J Sci Comput
DOI 10.1007/s10915-015-0006-7

Lax–Friedrichs Multigrid Fast Sweeping Methods for Steady State Problems for Hyperbolic Conservation Laws

Weitao Chen[1] · Ching-Shan Chou[2] · Chiu-Yen Kao[3]

Received: 31 December 2013 / Revised: 27 February 2015 / Accepted: 5 March 2015

Abstract Fast sweeping methods are efficient Gauss–Seidel iterative numerical schemes originally designed for solving static Hamilton–Jacobi equations. Recently, these methods have been applied to solve hyperbolic conservation laws with source terms. In this paper, we propose Lax–Friedrichs fast sweeping multigrid methods which allow even more efficient calculations of viscosity solutions of stationary hyperbolic problems. Due to the choice of Lax–Friedrichs numerical fluxes, general problems can be solved without difficult inversion. High order discretization, e.g., WENO finite difference method, can be incorporated to achieve high order accuracy. On the other hand, multigrid methods, which have been widely used to solve elliptic equations, can speed up the computation by smoothing errors of low frequencies on coarse meshes. We modify the classical multigrid method with regard to properties of viscous solutions to hyperbolic conservation equations by introducing WENO interpolation between levels of mesh grids. 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 singularities of the viscosity solutions.

Keywords Steady state · Hyperbolic conservation laws · Fast sweeping method · WENO · Multigrid method

C.-S. Chou: This author is supported by NSF DMS1253481.
C.-Y. Kao: This author is partially supported by NSF DMS1318364.

✉ Ching-Shan Chou
chou@math.ohio-state.edu

Weitao Chen
weitaoc1@uci.edu

Chiu-Yen Kao
Ckao@cmc.edu

[1] Department of Mathematics, University of California, Irvine, CA 92617, USA

[2] Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA

[3] Department of Mathematical Sciences, Claremont McKenna College, Claremont, CA 91711, USA

Published online: 18 March 2015

1 Introduction

Hyperbolic conservation laws, used to describe the conservation of quantities via first order nonlinear partial differential equations, arise in many applications including gas dynamics [20,32], shallow water waves [34], magneto-hydrodynamics [11], oil recovery [6], etc.

The solutions of these equations may develop singularities, such as discontinuities, and do not satisfy the equations in the classical sense. To find the physical relevant solutions, "vanishing viscosity solutions" and "entropy solutions" are introduced to define the solutions uniquely (see [19,21] and references therein). The main challenge in computing these vanishing viscosity solutions is that naive numerical discretization usually fails to capture solutions due to the lack of certain properties, e.g., domains of dependence and influence, conservation property, discontinuities, positivity preserving in solutions, or entropy conditions. Thus, development of successful numerical methods relies on a good understanding of the physical background and sophisticated approximations with built-in intrinsic properties, such as the use of consistent, conservative and dissipative numerical fluxes [12,17].

In this paper, we focus on solving the stationary problems of hyperbolic conservation laws with source terms,

$$\nabla \cdot f(u) = s(u), \quad u \in R^n,$$

in which the Jacobian matrix $f'(u)$ is diagonalizable with all the eigenvalues being real for any u. Most of the previous approaches are based on the pseudo time stepping, e.g., a class of schemes called "residual distribution schemes" [1–4,8,9,25,30] proposed to distribute the residuals, which are defined through integrating the flux and source terms on triangular or quadrilinear cells in a conservative fashion, and march in pseudo time. Recently, inspired by fast sweeping methods for time independent Hamilton–Jacobi equations, Chen et al. [7] proposed Lax–Friedrichs fast sweeping methods which discretize the steady state hyperbolic conservation laws directly, by approximating the spatial derivatives with consistent and conservative numerical fluxes, and iterating with Gauss–Seidel type nonlinear method with a finite number of alternating sweeping directions. In particular, the Lax–Friedrichs fluxes evaluated in WENO (Weighted Essentially Non-oscillatory) fashion [15,22,26–28] are used to achieve high order accuracy as well as high resolution of shocks. Besides Lax–Friedrichs fast sweeping scheme, another type of fast sweeping methods which relies on a paraxial form of the equations was proposed recently in [31]. The computational cost is very low and the method resolves shocks sharply by directly imposing the Rankine–Hugoniot jump conditions, together with appropriate entropy conditions. For complex interaction of shock waves and rarefaction waves in high dimensions, the implementation is more complicated and the scheme may require more iterations to converge.

Here we aim to base our study upon the Lax–Friedrichs fast sweeping methods proposed in [7] and improve the efficiency by integrating multigrid (MG) acceleration. It is known that traditional MG methods for solving elliptic equations cannot achieve the same efficiency when they are applied directly to solve steady state solutions of hyperbolic equations, especially in system problems. Previous approaches include, e.g., dimensional splitting [29], splitting of elliptic and advection parts of the operators [23], flux splitting with upwind fluxes [13,14] and upwind interpolation and restriction techniques [5,16,18,33]. Steady states of hyperbolic systems are directly solved by the MG procedure with upwind fluxes. Previous methods are usually limited by the complexity in the flux construction, thus only have first or second order accuracy. The challenge and novelty of our approach is to embed the high order Lax–Friedrichs WENO fluxes in MG framework and obtain the viscosity solutions with less computational cost. Due to the presence of discontinuous singularities, regular interpolation

may lose efficiency, sometimes even fail to converge, compared with Lax–Friedrichs fast sweeping method. In [5], the authors proposed a Riemann solver to replace the regular interpolation, which is still computationally expensive because it requires to solve for the steady state of another hyperbolic equation corresponding to the solution on the coarse mesh. Inspired by the upwind-biased WENO construction described in [35], we directly interpolate the error between coarse and fine meshes by WENO interpolation, which can deal with singularities and be straightforwardly extended to system cases after characteristic splitting. Based on the construction of our approach, it is flexible to adopt high order accurate numerical fluxes and improve the efficiency by the MG procedure.

The rest of the paper is organized as follows. First, we begin in Sect. 2 with a brief review of Lax–Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws with source terms described in [7]. The MG full approximation scheme is then described in Sect. 3, where we also explain upwind-biased WENO interpolation to avoid slow convergence. Numerical results obtained from our algorithm are included in Sect. 4, demonstrating the improved efficiency compared to the previous Lax–Friedrichs fast sweeping methods without MG acceleration. Conclusive remarks are given in Sect. 5.

2 Review of Lax–Friedrichs Fast Sweeping Method for Steady State Hyperbolic Conservation Problems

In this section, we review the iterative method for hyperbolic problems that was developed in our previous work [7].

2.1 High Order Lax–Friedrichs WENO Sweeping Method for 1D Problems

We first 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.

We discretize the interval 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 point x_j is 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 $f_x(u_j)$.

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

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

where $\sigma = \max_j\{|f'(u_j)|\}$, $j = 1, \ldots, N-1$, one obtains 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

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

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

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

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. This scheme is first order accurate and serves as a framework for the high order sweeping method.

To achieve high order accuracy, high order numerical fluxes $\hat{f}_{j\pm\frac{1}{2}}$ need to be used. Here, we adopt WENO numerical flux reconstruction [7,15,22,26–28] to obtain Lax–Friedrichs WENO fluxes, $\hat{f}_{j\pm\frac{1}{2}}$.

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 WENO 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

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

Thus, we obtain the iterative scheme

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

As in the first order sweeping method, the iterations will take alternating directions: if we sweep from left to right, then in (6), $u_{j-1} = u_{j-1}^{n+1}$ 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.

Similar approaches can be applied to 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 (6), 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 [24]. 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 in

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

where $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$ is the matrix whose columns are the right eigenvectors of $\mathbf{f}'(\mathbf{u}_{j+\frac{1}{2}})$, $\pm$ indicates fluxes with opposite characteristic directions after Lax–Friedrichs splitting. The numerical fluxes $\hat{\mathbf{v}}^{\pm}_{j+\frac{1}{2}}$ 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 (6).

2.2 High Order Lax–Friedrichs WENO Sweeping Method for 2D Problems

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

$$f(u)_x + g(u)_y = s(u, x, y), \quad (x, y) \in [a, b] \times [c, d]. \tag{7}$$

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 $y-$ directions, 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 (7) can be written as

$$\frac{\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}}{\Delta x} + \frac{\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}}{\Delta y} = s(u_{i,j}, x_i, y_j). \tag{8}$$

If one uses the high order Lax–Friedrichs WENO fluxes constructed in Eq. (5), then applying

$$\hat{f}_{i+\frac{1}{2},j} = \hat{\hat{f}}_{i+\frac{1}{2},j} - \frac{\sigma_x}{2}\left(u_{i+1,j} - u_{i,j}\right),$$
$$\hat{g}_{i,j+\frac{1}{2}} = \hat{\hat{g}}_{i,j+\frac{1}{2}} - \frac{\sigma_y}{2}\left(u_{i,j+1} - u_{i,j}\right),$$

in (8), where $\sigma_x = \max_{i,j}\{|f'(u_{i,j})|\}$ and $\sigma_y = \max_{i,j}\{|g'(u_{i,j})|\}$, leads to the equation

