INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 11, Number 1, Pages 117–130

FIXED-POINT FAST SWEEPING WENO METHODS FOR STEADY STATE SOLUTION OF SCALAR HYPERBOLIC CONSERVATION LAWS

SHANQIN CHEN

Abstract. Fast sweeping methods were developed in the literature to efficiently solve static Hamilton-Jacobi equations. This class of methods utilize the Gauss-Seidel iterations and alternating sweeping strategy to achieve fast convergence rate. They take advantage of the properties of hyperbolic partial differential equations (PDEs) and try to cover a family of characteristics of the corresponding Hamilton-Jacobi equation in a certain direction simultaneously in each sweeping order. In [16], the Gauss-Seidel idea and alternating sweeping strategy were adopted to the time-marching type fixed-point iterations to solve the static Hamilton-Jacobi equations, and numerical examples verified at least a 2 times acceleration of convergence even on relatively coarse grids. In this paper, we apply the same approach to solve steady state solution of hyperbolic conservation laws. We use numerical examples to verify that a 2 times acceleration of convergence is achieved. And the computational cost is exactly the same as the time-marching scheme at each iteration. Based on the Gauss-Seidel iterations, we explore the successive overrelaxation (SOR) approach to further improve the performance of our fixed-point sweeping methods.

Key words. fast sweeping methods, WENO methods, Jacobi iteration, Gauss-Seidel iteration, hyperbolic conservation laws, steady state.

1. Introduction

Steady state problems for hyperbolic conservation laws and related Hamilton-Jacobi (HJ) equations are common mathematical models appearing in many applications, such as fluid mechanics, optimal control, differential games, image processing and computer vision, geometric optics, etc. For these boundary value problems, their solution information propagates along characteristics starting from the boundary. A class of iterative methods, called fast sweeping (FS) methods [1, 6, 9, 12, 14, 16, 17, 18], take advantage of this property and try to cover a family of characteristics of the HJ equations in a certain direction simultaneously in each iteration. This iterative technique can achieve very fast convergence for computations of steady state solutions. Fast sweeping methods actually provide a general methodology / technique to accelerate the convergence of numerical schemes for steady state problems of hyperbolic type PDEs, although currently they are mostly used for solving HJ equations.

Since fast sweeping technique mainly takes advantage of the characteristics properties of hyperbolic PDEs to accelerate the iteration convergence, it is natural to apply this technique for solving steady states of general hyperbolic PDEs [2, 8]. In [16], we proposed fixed-point fast sweeping methods for static Hamilton-Jacobi equations. The fixed-point fast sweeping approach is based on the time marching approach and it has the advantages that the method is explicit and free of solving nonlinear equations, and it is straightforward to apply high order approximations and different numerical Hamiltonian for the general Hamilton-Jacobi equations.

Received by the editors December 22, 2012 and, in revised form, April 3, 2013.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.
This research was supported by IU South Bend Research and Development grant.

In this paper we apply this "explicit fast sweeping technique", or the "fixed-point fast sweeping method" to solve the steady state problems of hyperbolic conservation laws. The Gauss-Seidel idea and alternating sweeping strategy are adopted to the time marching approach to accelerate its convergence to steady states *without* any additional computational cost. Via numerical experiments, we verify this acceleration. In this paper, we use the standard third order finite difference weighted essentially non-oscillatory (WENO) scheme with Lax-Friedrichs flux splitting [5] as the representation of high order schemes for hyperbolic conservation laws. But this general approach can be directly applied to other schemes such as the residual distribution WENO schemes [3] or Runge-Kutta discontinuous Galerkin methods [4]. It can also be applied to other numerical fluxes such as Godunov flux, etc [10]. Based on the Gauss-Seidel iterations, we explore the successive overrelaxation (SOR) approach to further improve the performance of our fixed-point sweeping methods.

In Section 2, we describe the fixed-point fast sweeping WENO methods for solving hyperbolic conservation laws, based on Gauss-Seidel iterations and SOR iterations respectively. In Section 3, Numerical studies are performed to verify the faster convergence speed than the usual time-marching approach. Concluding remarks are given in Section 4.

2. Fixed-point fast sweeping WENO methods

Consider two-dimensional steady state problems of hyperbolic conservation laws with appropriate boundary conditions

$$f(u)_x + g(u)_y = h(u, x, y), \tag{1}$$

where u is the unknown function, f and g are flux functions, and h is the source term. A high order spatial discretization of (1) leads to a nonlinear system. In this paper, we use the third order finite difference WENO scheme with Lax-Friedrichs flux splitting [10] for the spatial discretization.

2.1. WENO discretization. For the hyperbolic terms $f(u)_x + g(u)_y$, the conservative finite difference scheme we use approximates the point values at a uniform (or smoothly varying) grid (x_i, y_j) in a conservative fashion. Namely, the derivative $f(u)_x$ at (x_i, y_j) is approximated along the line $y = y_j$ by a conservative flux difference

$$f(u)_x|_{x=x_i} \approx \frac{1}{\Delta x}(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}), \tag{2}$$

where for the third order WENO scheme the numerical flux $\hat{f}_{i+1/2}$ depends on the three-point values $f(u_l)$, $l = i-1, i, i+1$, when the wind is positive (i.e., when $f'(u) \geq 0$ for the scalar case, or when the corresponding eigenvalue is positive for the system case with a local characteristic decomposition). This numerical flux $\hat{f}_{i+1/2}$ is written as a convex combination of two second order numerical fluxes based on two different substencils of two points each, and the combination coefficients depend on a "smoothness indicator" measuring the smoothness of the solution in each substencil. The detailed formula is

$$\hat{f}_{i+1/2} = w_0 \left[\frac{1}{2}f(u_i) + \frac{1}{2}f(u_{i+1})\right] + w_1 \left[-\frac{1}{2}f(u_{i-1}) + \frac{3}{2}f(u_i)\right], \tag{3}$$

where

$$w_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \qquad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \qquad r = 0, 1. \tag{4}$$

$d_0 = 2/3, d_1 = 1/3$ are called the "linear weights", and $\beta_0 = (f(u_{i+1}) - f(u_i))^2, \beta_1 = (f(u_i) - f(u_{i-1}))^2$ are called the "smoothness indicators". ϵ is a small positive number chosen to avoid the denominator becoming 0. We take $\epsilon = 10^{-6}$ in this paper.

