Numer. Anal. 42 (6) (2005) 2612–2632.
[18] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, 2002.
[19] H.W. Liepmann, A. Roshko, Elements of Gas Dynamics, Wiley, New York, 1957.
[20] T-P. Liu, Hyperbolic and viscous conservation laws. SIAM, 2000.
[21] X.-D. Liu, S. Osher, T. Chan, Weighted essentially nonoscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[22] S. Osher, A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations, SIAM J. Math. Anal. 24 (1993) 1145–1152.
[23] S. Osher, B. Merriman, The Wulff shape as the asymptotic limit of a growing crystalline interface, Asian J. Math. 1 (3) (1997) 560–571.
[24] S. Osher, N. Paragios, Geometric Level Set Methods in Imaging, Vision, and Graphics, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
[25] D. Peng, S. Osher, B. Merriman, H.-K. Zhao, The geometry of Wulff crystal shapes and its relations with Riemann problems, in: Nonlinear Partial Differential Equations (Evanston, IL, 1998), Amer. Math. Soc., Providence, RI, 1999, pp. 251–303.
[26] J. Qian, Y.-T. Zhang, H.-K. Zhao, A fast sweeping method for static convex Hamilton–Jacobi equations, J. Scient. Comput. 31 (1–2) (2007) 237–271.
[27] P. Roe, Upwind differencing schemes for hyperbolic conservation laws with source terms, in: Claude Carasso, Denis Serre, Pierre-Arnaud Raviart (Eds.), Nonlinear Hyperbolic Problems, vol. 1270 of Lecture Notes in Mathematics, pp. 41–51. Springer, Berlin/Heidelberg, 1987. 10.1007/BFb0078316.
[28] P.L. Roe, Approximate Riemann solvers parameter vectors and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[29] P.L. Roe, D. Sidilkover, Optimum positive linear schemes for advection in two or three dimensions, SIAM J. Numer. Anal. 29 (1992) 1542–1588.
[30] M.D. Salas, S. Abarbanel, D. Gottlieb, Multiple steady states for characteristic initial value problems, Appl. Numer. Math. 2 (1986) 193–210.
[31] S. Serna, J. Qian, A stopping criterion for higher-order sweeping schemes for static Hamilton–Jacobi equations, J. Comput. Math. 28 (2010) 552–568.
[32] J.A. Sethian, Level set methods and fast marching methods, 2nd ed., Cambridge University Press, Cambridge, 1999, Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.
[33] J.A. Sethian, D. Adalsteinsson, An overview of level set methods for etching deposition and lithography development, IEEE Trans. Semicond. Dev. 10 (1996) 167–184.

[34] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: A. Quarteroni (Ed.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, vol. 1697, Springer, 1998, pp. 325–432.
[35] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[36] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. 83 (1989) 32–78.
[37] R. Struijs, H. Deconinck, P.L. Roe, Fluctuation splitting schemes for the 2D euler equations, computational fluid dynamics, VKI Lecture Series, 1991, pp. 1991–2001.
[38] J.A. Trangenstein, Numerical Solution of Hyperbolic Partial Differential Equations, Cambridge University Press, 2009.
[39] Y.-H.R. Tsai, L.-T. Cheng, S. Osher, H.-K. Zhao, Fast sweeping algorithms for a class of Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (2003) 673–694.
[40] W.G. Vincenti, C.H. Kruger, Introduction to Physical Gas Dynamics, Wiley, New York, 1965.
[41] P. Wesseling, Principles of Computational Fluid Dynamics, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.
[42] G. Whitham, Linear and Nonlinear Waves, Wiley-Interscience, New York, 1974.
[43] Y. Xing, C.-W. Shu, High-order well-balanced finite difference WENO schemes for a class of hyperbolic systems with source terms, J. Scient. Comput. 27 (2006) 477–494.
[44] Nail K. Yamaleev, Mark H. Carpenter, Third-order energy stable WENO scheme, J. Comput. Phys. 228 (8) (2009) 3025–3047.
[45] Y.-T. Zhang, H.-K. Zhao, J. Qian, High order fast sweeping methods for static Hamilton–Jacobi equations, J. Scient. Comput. 29 (1) (2006) 25–56.
[46] H. Zhao, A fast sweeping method for Eikonal equations, Math. Comput. 74 (250) (2004) 603–628.