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

Thus $u_{i,j}$ can be solved in terms of neighboring grid function values

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

We sweep the whole domain with four alternating orderings repeatedly,

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

where I and J are maximum values for indices in x- and y- directions, respectively. If the first sweeping direction is chosen, then $u_{i-1,j} = u^{n+1}_{i-1,j}$, $u_{i,j-1} = u^{n+1}_{i,j-1}$, $u_{i+1,j} = u^n_{i+1,j}$ and $u_{i,j+1} = u^n_{i,j+1}$. The updating formula is

$$u^{n+1}_{i,j} = \frac{\sigma_x \Delta y \left(u^n_{i+1,j} + u^{n+1}_{i-1,j}\right)}{2(\sigma_x \Delta y + \sigma_y \Delta x)} + \frac{\sigma_y \Delta x \left(u^n_{i,j+1} + u^{n+1}_{i,j-1}\right)}{2(\sigma_x \Delta y + \sigma_y \Delta x)} + \frac{\Delta x \Delta y \, s(u^n_{i,j}, x_i, y_j)}{\sigma_x \Delta y + \sigma_y \Delta x} - \frac{\Delta y \left(\hat{\hat{f}}_{i+\frac{1}{2},j} - \hat{\hat{f}}_{i-\frac{1}{2},j}\right) + \Delta x \left(\hat{\hat{g}}_{i,j+\frac{1}{2}} - \hat{\hat{g}}_{i,j-\frac{1}{2}}\right)}{\sigma_x \Delta y + \sigma_y \Delta x}. \tag{9}$$

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

$$u^{n+1}_{i,j} = \frac{\sigma_x \Delta y \left(u^{n+1}_{i+1,j} + u^n_{i-1,j}\right)}{2(\sigma_x \Delta y + \sigma_y \Delta x)} + \frac{\sigma_y \Delta x \left(u^n_{i,j+1} + u^{n+1}_{i,j-1}\right)}{2(\sigma_x \Delta y + \sigma_y \Delta x)} + \frac{\Delta x \Delta y \, s(u^n_{i,j}, x_i, y_j)}{\sigma_x \Delta y + \sigma_y \Delta x} - \frac{\Delta y \left(\hat{\hat{f}}_{i+\frac{1}{2},j} - \hat{\hat{f}}_{i-\frac{1}{2},j}\right) + \Delta x \left(\hat{\hat{g}}_{i,j+\frac{1}{2}} - \hat{\hat{g}}_{i,j-\frac{1}{2}}\right)}{\sigma_x \Delta y + \sigma_y \Delta x}. \tag{10}$$

Notice that the indices n and $n+1$ are switched for stencils involving $u_{i-1,j}$ and $u_{i+1,j}$ in (9) and (10) because the sweeping directions 1 and 2 visit these two points in reverse orders. Similarly, the updating formulas in the third and fourth sweeping directions require the switched indices n and $n+1$ for stencils involving $u_{i,j-1}$ and $u_{i,j+1}$.

For two-dimensional system problems

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

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 $L_x \Lambda_x R_x$ and $L_y \Lambda_y R_y$, 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 one-dimensional systems, 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 [24]. 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 in the construction of fluxes:

$$\Phi_p^{\pm} = \bar{\mathbf{R}}_x{}^{-1}\mathbf{f}_p^{\pm}, \quad \text{for } p \text{ in a neighborhood of } i,$$
$$\Psi_q^{\pm} = \bar{\mathbf{R}}_y{}^{-1}\mathbf{g}_q^{\pm}, \quad \text{for } q \text{ in a neighborhood of } j,$$

where $f_p^{\pm}$, $g_q^{\pm}$ are fluxes corresponding to Lax–Friedrichs splitting applied on each dimension. 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 $\bar{\mathbf{R}}_x$ and $\bar{\mathbf{R}}_y$, 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.

3 Fast Sweeping Method with Multigrid Fully Approximation Scheme

Although it has been demonstrated in our previous work [7] that Lax–Friedrichs fast sweeping method is very efficient, compared to the regular time evolution schemes, to converge to steady states, we attempt to further accelerate the convergence rate of the fast sweeping method by coupling our iterative scheme with the well-known MG framework, presented as follows.

3.1 Framework

Consider the one-dimensional scalar steady state problem

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

with an initial guess and appropriate boundary conditions. Based upon the finite difference discretization in space and the approximation of the derivative by Lax–Friedrichs WENO fluxes, it can be written as a nonlinear problem

$$A(U) = s(U, X), \tag{11}$$

where $X = [x_0, x_1, \ldots, x_N]^T$ is the vector of grid points, $U = [u_0, u_1, \ldots, u_N]^T$ is the vector of discretized u, and $A(U)$ is a nonlinear operator such that

$$A(U)_j = \frac{\hat{\hat{f}}_{j+\frac{1}{2}} - \frac{\sigma}{2}(u_{j+1} - u_j) - \hat{\hat{f}}_{j-\frac{1}{2}} + \frac{\sigma}{2}(u_j - u_{j-1})}{\Delta x},$$

with $\hat{\hat{f}}$ constructed in Eq. (5). To solve this nonlinear problem, Eq. (11), with MG solver, one typically uses the full approximation scheme, consisting of fine grid smoother, restriction and interpolation between fine and coarse meshes. In our algorithm, we will use Lax–Friedrichs fast sweeping iteration as the smoother, which is more efficient than time marching as demonstrated in [7]. We denote the restriction operator by R and the interpolation operator by P, then the two-level MG fast sweeping scheme in one-dimensional space can be summarized as follows:

1. **Setting up two levels of grids:** Divide the domain $[a, b]$ into $N = 2^k$ equal cells for some integer k. The fine mesh grid points are $x_j = a + jh$ for $j = 0, 1, \ldots, N$, where $h = \frac{b-a}{N}$. Let $u^h = [u_0^h, \ldots, u_N^h]^T$ be the discretized numerical solution on the fine mesh. Then the mesh size on the coarse mesh is $2h$ and the grid points are $x_i^{2h} = a + 2jh$ for $j = 0, \ldots, N/2$, which are identical to those even-indexed points on the fine mesh. Denote the numerical solution on the coarse mesh by u^{2h}.

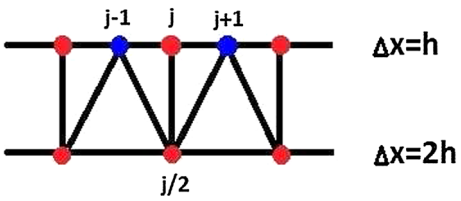

Fig. 1 Restriction between the fine mesh and the coarse mesh

2. **Smoothing on the fine grid:** Let $u^h = u^n$ be the numerical solution on the fine mesh after n two grid iterations. We update u^h by Lax–Friedrichs fast sweeping method shown in Eq. (6). The coefficient σ is only updated on the finest mesh and will be passed down to the coarse mesh. The number of iterations on the finest mesh is problem dependent and chosen to be the smallest value such that our MG method converges. We will specify this number in each example in Sect. 4.

3. **Residual restriction:** Compute the residual on the fine mesh

$$r^h(j) = \frac{\hat{\hat{f}}_{j+\frac{1}{2}} - \frac{\sigma}{2}\left(u^h_{j+1} - u^h_j\right) - \hat{\hat{f}}_{j-\frac{1}{2}} + \frac{\sigma}{2}\left(u^h_j - u^h_{j-1}\right)}{h} - s(x_j, u^h_j),$$

and restrict the residual to the coarse mesh by the restriction operator R. A typical choice of R involves all neighboring grid points with some fixed weights. For example, in one-dimensional space, $r^{2h}_{j/2} = \frac{1}{4}\left(r^h_{j-1} + 2r^h_j + r^h_{j+1}\right)$, as shown in Fig. 1. This is what we use in our scheme.

4. **Smoothing on coarse grid:** The system we need to solve on the coarse mesh is

$$A(u^{2h}) = A(I^{2h}_h u^h) + R(r^h),$$

where I^{2h}_h is the identical injection between different mesh grids. Apply the same Lax–Friedrichs fast sweeping iteration once on coarse mesh as Eq. (6) after replacing $s(u, x)$ by $A(I^{2h}_h u^h) + R(r^h)$ and update u^{2h}.

5. **Interpolating the error and correcting the solution on the fine mesh:** The error on the coarse mesh is

$$e^{2h} = u^{2h} - I^{2h}_h u^h.$$

Interpolate e^{2h} to the fine mesh and correct the solution u^h by

$$u^h \leftarrow u^h + P(e^{2h}).$$

For the interpolation operator P, the simplest way is the linear interpolation, which projects the values from the coarse mesh to the fine mesh directly at even-indexed fine-grid points and takes the average of the adjacent coarse-grid values at odd-indexed fine-grid points. However, since shocks or rarefaction waves may occur in our solutions of hyperbolic equations, we need to construct the interpolation operator P more carefully and the details are provided in Sect. 3.2.

The V-cycle MG fast sweeping method we use in the numerical tests is obtained by applying the two-level MG fast sweeping iteration recursively.

MG fast sweeping method for system problems follows the same mechanism as the scalar problems except that characteristic field splitting is implemented in the flux construction

and interpolation. In higher dimensional space, WENO fluxes are computed dimension by dimension, so is the interpolation.

3.2 Upwind-Biased WENO Interpolation in Scalar and System Problems

When the commonly used interpolation operators are applied in MG method to solve hyperbolic problems, it will converge at a slow rate, and in some cases it may not converge, especially for systems of equations. To avoid that, an upwind interpolation is proposed in [5]. The authors suggest a strategy which needs to solve a Riemann problem for the corrected solution on the fine mesh instead of interpolation. However, it requires a time evolution until the steady state is obtained, which is time-consuming. To deal with that, we will follow the upwind-biased WENO mechanism to construct the interpolation operator such that the information will propagate along the characteristic directions.

Let e^{2h} be the error on the coarse mesh. As we move up to the fine mesh, the errors on the fine mesh e^h_j, $j = 0, 2, 4, \ldots$ can be taken as $e^{2h}_{j/2}$, while e^h_j for $j = 1, 3, 5, \ldots$ need to be interpolated.