When the wind is negative (i.e., when $f'(u) < 0$), right-biased stencil with numerical values $f(u_i), f(u_{i+1})$ and $f(u_{i+2})$ are used to construct a third order WENO approximation to the numerical flux $\hat{f}_{i+1/2}$. The formulae for negative and positive wind cases are symmetric with respect to the point $x_{i+1/2}$. For the general case of $f(u)$, we perform the "Lax-Friedrichs flux splitting"

$$f^+(u) = \frac{1}{2}(f(u) + \alpha u), \qquad f^-(u) = \frac{1}{2}(f(u) - \alpha u), \tag{5}$$

where $\alpha = \max_u |f'(u)|$. $f^+(u)$ is the positive wind part, and $f^-(u)$ is the negative wind part. Corresponding WENO approximations are applied to find numerical fluxes $\hat{f}^+_{i+1/2}$ and $\hat{f}^-_{i+1/2}$ respectively. Then $\hat{f}_{i+\frac{1}{2}} = \hat{f}^+_{i+1/2} + \hat{f}^-_{i+1/2}$. Similar procedures are applied to the y direction for $g(u)_y$. See [10] for more details. Then we obtain a nonlinear system

$$0 = -(\hat{f}_{i+1/2,j} - \hat{f}_{i-1/2,j})/\Delta x - (\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2})/\Delta y + h(u_{ij}, x_i, y_j), \tag{6}$$
$$i = 1, \cdots, N; j = 1, \cdots, M.$$

where $\hat{f}$, $\hat{g}$ are numerical fluxes obtained by Lax-Friedrichs flux splitting and WENO approximation.

2.2. Gauss-Seidel fixed-point fast sweeping iterative scheme. We use pseudo-time-marching approach to derive the fixed-point sweeping method for the nonlinear system (6). We add a time derivative to the nonlinear system (6) and obtain an ODE system

$$du_{ij}(t)dt = -(\hat{f}_{i+1/2,j} - \hat{f}_{i-1/2,j})/\Delta x - (\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2})/\Delta y + h(u_{ij}, x_i, y_j), \tag{7}$$

The right-hand-side (RHS) of (7) is a nonlinear function of numerical values at the grid points of computational stencils. We denote it by L and can write (7) in the following form

$$du_{ij}(t)dt = L(u_{i-r,j}, \cdots, u_{i+s,j}; u_{ij}; u_{i,j-r}, \cdots, u_{i,j+s}), \tag{8}$$
$$i = 1, \cdots, N; j = 1, \cdots, M,$$

where r, s are values which depend on the order of the WENO approximation. For the third order WENO scheme used in this paper, we have $r = s = 2$. A time-marching method is often used to march the system (8) to the steady state. Time-marching approach is actually a Jacobi type fixed-point iterative scheme. For example the third order TVD Runge-Kutta scheme [11] has the following form for the iteration step n:

$$u^{(1)}_{ij} = u^n_{ij} + \Delta t L(u^n_{i-r,j}, \cdots, u^n_{i+s,j}; u^n_{ij}; u^n_{i,j-r}, \cdots, u^n_{i,j+s}), \tag{9}$$
$$i = 1, \cdots, N; j = 1, \cdots, M.$$

$$u^{(2)}_{ij} = \frac{3}{4}u^n_{ij} + \frac{1}{4}u^{(1)}_{ij} + \frac{1}{4}\Delta t L(u^{(1)}_{i-r,j}, \cdots, u^{(1)}_{i+s,j}; u^{(1)}_{ij}; u^{(1)}_{i,j-r}, \cdots, u^{(1)}_{i,j+s}), \tag{10}$$
$$i = 1, \cdots, N; j = 1, \cdots, M.$$

$$u^{n+1}_{ij} = \frac{1}{3}u^n_{ij} + \frac{2}{3}u^{(2)}_{ij} + \frac{2}{3}\Delta t L(u^{(2)}_{i-r,j}, \cdots, u^{(2)}_{i+s,j}; u^{(2)}_{ij}; u^{(2)}_{i,j-r}, \cdots, u^{(2)}_{i,j+s}), \tag{11}$$
$$i = 1, \cdots, N; j = 1, \cdots, M.$$