A first order upwind-biased WENO interpolation is simply the upwind interpolation. However, based on our experience, a high order WENO interpolation is needed to ensure the convergence of our MG method. The upwind-biased WENO3 interpolation is constructed as follows. To interpolate e^h_j, $j = 1, 3, 5, \ldots$, we follow the WENO procedure to construct $e^{\text{WENO3}}_{j^-}$ and $e^{\text{WENO3}}_{j^+}$, which are interpolations based on the left and right biased stencils respectively. The calculation of $e^{\text{WENO3}}_{j^-}$ is based on e^h_{j-3}, e^h_{j-1} and e^h_{j+1}, which are identical to $e^{2h}_{(j-3)/2}$, $e^{2h}_{(j-1)/2}$ and $e^{2h}_{(j+1)/2}$. The formula to compute $e^{\text{WENO3}}_{j^-}$ is

$$e^{\text{WENO3}}_{j^-} = w_0 \cdot \frac{e^h_{j-1} + e^h_{j+1}}{2} + w_1 \cdot \frac{3e^h_{j-1} - e^h_{j-3}}{2}, \tag{12}$$

where

$$w_0 = \frac{a_0}{a_0 + a_1}, \qquad w_1 = \frac{a_1}{a_0 + a_1},$$

$$a_0 = \frac{2/3}{(\epsilon + (e^h_{j+1} - e^h_{j-1})^2)^2}, \text{ and} \qquad a_1 = \frac{1/3}{(\epsilon + (e^h_{j-1} - e^h_{j-3})^2)^2}.$$

$e^{\text{WENO3}}_{j^+}$ is constructed similarly. We determine whether to use $e^{\text{WENO3}}_{j^-}$ or $e^{\text{WENO3}}_{j^+}$ by the characteristic direction calculated from $f'(u^{2h})$, where u^{2h} is the numerical solution on the coarse mesh after smoothing. The details are described in Algorithm 1.

For systems, we apply WENO3 construction within each characteristic field. First, the error is projected into the characteristic fields of the solution u^{2h} by multiplying the corresponding left eigenvectors. Each component is calculated independently as in Algorithm 1 and Eq. (12) upon the biased stencils which are determined by the sign of the associated eigenvalues. Once each component is obtained, we project them back to the physical space by multiplying their right eigenvectors. Such interpolation can avoid interaction among different characteristic fields and ensure that the information propagates through the correct direction.

For higher dimensional problems, this WENO interpolation is carried out dimension by dimension. Taking a two-dimensional problem as an example, there are two types of stencils where the values must be interpolated from their neighboring points as shown in Fig. 2. For blue points, e^{2h} is available either on x-dimension or y-dimension only. So we can implement the upwind-biased WENO3 interpolation in a single dimension to obtain the

Algorithm 1 Upwind-biased WENO3 interpolation

for $j = 1 : 2 : N - 1$ **do**
  if $f'(u^{2h}_{(j-1)/2}) < 0$ and $f'(u^{2h}_{(j+1)/2}) < 0$ **then** $e^h_j = e^{\text{WENO3}}_{j+}$;
  else if $f'(u^{2h}_{(j-1)/2}) > 0$ and $f'(u^{2h}_{(j+1)/2}) > 0$ **then** $e^h_j = e^{\text{WENO3}}_{j-}$;
  else if $f'(u^{2h}_{(j-1)/2}) \le 0$ and $f'(u^{2h}_{(j+1)/2}) \ge 0$ **then** $e^h_j = 0$;
  else
    if $\frac{f(u^{2h}_{(j+1)/2}) - f(u^{2h}_{(j-1)/2})}{u^{2h}_{(j+1)/2} - u^{2h}_{(j-1)/2}} > 0$ **then** $e^h_j = e^{\text{WENO3}}_{j-}$;
    else $e^h_j = e^{\text{WENO3}}_{j+}$;
    end if
  end if
end for

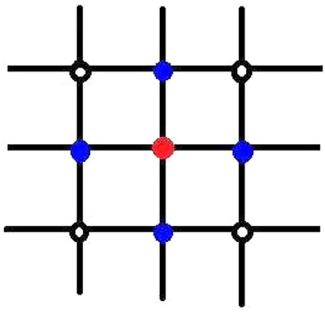

Fig. 2 Diagram of 2D interpolation from the coarse mesh to the fine mesh: *black circles* denote the coarse mesh, *blue* and *red* points are stencils that only belong to the fine mesh. (Color figure online)

numerical values on them. For red points, we simply use the average of the interpolated values on the blue points surrounding them. In some special cases, for example, the wave is propagating along either x or y direction, we need some other more effective techniques, e.g., semi-coarsening, in which the coarsening is only performed along the wave direction, so is the interpolation.

3.3 Numerical Boundary Condition for Problems with One-Side Boundary information

In hyperbolic problems, since solutions follow the characteristic directions, only the information on the boundaries with incoming characteristics is necessary. However, we still need numerical boundary conditions for boundaries with outgoing characteristics, and extrapolation is used in our calculations. In the fast sweeping iteration, a third order extrapolation is implemented on the finest mesh only to compute the values at the boundary without prescribed boundary condition. And the extrapolated boundary values will be stored and passed down to the other coarse meshes. The extrapolation is also needed for constructing WENO flux near the boundary, and for interpolation between coarse and fine meshes in the MG method.

4 Numerical Simulations for Hyperbolic Conservation Problems

In this section, we show the numerical results of the proposed Lax–Friedrichs fast sweeping MG method for hyperbolic steady state problems in one and two dimensions. The efficiency

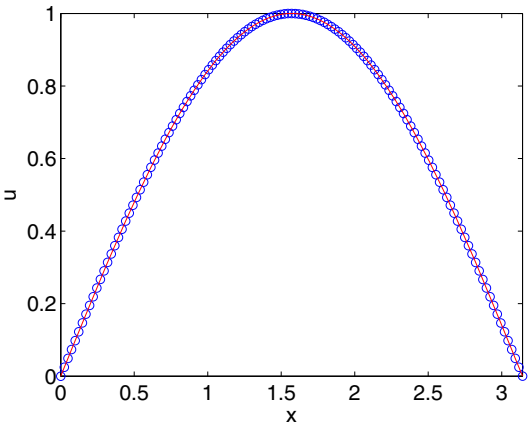

Fig. 3 Example 4.1.: the numerical solution (*blue circle*) by MG method with 128 cells on the finest mesh and the exact solution (*red solid line*) (Color figure online)

and high order accuracy of the proposed scheme are demonstrated by comparing it with the regular Lax–Friedrichs fast sweeping method on fixed meshes. In our settings, one sweep means visiting all grid points on the finest mesh once. The number of sweeps by MG method is calculated as the ratio of the total number of visits on any grid points through all levels of mesh grids and the number of grid points on the finest mesh. On a single level, we define one iteration by the total sweepings in all alternating directions. Without specification, the number of iterations on each level of grids is set to be 1, and a full V-cycle with coarsening until only two cells left within the domain is applied. As for the stopping criteria, let $res(i)$ be the residual of the steady state equation after i^{th} iteration, if the scheme converges, the simulations stop when $res(n) \leq 10^{-8} res(0)$, unless otherwise stated. All computations are carried out using MATLAB 2011b on a Macintosh computer with 2.6 GHz Intel Core i5 processor and 8 GB Memory.

4.1 The One-Dimensional Scalar Problems

We test our algorithm on scalar problems in one-dimensional space. In all examples, we use a uniform mesh with $N = 128$ on the finest mesh, unless specified otherwise.

Example 4.1 We solve the steady state solution of the one-dimensional Burgers' equation with a source term:

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

with the initial condition

$$u(x, 0) = 2 \sin x,$$

and boundary conditions $u(0, t) = u(\pi, t) = 0$. The steady state solution to this problem is smooth within the domain followed by a shock generated at the boundary and finally approaches $u(x, \infty) = \sin x$, which is smooth. We apply MG method on it to test the efficiency and the order of accuracy. Exact solutions are imposed on the ghost points in WENO3 flux construction. To ensure the convergence, we implement five iterations on the finest mesh of the MG method. The numerical solution is shown in Fig. 3.

We compare the fast sweeping MG method with Lax–Friedrichs fast sweeping method by plotting the residual of the steady state equation with respect to the number of sweeps under

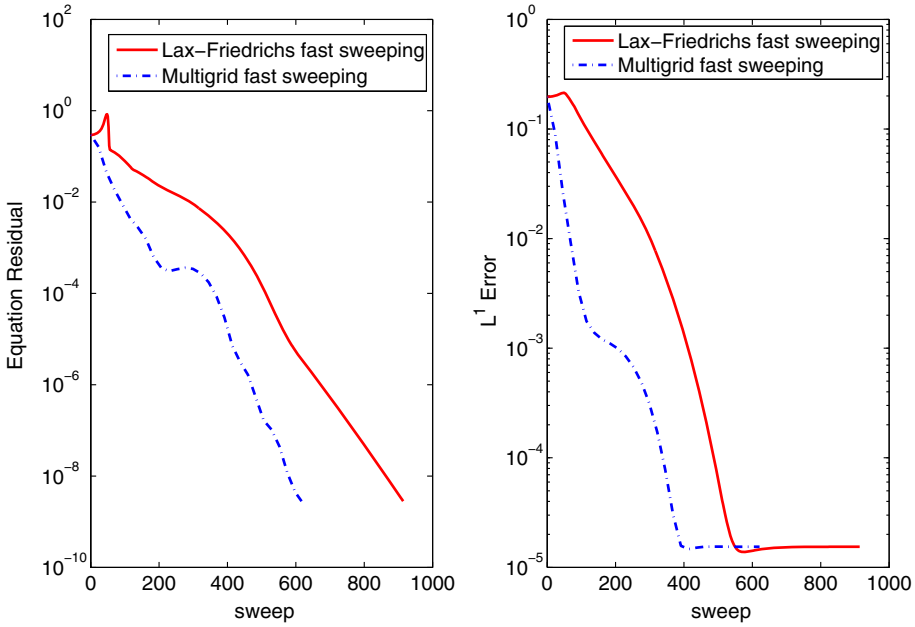

Fig. 4 Example 4.1: comparison of the equation residual (*left*) and L^1 error (*right*) by MG method (*blue dotted line*) and regular fast sweeping method (*red solid line*) with $N = 128$ (Color figure online)

Table 1 Comparison of sweeps and CPU time of the LF fast sweeping (LF) and MG framework for Example 4.1 on meshes with N cells

N	MG Sweeps	MG CPU (s)	LF Sweeps	LF CPU (s)
64	322	2.17	436	2.36
128	621	7.80	914	9.83
256	1081	27.19	1872	38.78
512	1518	75.53	3842	166.07
1024	3430	328.78	7920	696.97

the same stopping criteria. The comparison is displayed in Fig. 4. We can see that errors for both methods stagnate at the same level. The MG fast sweeping method saves about 1/3 of the sweeps to approach the exact solution.