To accelerate the computation and reduce the iteration steps to the steady state, we apply the fast sweeping technique, namely, the Gauss-Seidel philosophy and alternating direction iterations. By the Gauss-Seidel philosophy, the newest available numerical values of u are used in the interpolation stencils as long as they are available. Hence (9)-(11) give the following fixed-point fast sweeping scheme:

$$u_{ij}^{(1)} = u_{ij}^{n} + \frac{\gamma}{\alpha_x/\Delta x + \alpha_y/\Delta y} L(u_{i-r,j}^{*}, \cdots, u_{i+s,j}^{*}; u_{ij}^{n}; u_{i,j-r}^{*}, \cdots, u_{i,j+s}^{*}), \tag{12}$$
$$i = i_1, \cdots, i_N; j = j_1, \cdots, j_M.$$

$$u_{ij}^{(2)} = u_{ij}^{(1)} + \frac{\gamma}{4(\alpha_x/\Delta x + \alpha_y/\Delta y)} L(u_{i-r,j}^{**}, \cdots, u_{i+s,j}^{**}; u_{ij}^{(1)}; u_{i,j-r}^{**}, \cdots, u_{i,j+s}^{**}), \tag{13}$$
$$i = i_1, \cdots, i_N; j = j_1, \cdots, j_M.$$

$$u_{ij}^{n+1} = u_{ij}^{(2)} + \frac{2\gamma}{3(\alpha_x/\Delta x + \alpha_y/\Delta y)} L(u_{i-r,j}^{***}, \cdots, u_{i+s,j}^{***}; u_{ij}^{(2)}; u_{i,j-r}^{***}, \cdots, u_{i,j+s}^{***}), \tag{14}$$
$$i = i_1, \cdots, i_N; j = j_1, \cdots, j_M.$$

The above schemes (12)-(14) denote a complete iteration step n which includes three sub-iterations. The complete iterations do *not* just proceed in only one direction $i = 1 : N, j = 1 : M$ as the time-marching approach (9)-(11), but in the following four alternating directions repeatedly,

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

Note that the sweeping directions of the three sub-iterations (12)-(14) are the same inside a complete iteration step n. Since the strategy of alternating direction sweepings utilizes the characteristics property of hyperbolic PDEs, combining with the Gauss-Seidel philosophy, we will observe the acceleration of convergence speed for time-marching approach, which will be shown in the following numerical experiments. By the Gauss-Seidel philosophy, we use the newest numerical values on the computational stencil of the WENO scheme whenever they are available. That is the reason why we use notations such as u^*, u^{**}, u^{***} to represent the values in the scheme (12)-(14). For example $u_{k,l}^{*}$ could be $u_{k,l}^{n}$ or $u_{k,l}^{(1)}$, depending on the current sweeping direction; similarly $u_{k,l}^{**}$ could be $u_{k,l}^{(1)}$ or $u_{k,l}^{(2)}$, and $u_{k,l}^{***}$ could be $u_{k,l}^{(2)}$ or $u_{k,l}^{n+1}$. In our computer implementation, we just use one array for all of these u^n, $u^{(1)}$, and u^{n+1}. $\alpha_x = \max_u |f'(u)|$ and $\alpha_y = \max_u |g'(u)|$ are viscosity constants in the Lax-Friedrichs flux [10]. The parameter γ is chosen to be suitable values which can gurantee that the fixed-point iteration is a contractive mapping and it converges. The parameter γ actually represents the CFL number in the original time-marching approach (9)-(11) which determines the time step size Δt. In this paper, γ is choosen as 0.5.

Remark: There are two important components in fast sweeping methods. First a stable numerical scheme is needed to discretize the hyperbolic PDEs. A monotone numerical flux such as the Lax-Friedrichs flux is used as the building block for high order WENO schemes. Monotone numerical fluxes capture the up-winding information of the hyperbolic PDEs and guarantee the linear stability of the numerical scheme. Characteristic information has been embedded in the resulting nonlinear

system. Fast sweeping method uses Gauss-Seidel iterations to solve the system in alternating sweeping directions. If the sweeping direction is consistent with the characteristic propagation direction for a group of grid points, correct characteristic information from the given boundary values will propagate to that group and correct numerical values are obtained. So the second important component is to have a systematic ordering of all grid points, which can cover all directions of the characteristics. On a two-dimensional rectangular mesh, a characteristic direction can always be decomposed to two directions ($x-$ and $y-$ directions), and the natural orderings of all grid points give four sweeping directions as shown in this section. See [18] for detailed discussion on this.

2.3. SOR fixed-point fast sweeping iterative scheme. The successive overrelaxation method (SOR) is a method of solving a linear system of equations $Ax = b$ derived by extrapolating the Gauss-Seidel method. It modifies Gauss-Seidel by adding a relaxation parameter ω to improve the computation performance [7]. This extrapolation takes the form of a weighted average between the previous iterate and the computed Gauss-Seidel iterate successively for each component:

$$x_i^{(k)} = \omega \tilde{x}_i^{(k)} + (1-\omega) x_i^{(k-1)},$$

where $\tilde{x}$ denotes a Gauss-Seidel iterate and ω is the extrapolation factor. The idea is to choose a value for ω that will accelerate the rate of convergence of the iterates to the solution. The performance of Gauss-Seidel iterations can be improved with a good choice of ω. However, the optimal value of ω and the effectiveness of acceleration are problem dependent [7]. Implementing SOR into our Gauss-Seidel fixed-point fast sweeping iterative scheme (12)-(14), we obtain the following SOR fixed-point fast sweeping iterative scheme:

$$\begin{cases} \tilde{u}_{ij}^{(1)} = u_{ij}^n + \frac{\gamma}{\alpha_x/\Delta x + \alpha_y/\Delta y} L(u_{i-r,j}^*, \cdots, u_{i+s,j}^*; u_{ij}^n; u_{i,j-r}^*, \cdots, u_{i,j+s}^*), \\ u_{ij}^{(1)} = \omega \tilde{u}_{ij}^{(1)} + (1-\omega) u_{ij}^n; \\ \qquad\qquad i = i_1, \cdots, i_N; j = j_1, \cdots, j_M. \end{cases}$$