We have tested the order of accuracy of the MG method with different mesh sizes on the finest mesh. The L^1 and L^∞ errors are calculated. The third order of accuracy is observed in L^1 error. The order of L^∞ error is not exactly 3, but this can be expected when third order WENO Lax–Friedrichs fluxes are used for this case with a shock formed at the boundary. The sweep numbers and the CPU time are provided in Table 1 to show the efficiency improved by the MG mechanism compared with Lax–Friedrichs fast sweeping method.

Example 4.2 We consider the steady state solutions of Burgers' equation with a source term depending on the solution itself:

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

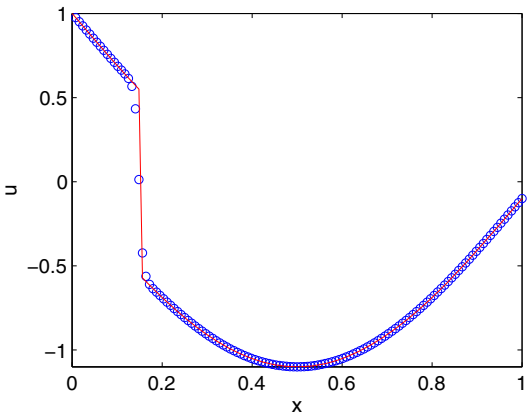

Fig. 5 Example 4.2.: the stable entropy solution (*red solid line*), and the numerical solution (*blue circles*) computed by MG method with 128 cells on the finest mesh (Color figure online)

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

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

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 [10] as an example of multiple steady states for one-dimensional transonic flows and had been tested to demonstrate that starting with a reasonable perturbation of the stable steady state, the fast sweeping method can capture the stable shock in [7].

The initial guess is given by:

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

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 and WENO interpolation. The number of iterations on the finest mesh is chosen to be 3. The numerical result with 128 cells and the exact solution are displayed in Fig. 5. We can see the well-captured shock location and good resolution of the shock.

To show the advantage of MG method, we compare the residuals of the steady state equation and L^1 errors by MG fast sweeping and regular fast sweeping, displayed in Fig. 6. We can observe that to achieve the same level of residual, our MG method saves more than half of the iterations compared with the LF fast sweeping method. Additionally, from the history of L^1 error, the solution obtained by MG method approaches the steady state within very few iterations. The CPU time cost for different N is shown in Table 2. We also show the number of sweeps by LF fast sweeping and MG method with different levels, denoted by L, on different mesh sizes in Table 3. As the number of levels used in MG increases, it saves more sweeps. Although the MG method is not optimal and the sweep number is mesh dependent, the computation cost increases less than the LF fast sweeping method.

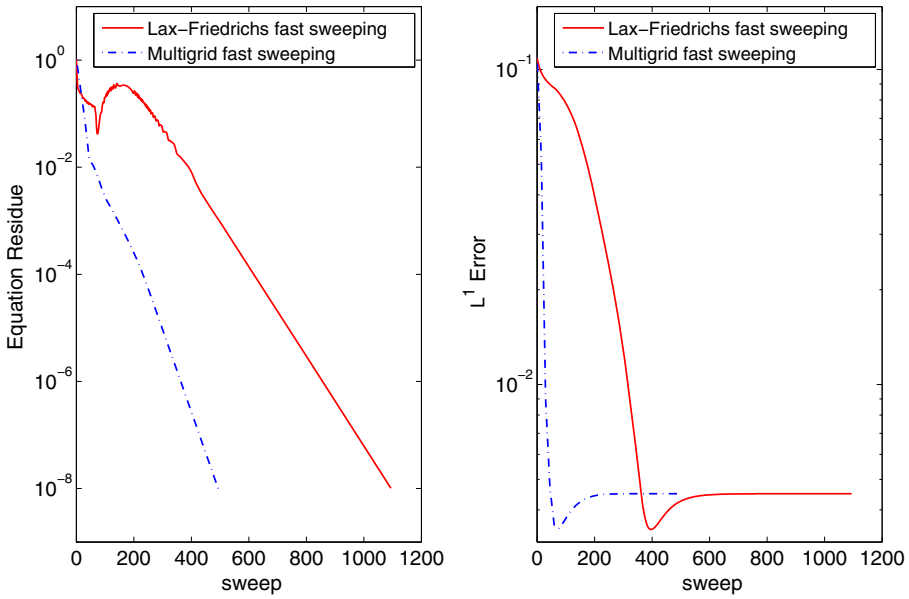

Fig. 6 Example 4.2.: comparison of the residual (*left*) and L^1 error (*right*) by MG method (*blue dotted line*) and regular fast sweeping method (*red solid line*) (Color figure online)

Table 2 Comparison of CPU time (s) of the LF fast sweeping method (LF) and MG framework for Example 4.2. on meshes with N cells

N	MG CPU (s)	LF CPU (s)
64	2.85	3.21
128	7.62	13.23
256	24.91	52.70
512	60.19	213.71

Table 3 Sweeps by LF fast sweeping and MG solver for Example 4.2. on meshes with N cells

	$N = 64$	$N = 128$	$N = 256$	$N = 512$
LF fast sweeping	540	1094	2194	4416
$L = 4$	375	645	1080	2115
$L = 5$	345	570	870	1545
$L = 6$	360	525	765	1170

Example 4.3 In this example, we study Burgers' equation with another source term on the right-hand side:

$$u_t + \left(\frac{u^2}{2}\right)_x = (6x - 3)u,$$

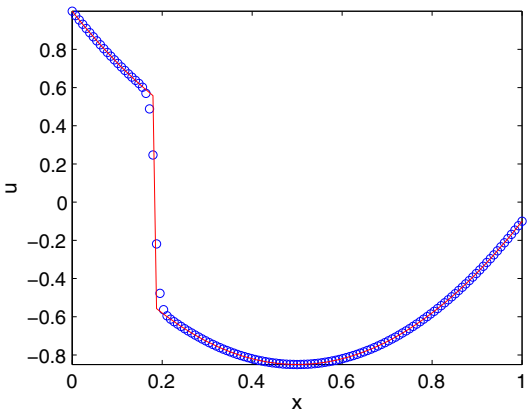

Fig. 7 Example 4.3.: the stable entropy solution (*red solid line*), and the numerical solution (*blue circles*) computed by MG method with 128 cells on the finest mesh (Color figure online)

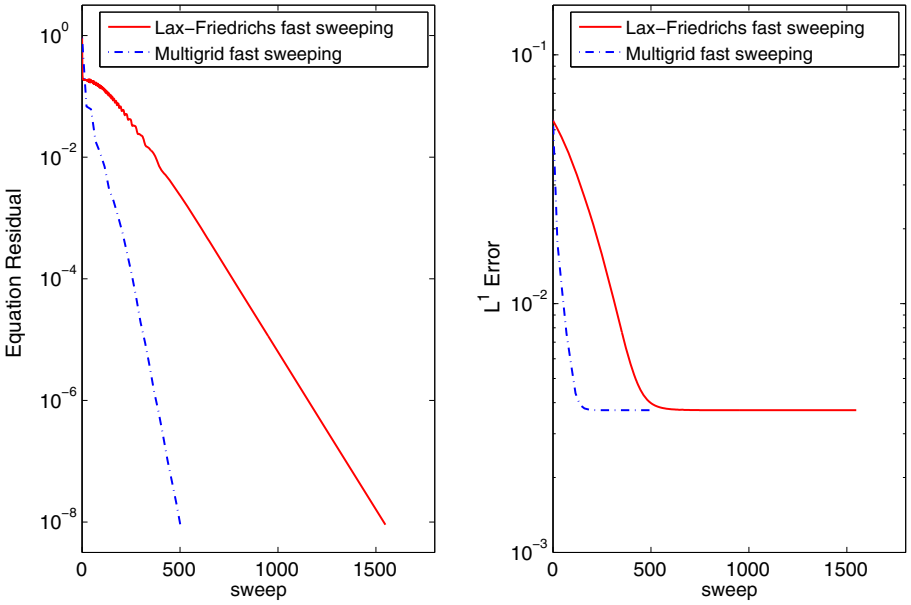

Fig. 8 Example 4.3.: comparison of the residual (*left*) and L^1 error (*right*) by MG method (*blue dotted line*) in 6.96 s and Lax–Friedrichs fast sweeping method (red solid line) in 18.41 s (Color figure online)

with boundary conditions $u(0) = 1$ and $u(1) = -0.1$. The steady state is

$$u = \begin{cases} 1 + 3x^2 - 3x, & \text{if } x < x_s, \\ -0.1 + 3x^2 - 3x, & \text{if } x > x_s, \end{cases}$$

where $x_s = 0.18377223398316$. We test our MG algorithm on this problem and compare the results with fast sweeping iterative scheme. The initial condition is obtained by shifting the shock location to $x = 0.5$. Exact solution is imposed at the boundary. The number of iterations on the finest mesh is set to be 5. The numerical solution is shown in Fig. 7. The comparison of the MG method and Lax–Friedrichs fast sweeping scheme is displayed in Fig. 8. The MG method converges about three times faster than the fast sweeping method, which also can be observed from CPU time cost.