$$\begin{cases} \tilde{u}_{ij}^{(2)} = u_{ij}^{(1)} + \frac{\gamma}{4(\alpha_x/\Delta x + \alpha_y/\Delta y)} L(u_{i-r,j}^{**}, \cdots, u_{i+s,j}^{**}; u_{ij}^{(1)}; u_{i,j-r}^{**}, \cdots, u_{i,j+s}^{**}), \\ u_{ij}^{(2)} = \omega \tilde{u}_{ij}^{(2)} + (1-\omega) u_{ij}^{(1)}; \\ \qquad\qquad i = i_1, \cdots, i_N; j = j_1, \cdots, j_M. \end{cases}$$

$$\begin{cases} \tilde{u}_{ij}^{(n+1)} = u_{ij}^{(2)} + \frac{2\gamma}{3(\alpha_x/\Delta x + \alpha_y/\Delta y)} L(u_{i-r,j}^{***}, \cdots, u_{i+s,j}^{***}; u_{ij}^{(2)}; u_{i,j-r}^{***}, \cdots, u_{i,j+s}^{***}), \\ u_{ij}^{n+1} = \omega \tilde{u}_{ij}^{(n+1)} + (1-\omega) u_{ij}^{(2)}; \\ \qquad\qquad i = i_1, \cdots, i_N; j = j_1, \cdots, j_M. \end{cases}$$

The Gauss-Seidel approach is a special case of the SOR approach with $\omega = 1$. In numerical experiments of the next section, we observe that the optimal value of ω is *not* sensitive to the computational mesh sizes. Namely, for a specific problem, the optimal value of ω is approximately same for all mesh sizes. This provides a method on how to apply SOR to accelerate computations. We can first test SOR on a coarse mesh using different ω values and find an optimal ω value. Then this optimal ω value can be applied to more refined meshes to accelerate computations.

3. Numerical examples

In this section, a set of one- and two- dimensional numerical examples are presented to test the proposed fixed-point fast sweeping idea for solving steady state

scalar conservation laws. Numerical experiments verify at least 2 times acceleration of convergence, without any additional computational cost added to the original time-marching approach such as (9)-(11). For some mesh sizes, especially in two dimensional cases, the original time-marching scheme does not even converges, but our fast sweeping scheme can converge quickly. We consider an iteration is convergent if L^1 norm of the difference between two successive iteration steps $||u^n - u^{n-1}||_{L^1} \leq \delta$ (called L^1 convergence error). δ is taken to be 10^{-11} in this paper. In all of the examples to test our numerical algorithms, we use exact solutions to specify numerical values in the inflow boundaries, and extrapolations for the outflow boundaries.

Example 1. We solve the steady state solution of a one-dimensional Burgers equation with a source term

$$\begin{cases} u_t - (\frac{1}{2}u^2)_x = u, & 0 \leq x \leq 1; \\ u(0) = \frac{3}{4}, \quad u(1) = -\frac{1}{2}. \end{cases}$$

The initial condition is given by

$$u(x,0) = \begin{cases} \frac{3}{4}, & x < 0.5; \\ -\frac{1}{2}, & x \geq 0.5. \end{cases}$$

The exact steady state solution is a piecewise linear function with one shock wave

$$u(x) = \begin{cases} \frac{3}{4} - x, & 0 \leq x < 0.625; \\ -x + \frac{1}{2}, & 0.625 \leq x \leq 1. \end{cases}$$

We perform the convergence study by using successive refined meshes. The numerical solution by our Gauss-Seidel fixed-point fast sweeping method and the exact solution are displayed on the left in Figure 1. We can see that the numerical shock is at the correct location and is resolved well. The numerical accuracy errors and orders, and the iteration numbers by both the time-marching and our Gauss-Seidel fixed-point fast sweeping method are reported in Table 1. The numerical errors are computed away from shocks. We can see super convergence for this example, since the exact solution of the problem is piecewise linear. We also observe that the iteration numbers for the Gauss-Seidel fixed-point fast sweeping method are all about half of those for the time-marching approach.

Then we study the further convergence acceleration by the SOR fixed-point fast sweeping method with different ω values. The study results are shown in Table 2. We achieve less iteration numbers when using bigger ω until the method diverges. In this case, the SOR fixed-point fast sweeping method converges for $\omega \leq 2.2$. We see that the iterative numbers with $\omega = 2.2$ are less than half of those with $\omega = 1$, which corresponds to the Gauss-Seidel approach. This shows a significant improvement on convergence speed by using the SOR fixed-point fast sweeping method. In Table 2, we show the percentage of iteration numbers saved by using SOR with the optimal ω rather than the Gauss-Seidel iterations. For example, for $N = 160$, the percentage of optimal saving is $(1964 - 834)/1964 = 57.5\%$. For different meshes, the percentage of optimal saving is close to 60%. Also as we mention in the last section, how to choose the optimal ω in SOR for a general problem is still an open problem [7]. However, we observe that $\omega = 2.2$ is the optimal one for all meshes in Table 2.

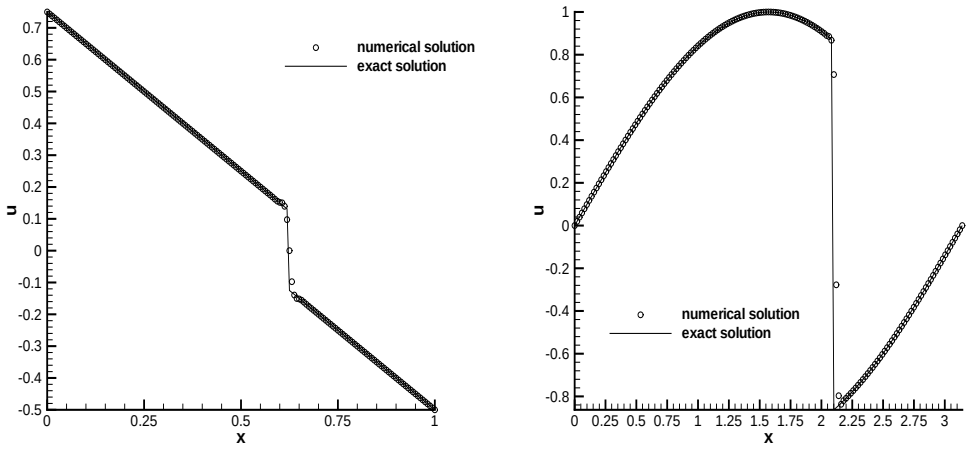

FIGURE 1. The numerical solution (symbols) versus the exact solution (solid line) with $N = 160$ cells for Example 1 (left) and Example 2 (right).

TABLE 1. Example 1, errors and numerical orders of accuracy in the smooth region of the solution, and iteration numbers of time-marching and Gauss-Seidel (GS) fixed-point fast sweeping methods.

N	L_∞ error	order	L_1 error	order	iteration # (time-marching)	iteration # (GS fast sweeping)
40	3.93E-03	-	2.84E-04	-	1098	538
80	5.80E-04	2.76	2.77E-05	3.36	2091	1026
160	2.92E-05	4.31	6.11E-07	5.50	3993	1964
320	4.73E-08	9.27	5.14E-10	10.22	7629	3758
640	1.23E-13	18.56	1.87E-15	18.06	14569	7196

Example 2. We solve the steady state solutions of the Burgers equation with a different source term

$$\begin{cases} u_t + (\frac{1}{2}u^2)_x = \sin x \cos x, & 0 \le x \le \pi; \\ u(0) = u(\pi) = 0. \end{cases}$$

The initial condition is given by $u(x,0) = \frac{1}{2}\sin x$. This problem has a steady state solution with a shock:

$$u(x) = \begin{cases} \sin x, & 0 \le x < \frac{2\pi}{3}; \\ -\sin x, & \frac{2\pi}{3} < x \le \pi. \end{cases}$$

The numerical solution our Gauss-Seidel fixed-point fast sweeping method and the exact solution are displayed on the right in Figure 1. We can see, again, that the numerical shock is at the correct location and is resolved well. We perform the convergence study by using successive refined meshes. The numerical errors are computed away from shocks. From Table 3, we can also see clearly that third order accuracy is achieved in the smooth region of the solution. Again, we observe that

TABLE 2. Example 1, iteration numbers of the SOR fixed-point fast sweeping method on computational grids with $N = 160$, $N = 320$, and $N = 640$.

ω	$N = 160$	$N = 320$	$N = 640$
0.9	2196	4202	8044
1.0	1964	3758	7196
1.5	1264	2424	4646
2.0	920	1756	3374
2.1	876	1660	3192
2.2	834	1608	3026
2.3	not converge	not converge	not converge
optimal saving	57.5%	57.2%	57.9%

the iteration numbers of the Gauss-Seidel fast sweeping method are all about half of those of the time-marching method. Further convergence acceleration can be achieved by using the SOR approach. The iteration numbers of the SOR approach are shown in Table 4. Similar as Example 1, we need less iteration numbers when using bigger ω until the method diverges. In this example, the SOR fixed-point fast sweeping method converges for $\omega \leq 1.9$. We see that the iterative numbers with $\omega = 1.9$ are less than half of those with $\omega = 1$, which corresponds to the Gauss-Seidel approach (the percentage of optimal saving is more than 50% for different meshes). Again we obtain a significant improvement on convergence speed by using the SOR fixed-point fast sweeping method. $\omega = 1.9$ is the optimal one for all meshes, as shown in Table 4.

TABLE 3. Example 2, errors and numerical orders of accuracy in the smooth region of the solution, and iteration numbers of time-marching and GS fixed-point fast sweeping methods.

N	L_∞ error	order	L_1 error	order	iteration # (time-marching)	iteration # (GS fast sweeping)
80	1.26E-05	-	7.37E-06	-	538	270
160	9.17E-07	3.78	4.59E-07	4.01	1041	526
320	9.10E-08	3.33	3.68E-08	3.64	2013	1016
640	1.53E-08	2.57	5.19E-09	2.83	3885	1962

TABLE 4. Example 2, iteration numbers of the SOR fixed-point fast sweeping method on computational grids with $N = 160$, $N = 320$, and $N = 640$.

ω	$N = 160$	$N = 320$	$N = 640$
0.9	588	1134	2192
1.0	526	1016	1962
1.5	336	650	1258
1.9	256	492	946
2.0	not converge	not converge	not converge
optimal saving	51.3%	51.6%	51.8%

Example 3. We test our methods for solving the steady state solution of the two-dimensional nonlinear problem with given inflow boundary conditions:

$$\text{(15)} \qquad \begin{cases} u_t + u_y + (\frac{u^2}{2})_x = 0, & (x,y) \in [0,1] \times [0,1]; \\ u(x,y) = \begin{cases} 1.5, & x = 0, \\ -2.5x + 1.5, & y = 0, \\ -1.0, & x = 1. \end{cases} \end{cases}$$