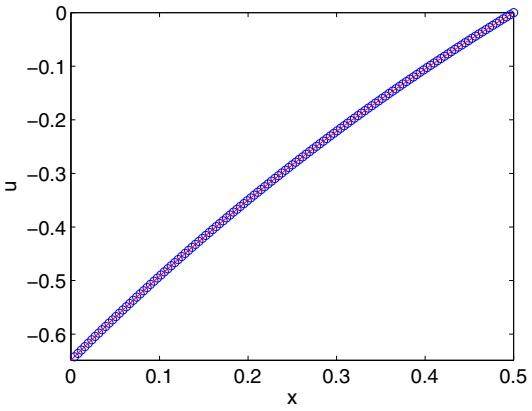

Fig. 9 Example 4.4.: the stable entropy solution (*red solid line*) and the numerical solution (*blue circles*) computed by MG method with 128 cells on the finest mesh (Color figure online)

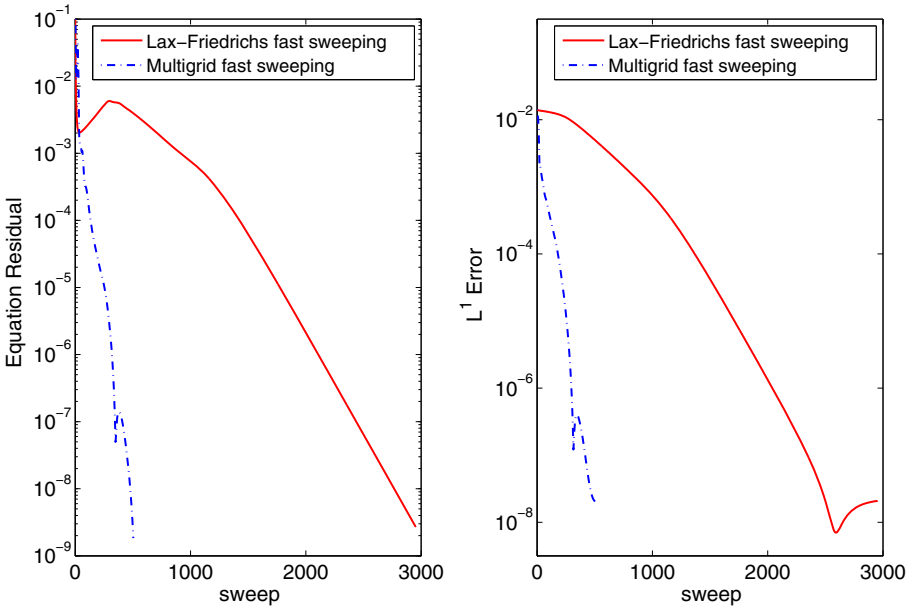

Fig. 10 Example 4.4.: comparison of the residual (*left*) and L^1 error (*right*) by MG method (*blue dotted line*) in 17.65 s and LF fast sweeping method (*red solid line*) in 57.22 s (Color figure online)

Example 4.4 This is another example involving Burgers' equation with a nonlinear source term which will reach a smooth stationary solution. The equation we study is

$$u_t + \left(\frac{u^2}{2}\right)_x = u(1-u), \quad x \in [0, 0.5],$$

with boundary conditions $u(0) = 1 - \sqrt{e}$ and $u(0.5) = 0$. The exact steady state is $u = 1 - e^{0.5-x}$. We choose a polynomial $u(x, 0) = -x(x - 0.5)(x - 1)$ as the initial guess and impose the exact solution at boundaries. The number of iterations on the finest mesh in MG V-cycle is chosen to be 1. The numerical solution is shown in Fig. 9 and the comparison of two methods is displayed in Fig. 10. The MG method only requires 1/5 of the sweeps

Table 4 Comparison of CPU time (s) of the LF fast sweeping method (LF) and MG framework for Example 4.4. on meshes with N cells

N	MG CPU (s)	LF CPU (s)
64	5.95	13.06
128	17.65	57.22
256	47.63	203.17
512	140.01	788.01

Table 5 Sweeps by LF fast sweeping and MG solver for Example 4.4. on meshes with N cells

	$N = 64$	$N = 128$	$N = 256$	$N = 512$
LF fast sweeping	2168	4268	8296	>10000
$L = 4$	602	973	1638	2891
$L = 5$	560	854	1316	2093
$L = 6$	553	826	1239	1918

of regular fast sweeping method to achieve the same residual. We have tested the L^1 and L^∞ errors associated with different N and a third order of accuracy is clearly achieved. The CPU time cost on different meshes is shown in Table 4. As N increases, the CPU time cost by MG method is almost 1/5 of that of LF fast sweeping method. The number of sweeps by LF fast sweeping and MG method with different levels, denoted by L, on different mesh sizes are shown in Table 5. As the number of levels used in MG increases, it saves more sweeps. Although the MG method is not optimal and the sweep number is mesh dependent, the increasing rate on the computation cost is less than that of LF fast sweeping method.

Example 4.5 Consider the steady state problem

$$(u^2)_x = u, \quad x \in [0, \frac{1}{4}],$$

with initial guess $u(x) = 0$. The exact solution will be $u(x) = 1 + \frac{x}{2}$ without singularities within the domain, and the wind direction is uniformly from left to right. We impose the exact boundary condition only at $x = 0$ and extend the solution at $x = \frac{1}{4}$ by the third order extrapolation. In MG method, when we interpolate the errors between two successive mesh grids by WENO scheme, zero boundary condition is imposed at $x = 0$ while extrapolation is used at $x = \frac{1}{4}$ corresponding to the one-sided boundary condition. The numerical solution by MG fast sweeping method is shown in Fig. 11, and the comparison of the efficiency between the MG method and Lax–Friedrichs fast sweeping method is shown in Fig. 12. It is clear to see that MG method saves at least half of the sweeps of regular fast sweeping method. Because its solution is a simple linear function, the numerical error is close to the machine error even on the coarse mesh and the order of accuracy is not shown in this case.

Remark 4.1 Based on the tests of one-dimensional scalar problems, we find that for problems with smooth stationary solutions without shocks (e.g., Examples 4.4 and 4.5), our MG method requires less iterations on the finest mesh to guarantee convergence compared with shock cases (e.g., Examples 4.1, 4.2 and 4.3). The reason may be that since the iterative scheme approaches the steady state very efficiently, the errors on the fine mesh also have discontinuities if the steady state has shocks. Therefore, MG method needs more iterations

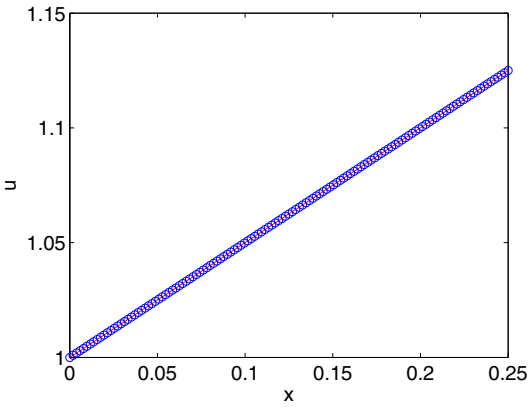

Fig. 11 Example 4.5.: the exact solution (*red solid line*) and the numerical solution (*blue circles*) computed by MG method with 128 cells on the finest mesh (Color figure online)

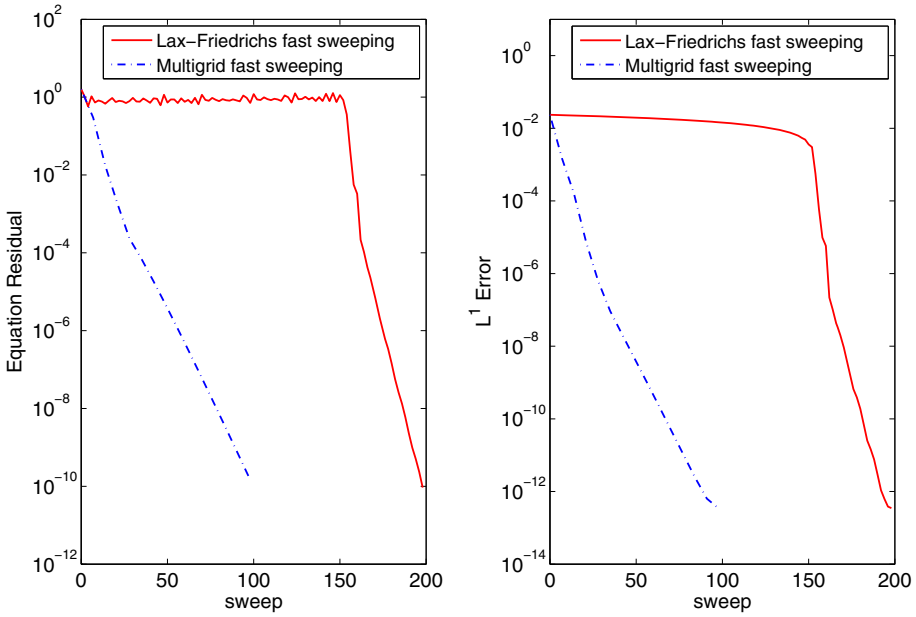

Fig. 12 Example 4.5.: comparison of the equation residual and the L^1 error history by LF fast sweeping (*red*) in 6.7 s and MG(*blue dashed*) in 4.82 s (Color figure online)

on the fine mesh to damp errors at high frequencies, especially around shock locations, so that a relative smooth error at low frequencies can be passed down to coarse meshes.

4.2 The One-Dimensional Systems

In this section, we test the MG method for hyperbolic conservation problems on one-dimensional systems of equations. The finest mesh is set to be $N = 256$. The stopping criteria is $res(n) \leq 10^{-8} res(0)$.

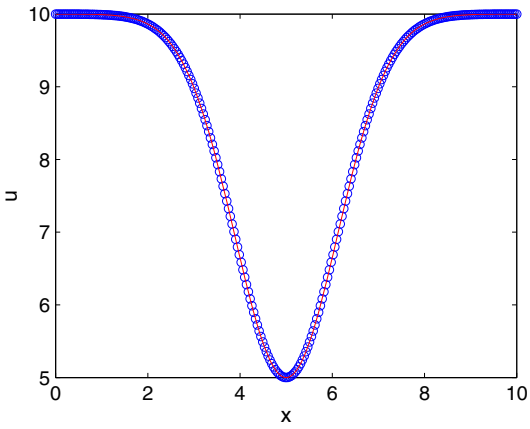

Fig. 13 Example 4.6.: the exact solution (*red solid line*), and the numerical solution (*blue circles*) computed by MG method with 256 cells on the finest mesh (Color figure online)

Example 4.6 We solve the steady state solutions of the one-dimensional shallow water equations:

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

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 will check the efficiency of our method. 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 solution of h is displayed in Fig. 13. In this example, we implement four times of fast sweeping iterations on the finest mesh to achieve convergence. The comparison of the residuals of the equation and L^1 errors by MG fast sweeping and Lax–Friedrichs fast sweeping are shown in Fig. 14. It can be clearly seen that the MG method is much more efficient.

We have tested the order of accuracy of the MG method with different mesh sizes. The third order of accuracy can be observed in L^1 and L^∞ error.

Example 4.7 This example is studied in [35] as a one-dimensional stationary shock of the Euler equations

$$\begin{pmatrix} \rho \\ \rho u \\ e \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(e + p) \end{pmatrix}_x = 0,$$

where ρ denotes the density, u is the velocity, e is the energy and p is the pressure related to e by $p = (\gamma - 1)(e - \frac{1}{2}\rho u^2)$ with $\gamma = 1.4$. The domain of x is $[-1, 1]$. A steady shock occurs at $x = 0$ such that

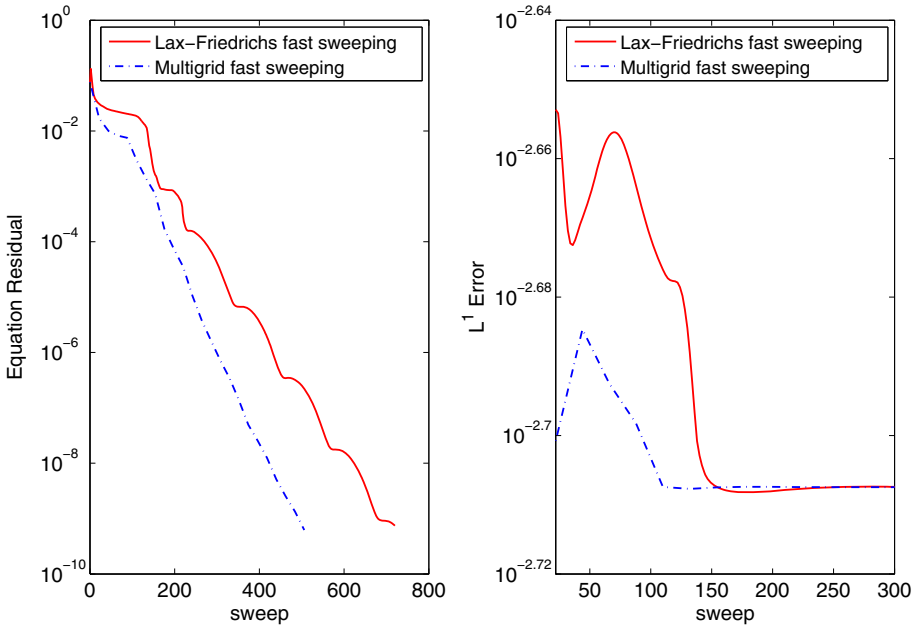

Fig. 14 Example 4.6.: comparison of the residual (*left*) and L^1 error (*right*) by MG method (*blue dotted line*) and Lax–Friedrichs fast sweeping method (*red solid line*). To reach the same stopping criterion, MG solver takes 45.07 s of CPU time while Lax–Friedrichs fast sweeping method takes 68.67 s (Color figure online)

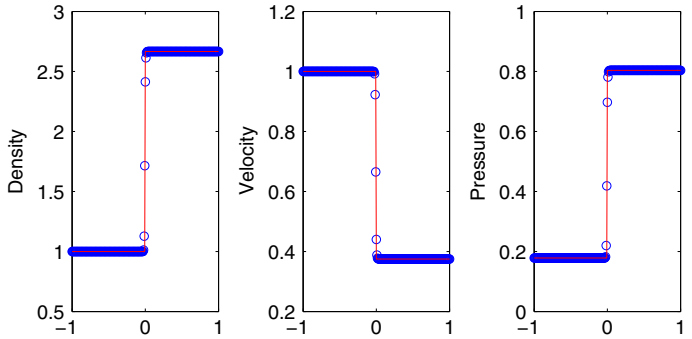

Fig. 15 Example 4.7.: the exact solution (*red solid line*) and numerical solution (*blue circles*) computed by MG method with 256 cells on the finest mesh (Color figure online)

$$\begin{pmatrix} p_l \\ \rho_l \\ u_l \end{pmatrix} = \begin{pmatrix} \frac{1}{\gamma M_\infty^2} \\ 1 \\ 1 \end{pmatrix} \quad \text{for } x < 0, \qquad \begin{pmatrix} p_r \\ \rho_r \\ u_r \end{pmatrix} = \begin{pmatrix} p_l \frac{2\gamma M_\infty^2 - (\gamma-1)}{\gamma+1} \\ \frac{\frac{\gamma+1}{\gamma-1}\frac{p_r}{p_l}+1}{\frac{\gamma+1}{\gamma-1}+\frac{p_r}{p_l}} \\ \sqrt{\gamma \frac{(2+(\gamma-1)M_\infty^2)p_r}{(2\gamma M_\infty^2+(1-\gamma))\rho_r}} \end{pmatrix} \quad \text{for } x > 0,$$

with Mach number $M_\infty = 2$. It's easy to check this shock solution satisfies Rankine–Hugoniot condition. We let the initial guess be the steady shock and test our MG method with five iterations on the finest mesh. The numerical solution and the comparison with Lax–Friedrichs fast sweeping method are shown in Figs. 15 and 16.

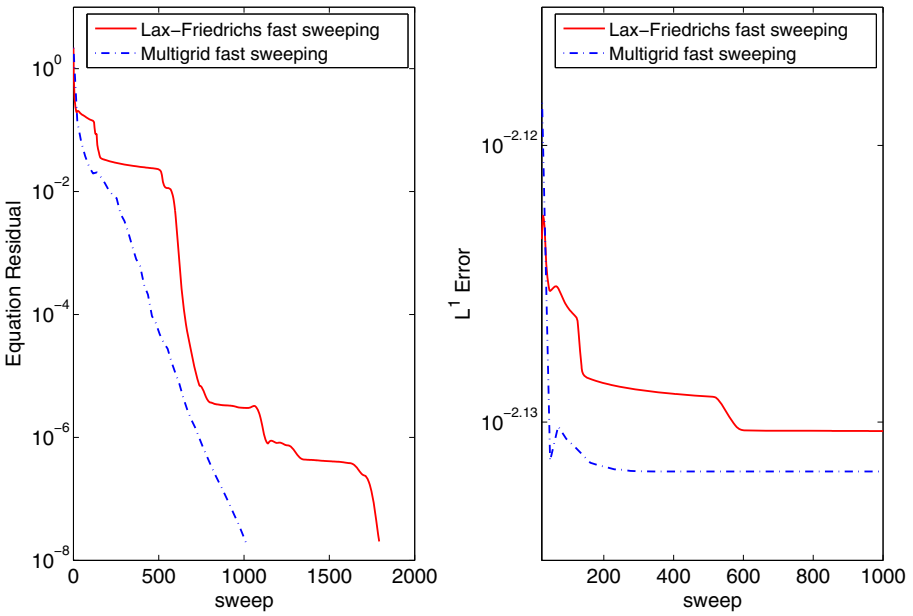

Fig. 16 Example 4.7.: comparison of the residual (*left*) and L^1 error (*right*) by MG method (*blue dotted line*) in 97.08 s and Lax–Friedrichs fast sweeping method (*red solid line*) in 176.49 s (Color figure online)

4.3 The Two-Dimensional Scalar Problems

Example 4.8 We consider the steady state solution of the following problem:

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

where $(x, y) \in [0, \frac{1}{\sqrt{2}}] \times [0, \frac{1}{\sqrt{2}}]$, ϵ can be varied among different values. This is the one-dimensional problem in Example 4.2 along the diagonal direction if $\epsilon = 1$. Inflow boundary conditions are given by the exact solution of the steady state problem. Since our grids are not aligned with the flow direction, this is a truly two-dimensional test case. As before, this problem has two steady state solutions with shocks

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

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

The initial condition is given by:

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

where the initial jump is located in the middle of the positions of the shocks in the two admissible steady state solutions. We test the cases $\epsilon = 1$ and $\epsilon = 10^{-6}$. When ϵ is small, semi-coarsening is performed to guarantee the convergence of MG method. The mesh size we used is 64×64 and three iterations are implemented on the finest mesh. First, we test the