The initial condition is given by $u(x, y, 0) = -2.5x + 1.5$. The steady state solution is

$$u(x,y) = \begin{cases} u_L, & (x,y) \text{ is between lines } g_1 \text{ and } g_3, \\ u_R, & (x,y) \text{ is between lines } g_2 \text{ and } g_3, \\ (u_R - u_L)(x_p + \frac{y_p(x-x_p)}{y_p - y}) + u_L, & (x,y) \text{ is between lines } g_1 \text{ and } g_2, \end{cases}$$

where $u_L = 1.5, u_R = -1.0, x_p = u_L/(u_L - u_R), y_p = 1/(u_L - u_R)$, and the leftmost characteristic g_1 is given by $y = x/u_L$, the rightmost characteristic g_2 is given by $y = (x-1)/u_R$, the shock g_3 is given by $y = (2x-1)/(u_L + u_R)$. Table 5 gives the numerical results of our GS fast sweeping scheme, and we see clear third order accuracy in the smooth region of the solution, and much less iteration numbers are needed by using the GS fast sweeping method than the time-marching approach. Actually for this example, the time-marching method does not even converge to the steady state when mesh is refined to the level of 160×160, but the GS fast sweeping method still converges. The possible reason that the time-marching method may not converge is due to the "slight post-shock oscillations" of WENO schemes, studied by Zhang and Shu in [15]. Even though these oscillations are small in magnitude and do not affect the "essentially non-oscillatory" property of WENO schemes, they are indeed responsible for the numerical residue to hang at the truncation error level of the scheme instead of settling down to machine zero when steady state problems are solved by the usual time-marching method. Iteration numbers by the SOR fixed-point fast sweeping method are given in Table 6, again, less iteration numbers can be obtained when $\omega > 1.0$. The percentages of optimal saving are in the range of $15\% \sim 20\%$ for different meshes, although the acceleration is not as significant as that for the one dimensional examples. These numerical experiments verify the fact that the effectiveness of SOR acceleration is problem dependent. The optimal value of ω is *not* sensitive to the computational mesh sizes. It is about 1.1 for this example.

The contour lines of the numerical solution and the cross-sections for $y = 0.25$ across the fan, for $y = 0.5$ and $y = 0.75$ across the shock, are displayed in Figure 3. We can clearly observe good resolution of the numerical scheme for this example.

TABLE 5. Example 3, errors and numerical orders of accuracy in the smooth region of the solution, and iteration numbers of time-marching and GS fixed-point fast sweeping methods.

N	L_∞ error	order	L_1 error	order	iteration # (time-marching)	iteration # (GS fast sweeping)
80×80	5.73E-04	2.71	4.45E-05	2.81	347	144
160×160	6.52E-05	3.14	3.74E-06	3.57	not converge	268
320×320	3.76E-06	4.11	2.06E-07	4.18	not converge	528
640×640	3.36E-07	3.48	2.33E-08	3.15	not converge	1052

TABLE 6. Example 3, iteration numbers of the SOR fixed-point fast sweeping method on computational grids with $N = 160 \times 160$, $N = 320 \times 320$, and $N = 640 \times 640$.

ω	$N = 160 \times 160$	$N = 320 \times 320$	$N = 640 \times 640$
1.0	268	528	1052
1.1	228	428	904
1.2	228	456	896
1.3	768	1796	not converge
optimal saving	14.9%	18.9%	14.8%

Example 4. We solve the steady state solution of the two-dimensional Burgers equation with a source term

(16)
$$\begin{cases} u_t + (\frac{u^2}{2\sqrt{2}})_x + (\frac{u^2}{2\sqrt{2}})_y = -\pi \cos(\pi \frac{x+y}{\sqrt{2}})u, & (x,y) \in [0, \frac{1}{\sqrt{2}}] \times [0, \frac{1}{\sqrt{2}}]; \\ u(x,y) = \begin{cases} 1 - \sin(\pi \frac{y}{\sqrt{2}}), & x = 0, \frac{y}{\sqrt{2}} \le x_s, \\ 1 - \sin(\pi \frac{x}{\sqrt{2}}), & y = 0, \frac{x}{\sqrt{2}} \le x_s, \\ -0.1 - \sin(\pi(\frac{1}{2} + \frac{y}{\sqrt{2}})), & x = \frac{1}{\sqrt{2}}, \\ -0.1 - \sin(\pi(\frac{1}{2} + \frac{x}{\sqrt{2}})), & y = \frac{1}{\sqrt{2}}, \end{cases} \end{cases}$$

Where x_s is the shock location. This problem has two steady state solutions with shocks

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

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 first one is stable for small perturbation. The initial condition is given by

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

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 plotted in Figure 3. We can see the correct shock location and a good resolution of the shock. The numerical errors and numerical orders of accuracy in the smooth region of the solution are displayed in Table 7. A slightly better than third order accuracy in the smooth region of the solution is obtained. Again as that in the example 3, the iteration numbers of GS fixed-point fast sweeping are much less than those of time-marching approach. For this example, the time-marching method does not even converge to the steady state when mesh is refined to the level of 160×160, but the GS fixed-point fast sweeping method still converges. Iteration numbers by the SOR fixed-point fast sweeping method are given in Table 8. Again similar as the example 3, less iteration numbers can be obtained when $\omega > 1.0$. The percentages of optimal saving are in the range of $15\% \sim 20\%$ for different meshes. Again, the optimal value of ω is *not* sensitive to the computational mesh sizes. It is 1.1 for this example.

At last, we compared the convergence history between the GS fixed-point fast sweeping and the direct time-marching method on meshes of $N = 80 \times 80$ and $N = 160 \times 160$ cells. The result is displayed in Figure 4. On the 80×80 mesh, the L_1 convergence error of the GS fixed-point fast sweeping method converges to

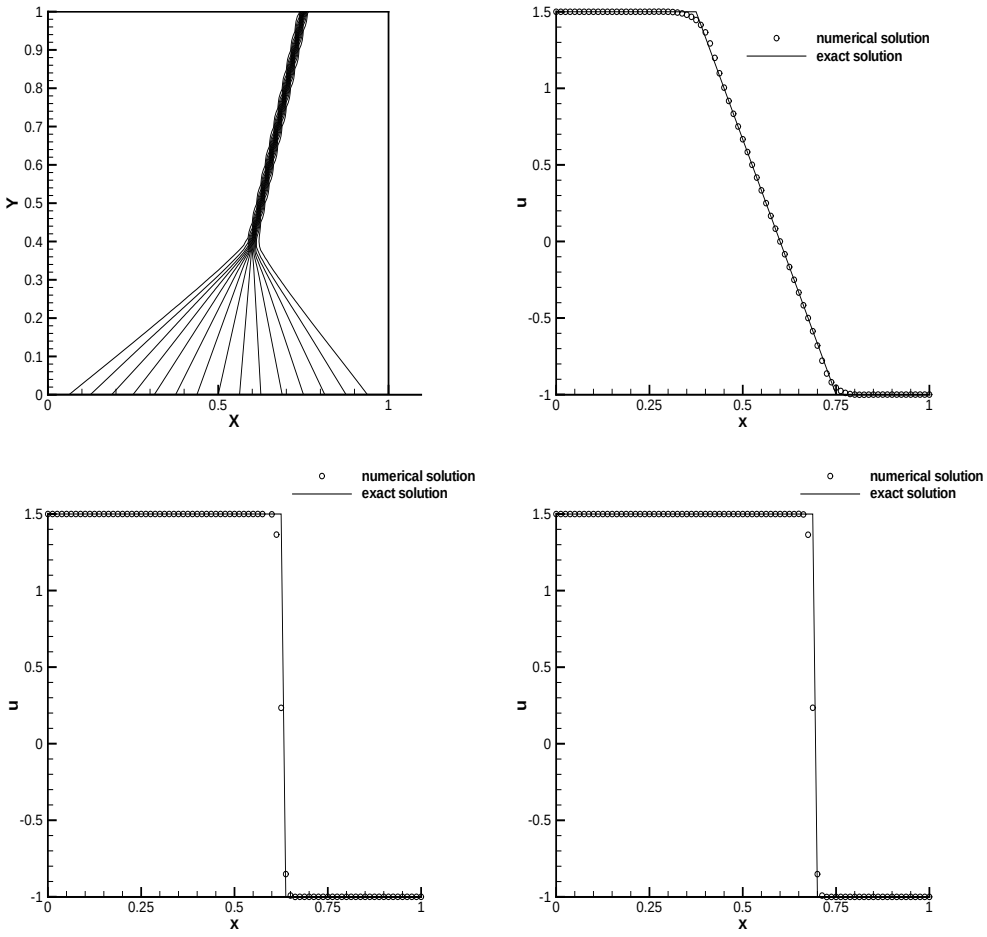

FIGURE 2. Example 3. Numerical result on the mesh with $N = 80 \times 80$ cells. Top left: contour lines from -0.6 to 1.6. Top right: the numerical solution (symbols) versus the exact solution (solid line) along the cross-section at $y = 0.25$; bottom left: cross-section at $y = 0.5$; bottom right: cross-section at $y = 0.75$.

machine zero in less than 500 iterations, while the direct time-marching method needs more than 1200 iterations. On the 160×160 mesh, the L_1 convergence error of the GS fixed-point fast sweeping method converges to machine zero in less than 1000 iterations, while the direct time-marching method fails to converge to the steady state.

4. Concluding remarks

In this paper we extend our work in [16] to solve steady state problems of scalar hyperbolic conservation laws. The Gauss-Seidel idea and alternating sweeping strategy are adopted to the time-marching type fixed-point iterations to accelerate the convergence to steady state solutions of hyperbolic conservation laws. As in

TABLE 7. Example 4, errors and numerical orders of accuracy in the smooth region of the solution, and iteration numbers of time-marching and Gauss-Seidel (GS) fixed-point fast sweeping methods.

N	L_∞ error	order	L_1 error	order	iteration # (time-marching)	iteration # (GS fast sweeping)
80×80	5.63E-05	4.37	8.32E-06	3.62	1299	468
160×160	4.06E-06	3.79	7.16E-07	3.54	not converge	864
320×320	2.44E-07	4.06	5.85E-08	3.68	not converge	1932
640×640	1.27E-07	4.26	3.88E-09	3.85	not converge	3808

TABLE 8. Example 4, iteration numbers of the SOR fixed-point fast sweeping method on computational grids with $N = 160 \times 160$, $N = 320 \times 320$.

ω	$N = 160 \times 160$	$N = 320 \times 320$
1	864	1932
1.05	780	1768
1.1	704	1616
optimal saving	18.5%	16.4%

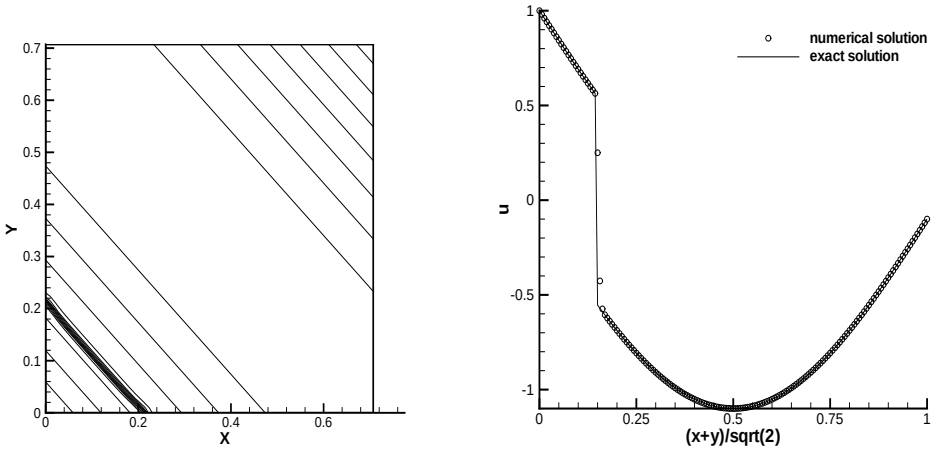

FIGURE 3. Example 4. Numerical result on the mesh with 80×80 cells. Left: Contours of the solution from -1.2 to 1.1; right: the numerical solution (symbols) versus the exact solution (solid line) along the cross-section through the northeast to southwest diagonal.