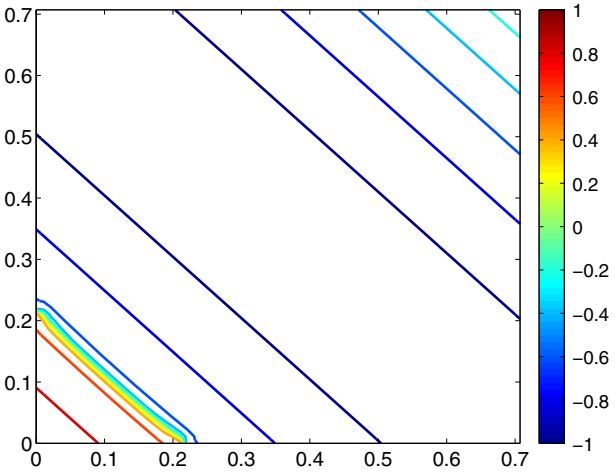

Fig. 17 Example 4.8. with $\epsilon = 1$: contour plot of the numerical solution on a 64×64 mesh by MG fast sweeping method

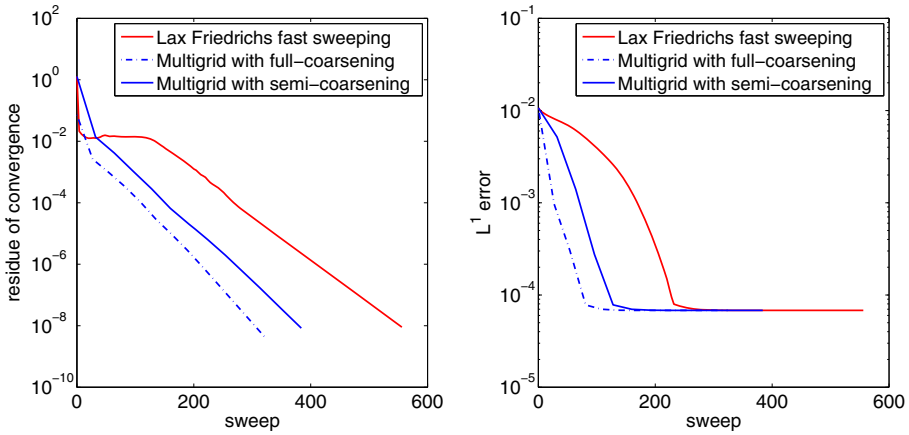

Fig. 18 Example 4.8. with $\epsilon = 1$: comparison of the residual (*left*) and L^1 error (*right*) by MG method with full coarsening (*blue dotted line*) in 163 s, and Lax–Friedrichs fast sweeping method (*red solid line*) in 284.02 s (Color figure online)

case $\epsilon = 1$ and apply both the MG and Lax–Friedrichs fast sweeping method. We show the solution in Fig. 17 and compare the convergence history in Fig. 18. We also implement the semi-coarsening in this case and it is not as efficient as full coarsening.

In the case of $\epsilon = 10^{-6}$ shown in Fig. 19, MG with semi-coarsening works well and saves at least half of the sweeps while the residue in full-coarsening cannot drop to machine error.

Example 4.9 We consider the two-dimensional problem

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

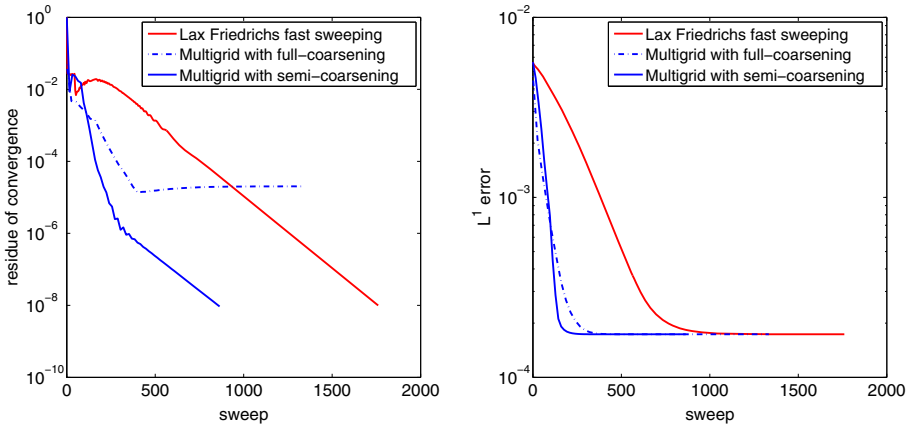

Fig. 19 Example 4.8. With $\epsilon = 10^{-6}$: comparison of residue and L^1 error on a 128×128 mesh by MG and LF fast sweeping method

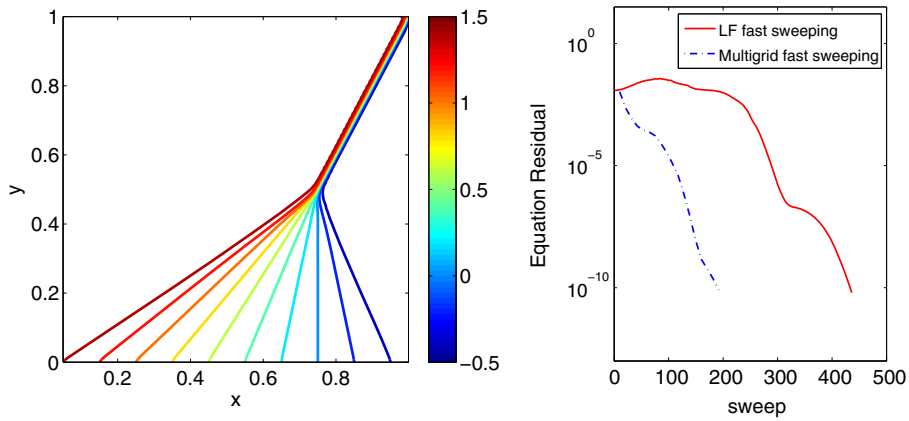

Fig. 20 Example 4.9.: contour plot of the steady state by MG method (*left*) and comparison of the residual (*right*) by MG method (*blue dotted line*) and Lax–Friedrichs fast sweeping method (*red solid line*) (Color figure online)

with boundary conditions:

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

The initial condition is $u(x, y, 0) = 1.5 - 2x$. The steady state has a rarefaction wave for $y < 0.5$ and a singular shock for $y > 0.5$. Because the flux on y is $g(u) = u$, only the boundary condition at $y = 0$ is necessary. Numerically we apply the third order extrapolation on the top boundary of y. The solution and the comparison of residual history are provided in Fig. 20. We can see that the MG method can resolve both the rarefaction wave and shock more efficiently than Lax–Friedrichs fast sweeping and saves about half of the sweeps. We also look into the CPU time by different methods. To achieve the same stopping criteria, Lax–Friedrich fast sweeping method takes 1237.06 s while the MG solver takes 594.54 s.

4.4 The Two-Dimensional Systems

Example 4.10 We consider a Cauchy–Riemann problem:

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} + B\frac{\partial W}{\partial y} = 0, \qquad (x, y) \in [-2, 2] \times [-2, 2], \quad t > 0, \tag{13}$$

where

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

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

$$u = \begin{cases} 1, & \text{if } x > 0 \text{ and } y > 0 \\ -1, & \text{if } x < 0 \text{ and } y > 0 \\ -1, & \text{if } x > 0 \text{ and } y < 0 \\ 1, & \text{if } x < 0 \text{ and } y < 0 \end{cases}, \text{ and } v = \begin{cases} 1, & \text{if } x > 0 \text{ and } y > 0 \\ -1, & \text{if } x < 0 \text{ and } y > 0 \\ -1, & \text{if } x > 0 \text{ and } y < 0 \\ 2, & \text{if } x < 0 \text{ and } y < 0 \end{cases}. \tag{14}$$

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

$$(-\xi I + A)\frac{\partial \tilde{W}}{\partial \xi} + (-\eta I + B)\frac{\partial \tilde{W}}{\partial \eta} = 0,$$

which can be written as

$$\frac{\partial}{\partial \xi}[(-\xi I + A)\tilde{W}] + \frac{\partial}{\partial \eta}[(-\eta I + B)\tilde{W}] = -2\tilde{W}, \tag{15}$$

with the boundary conditions at infinity given by the Riemann data in (13) and (14) at time $t = 1$. Eq. (15) can be solved with boundary conditions as the exact solution and the same initial condition as in (14). Boundary values on the ghost points in the WENO reconstruction are set as

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

We test it by MG fast sweeping method with a single iteration on the finest mesh and compare the convergence history with Lax–Friedrichs fast sweeping method. The numerical solution is shown in Fig. 21 and the comparison plot is in Fig. 22.

Example 4.11 We consider a regular shock reflection problem of the steady state solution of the two-dimensional Euler equations:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = 0, \quad (x, y) \in [0, 4] \times [0, 1] \tag{16}$$

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 test.

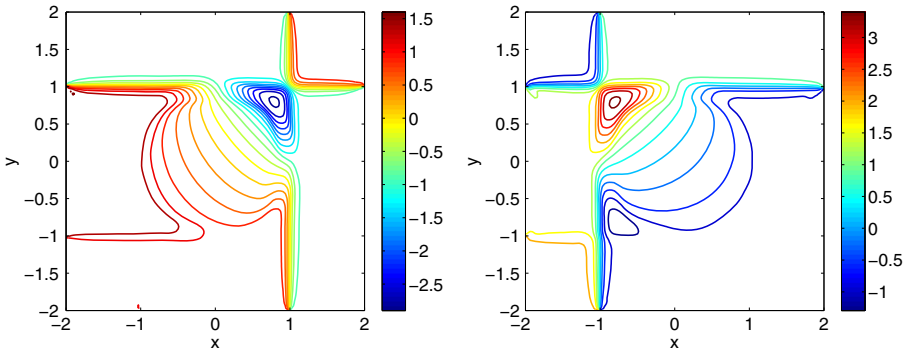

Fig. 21 Example 4.10.: contour plot of numerical solution u (*left*) and v (*right*) by MG fast sweeping method with 128×128 cells on the finest meth

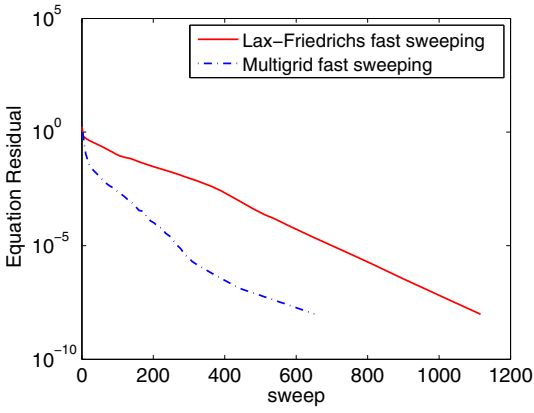

Fig. 22 Example 4.10.: comparison of the residual by MG method (*blue dotted line*) in 10470 s and Lax–Friedrichs fast sweeping method (*red solid line*) in 18300 s (Color figure online)

The initial condition is taken to be

$$(\rho, u, v, p) = (1.69997, 2.61934, -0.50632, 1.52819) \quad \text{on} \quad y = 1,$$
$$(\rho, u, v, p) = (1, 2.9, 0, \frac{1}{\gamma}) \quad \text{otherwise.}$$

The boundary conditions are given by

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

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. When we apply the MG procedure, the finest mesh is set to be 160×40 and it only allows us to implement relaxation on four levels of grids. The numerical results are shown in Fig. 23. It can be clearly seen that the incident and reflected shocks are well-resolved. By comparing with Lax–Friedrichs fast sweeping method in Fig. 24, the residue can drop to 10^{-3} within much less sweeps when MG solver is applied.

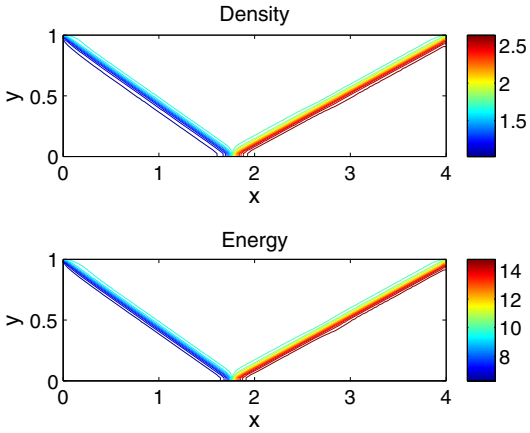

Fig. 23 Ex 4.11.: contour plot of numerical solution ρ (*top*) and E (*bottom*) by MG fast sweeping method with 160×40 cells on the finest meth

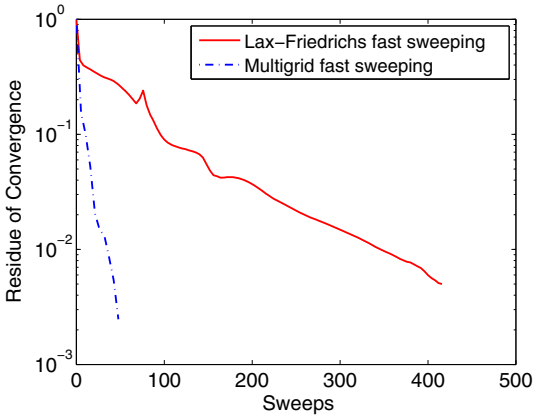

Fig. 24 Ex 4.11.: comparison of the residual by MG method (*blue dotted line*) in 1193.5 s and Lax–Friedrichs fast sweeping method (*red solid line*) in 4933.8 s (Color figure online)

5 Conclusions

In this paper, we proposed the Lax–Friedrichs fast sweeping WENO method coupled with MG framework. Based on the previously developed fast steady state solver for hyperbolic conservation problems, coupling this smoother with MG method significantly accelerates the computation. In our algorithm, full approximation scheme is used to solve the nonlinear system; however, due to the hyperbolicity of the problems, the regular interpolation within MG method cannot be used. Instead, we need to use upwinding in the interpolation, and in particular, WENO upwind interpolation between coarse and fine meshes is used to ensure convergence.

This method has been applied to various problems. The numerical results demonstrate its high order accuracy, capability of capturing shocks and its efficiency compared to the regular Lax–Friedrichs fast sweeping WENO method, which has previously been shown to be much faster than time evolution schemes. Future work includes developing fast sweeping method using upwind fluxes and coupling with MG framework to achieve even lower computational costs.

References

1. Abgrall, R.: Toward the ultimate conservative scheme: following the quest. J. Comput. Phys. **167**, 277–315 (2001)
2. Abgrall, R., Barth, T.: Residual distribution schemes for conservation laws via adaptive quadrature. SIAM J. Sci. Comput. **24**, 732–769 (2002)
3. Abgrall, R., Mezine, M.: Construction of second order accurate monotone and stable residual distribution schemes for steady problems. J. Comput. Phys. **195**, 474–507 (2004)
4. Abgrall, R., Roe, P.L.: High order fluctuation scheme on triangular meshes. J. Sci. Comput. **19**, 3–36 (2003)
5. Amarala, S., Wan, J.: Multigrid methods for systems of hyperbolic conservation laws. Multiscale Model. Simul. **11**(2), 586–614 (2013)
6. Buckley, Se E., Leverett, MCi: Mechanism of fluid displacement in sands. Trans. AIME **146**(107), 1–7 (1942)
7. Chen, W., Chou, C.-S., Kao, C.-Y.: Lax-Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws. J. Comput. Phys. **234**, 452–471 (2013)
8. Chou, C.-S., Shu, C.-W.: High order residual distribution conservative finite difference WENO schemes for steady state problems on non-smooth meshes. J. Comput. Phys. **214**, 698–724 (2006)
9. Deconinck, H., Struijs, R., Bourgeois, G., Roe, P.: Compact advection schemes on unstructured meshes, computational fluid dynamics. Computational Fluid Dynamics, VKI Lecture Series 1993–04, 1993
10. Embid, P., Goodman, J., Majda, A.: Multiple steady states for 1-d transonic flow. SIAM J. Sci. Stat. Comput. **5**, 21–41 (1984)
11. Goedbloed, J., Poedts, S.: Principles of Magnetohydrodynamics: With Applications to Laboratory and Astrophysical Plasmas. Cambridge University Press, (2004)
12. Harten, A., Hyman, J., Lax, P., Keyfitz, B.: On finite-difference approximations and entropy conditions for shocks. Commun. Pure Appl. Math. **29**(3), 297–322 (1976)
13. Hemker, P.W., Spekreijse, S.P.: Multigrid solution of the steady euler equations. In Dietrich B., Wolfgang H., and Ulrich T., (eds.), Advances in Multi-Grid Methods, of Notes on Numerical Fluid Mechanics, vol. 11, pp. 33–44. Vieweg+Teubner Verlag, (1985)
14. Jespersen, D.: Design and implementation of a multigrid code for the euler equations. Appl. Math. Comput. **13**(3–4), 357–374 (1983)
15. Jiang, G., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
16. Koren, B., Hemker, P.W.: Damped, direction-dependent multigrid for hypersonic flow computations. Appl. Numer. Math. **7**(4), 309–328 (1991)
17. Lax, P., Wendroff, B.: Systems of conservation laws. Commun. Pure Appl. Math. **13**(2), 217–237 (1960)
18. Leclercq, M.P., Stoufflet, B.: Characteristic multigrid method application to solve the euler equations with unstructured and unnested grids. J. Comput. Phys. **104**(2), 329–346 (1993)
19. Leveque, R.J.: Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge (2002)
20. Liepmann, H.W., Roshko, A.: Elements of Gas Dynamics. Wiley, New York (1957)
21. Liu, T.P.: Hyperbolic and viscous conservation laws. CBMS-NSF regional conference series in applied mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000
22. Liu, X.-D., Osher, S., Chan, T.: Weighted essentially nonoscillatory schemes. J. Comput. Phys. **115**, 200–212 (1994)
23. Roberts, T., Sidilkover, D., Swanson, R.C.: Textbook multigrid efficiency for the steady Euler equations. Aiaa Pap. **97**, 1949 (1997)
24. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys. **43**, 357–372 (1981)
25. Roe, P.L., Sidilkover, D.: Optimum positive linear schemes for advection in two or three dimensions. SIAM J. Numer. Anal. **29**, 1542–1588 (1992)
26. Shu, C.-W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In Quarteroni, A. (ed.) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, vol. 1697, pp. 325–432. Springer (1998)
27. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
28. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. II. J. Comput. Phys. **83**, 32–78 (1989)
29. Sidikover, D., Brandt, A.: Multigrid solution to steady-state two-dimensional conservation laws. SIAM J. Numer. Anal. **30**(1), 249–274 (1993)

30. Struijs, R., Deconinck, H., Roe, P.L.: Fluctuation splitting schemes for the 2d euler equations. In: In its Computational Fluid Dynamics 94 p (SEE N91-32426 24–34), vol. 1, 1991
31. Tsai, Y.-H.R., Cheng, L.-T., Osher, S., Zhao, H.-K.: Fast sweeping algorithms for a class of Hamilton–Jacobi equations. SIAM J. Numer. Anal. **41**, 673–694 (2003)
32. Vincenti, W.G., Kruger, C.H.: Introduction to Physical Gas Dynamics. Wiley, New York (1965)
33. Wan, J.W.L., Jameson, A.: Monotonicity preserving multigrid time stepping schemes for conservation laws. Comput. Vis. Sci. **11**(1), 41–58 (2008)
34. Wesseling, P.: Principles of Computational Fluid Dynamics. Springer, Secaucus, NJ (2000)
35. 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)