[16], we obtain a 2 times acceleration of convergence, without any additional computational cost. In numerical examples, we also find that even for the cases that the time-marching approach does *not* converge to the steady state solution, our Gauss-Seidel fixed-point fast sweeping approach converges very well to the desired solution. Based on the Gauss-Seidel iterations, the successive overrelaxation (SOR)

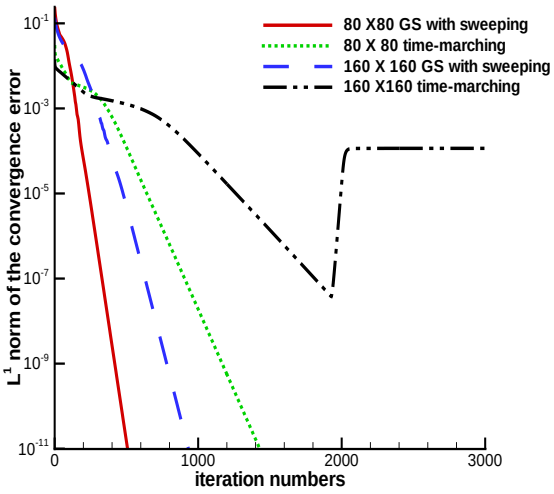

FIGURE 4. Example 4. Comparison of convergence history between the GS fixed-point fast sweeping method and the direct time-marching method for meshes of $N = 80 \times 80$ and $N = 160 \times 160$.

fixed-point fast sweeping method can accelerate the convergence to steady state further, with a suitable relaxation parameter ω. We observe the improvement of convergence in all numerical examples, and acceleration by the SOR approach is especially significant for the one-dimensional problems. We also observe that the optimal ω value is problem-dependent, which is consistent with the SOR method for solving linear systems [7]. How to choose the optimal ω in SOR for a general problem is still an open problem. However, in numerical experiments we observe that the optimal value of ω is *not* sensitive to the computational mesh sizes. This provides a method on how to apply SOR to accelerate computations. We can first test SOR on a coarse mesh using different ω values and find an optimal ω value. Then this optimal ω value can be applied to more refined meshes to accelerate computations. This general approach of applying Gauss-Seidel and alternating sweeping strategy to accelerate convergence to steady state solutions can be directly applied to other time-marching schemes, other numerical fluxes and the system cases. This consists of our future research.

References

[1] M. Boué and P. Dupuis, Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control, SIAM Journal on Numerical Analysis, 36 (1999), 667–695.

[2] W. Chen, C.-S. Chou and C.-Y. Kao, Lax-Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws, Journal of Computational Physics, 234 (2012), 452-471.

[3] C.-S. Chou and C.-W. Shu, High order residual distribution conservative finite difference WENO schemes for steady state problems on non-smooth meshes, Journal of Computational Physics, 214 (2006), 698-724.

[4] B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, Journal of Scientific Computing, 16 (2001), 173–261.

[5] G.-S. Jiang and D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, SIAM Journal on Scientific Computing, 21 (2000), 2126–2143.
[6] C.Y. Kao, S. Osher and J. Qian, Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations, Journal of Computational Physics, 196 (2004), 367–391.
[7] C.T. Kelley, Iterative methods for linear and nonlinear equations. SIAM, 1995.
[8] S. Luo, S. Leung, and J. Qian, An Adjoint State Method for Numerical Approximation of Continuous Traffic Congestion Equilibria, Communications in Computational Physics, 10 (2011), 1113-1131.
[9] J. Qian, Y.-T. Zhang and H.-K. Zhao, Fast sweeping methods for Eikonal equations on triangular meshes, SIAM Journal on Numerical Analysis, 45 (2007), 83–107.
[10] C.-W. Shu,Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In A. Quarteroni, editor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, pages 325-432. Lecture Notes in Mathematics, v 1697, Springer, 1998.
[11] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, Journal of Computational Physics, 77 (1988), 439-471.
[12] Y.-H. R. Tsai, L.-T. Cheng, S. Osher and H.-K. Zhao, Fast sweeping algorithms for a class of Hamilton-Jacobi equations, SIAM Journal on Numerical Analysis, 41 (2003), 673–694.
[13] J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Transactions on Automatic Control, 40 (1995), 1528–1538.
[14] T. Xiong, M. Zhang, Y.-T. Zhang and C.-W. Shu, Fifth order fast sweeping WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment, Journal of Scientific Computing, 45 (2010), 514–536.
[15] S. Zhang and C.-W. Shu, A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions, Journal of Scientific Computing, 31 (2007), 273–305.
[16] Y.-T. Zhang, H.-K. Zhao and S. Chen, Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations, Methods and Applications of Analysis, 13 (2006), 299–320.
[17] Y.-T. Zhang, H.-K. Zhao and J. Qian, High order fast sweeping methods for static Hamilton-Jacobi equations, Journal of Scientific Computing, 29 (2006), 25–56.
[18] H.-K. Zhao, A fast sweeping method for Eikonal equations, Mathematics of Computation, 74 (2005), 603–627.

Department of Mathematical Sciences, Indiana University South Bend, South Bend, IN 46634, USA

E-mail: chen39@iusb.edu