MONOTONICITY PRESERVING AND TOTAL VARIATION DIMINISHING MULTIGRID TIME STEPPING METHODS

ANTONY JAMESON * AND JUSTIN WAN (APRIL, 2001) †

Abstract. We propose a fast multiplicative and additive multigrid time stepping schemes for solving linear and nonlinear wave equations in one dimension. The idea is based on an upwind biased interpolation and residual restriction operators, and a nonstandard coarse grid update formula for linear equations. We prove that the two-level schemes preserve monotonicity and are total variation diminishing, and the same results hold for the multilevel additive scheme. We generalize the idea to nonlinear equations by solving local Riemann problems. We demonstrate numerically that these schemes are essentially nonoscillatory, and that the optimal speed of wave propagation of $2^M - 1$ is achieved, where M is the number of grids.

Key words. monotonicity preserving, total variation diminishing, multigrid time stepping

AMS subject classifications. 65M12, 65M25, 65M55, 65F10

1. Introduction.

Multigrid has shown to be a powerful and one of the most efficient numerical techniques for solving elliptic partial differential equations (PDEs) [4, 5, 14, 38]; its convergence rate is often independent of the mesh size. Well established convergence theory [2, 14, 40] and sophisticated smoothing [4, 35, 38, 39], coarsening [7, 8, 34, 36], and interpolation [1, 6, 31, 32, 37, 41] techniques have been developed. However, the fundamental study of multigrid for hyperbolic equations is less well-developed. One major difficulty is that the discretization matrices of hyperbolic equations are in general nonsymmetric, and hence the smoothing property of relaxation methods, and the minimization property of Galerkin coarse grid correction, both of which are essentially due to symmetry and positive definiteness, may not hold anymore for hyperbolic equations. If the multigrid principle of reducing the high and low frequency errors reflects the smoothing nature of elliptic operators, then the intrinsic wave propagation nature of hyperbolic equations must also be reflected in the hyperbolic multigrid methods. The objective of this paper is to study and analyze the wave propagation property of specially designed monotonicity preserving multigrid time stepping methods.

Multigrid methods for hyperbolic equations, in particular, steady Euler equations, are first proposed by Ni [30] and Jameson [19]. In their approach, the key is to accelerate wave propagation on multiple grids since larger time steps can be taken on coarse grids without violating the CFL condition. Thus, low frequency disturbances are rapidly expelled through the outer boundary whereas high frequency errors are locally damped. Multigrid time stepping schemes exploiting this effect have also been proposed in [15,

* Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305-4035. Email: jameson@baboon.stanford.edu.

† Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Email: jwlwan@bryce1.uwaterloo.ca. This author has been partially supported by the Natural Sciences and Engineering Research Council of Canada, and the NSF under Contract ACR-0072112.

16]. Other references using a similar approach can be found in the survey paper by Hemker and Johnson [17].

In the approach of Ni [30], Lax-Wendroff type time stepping as smoother is used which is not dissipative. Thus, artificial viscosity is introduced to damp the high frequencies. Intergrid transfer operator is the standard linear interpolation. In the approach of Jameson [19], multistage Runge Kutta type time stepping [20] is used in conjunction with volume-weighted restriction and linear interpolation. The Runge-Kutta coefficients are selected to optimize the local damping and hence artificial viscosity is not necessary. For both approaches, it can be verified numerically [22] and proved theoretically [13, 27] that the multigrid time stepping scheme on M-grid is consistent and first order accurate, with effective time step

$$\Delta t = \sum_{k}^{M} \Delta t_k,$$

where Δt_k is the time step taken on grid k.

Further algorithmic improvements have been developed. In [23], an upwind prolongation is used, and the restriction essentially is the adjoint of the prolongation. A similar prolongation and restriction technique based on characteristics is use in [24] for unstructured grid computations. Upwind prolongation based on MUSCL reconstruction is used in [12] for solving the Navier-Stokes equations. Residual dependent restriction is used in [11]. Different relaxation smoothings are compared in [29] to solve the one-dimensional Burgers' equation. Multiple semi-coarsening is used in [28] to handle anisotropic PDE coefficients arising from the alignment of flow with the grid. Higher order transfer operators are considered in [9] to maintain stability of the coarse grid problem.

In this paper, the numerical analysis of two schemes proposed by Jameson is studied and extended for the steady state solution of one-dimensional scalar linear and nonlinear wave equations. The primary focus is on the monotonicity preserving and total variation diminishing (TVD) properties which have not been studied in the context of multigrid. Although the main purpose is to rapidly expell the low frequency disturbances out of the boundary, it turns out that numerical oscillations can delay the propagation substantially. As an example, we apply the standard multigrid with Jacobi smoothing, linear interpolation, and Galerkin coarse grid correction to solve the one-dimensional linear wave equation. The initial condition (disturbance) is a square wave. The numerical solution at the subsequent multigrid cycles are shown in Figure 1. On a grid of 128 grid points, a single-grid method requires 128 time step to propagate the square wave out of the boundary on the right. The 3-level multigrid method still requires more than 60 time steps (or iterations) to convergence due to the numerical oscillations generated at the tail of the square wave. Thus, it is imperative to design multigrid algorithms which preserve monotonicity and are TVD.

In Section 2, we describe an upwind biased residual restriction and interpolation operators for solving the linear wave equation, and in Section 3, their generalization to the nonlinear inviscid Burgers' equation. In Section 4, we present analysis on the monotonicity preserving and TVD properties of the proposed multigrid schemes in the case of two-level as well as multilevel. Finally, numerical results are presented in Section 5 to demonstrate their effectiveness of accelerating wave propagation on multiple grids.

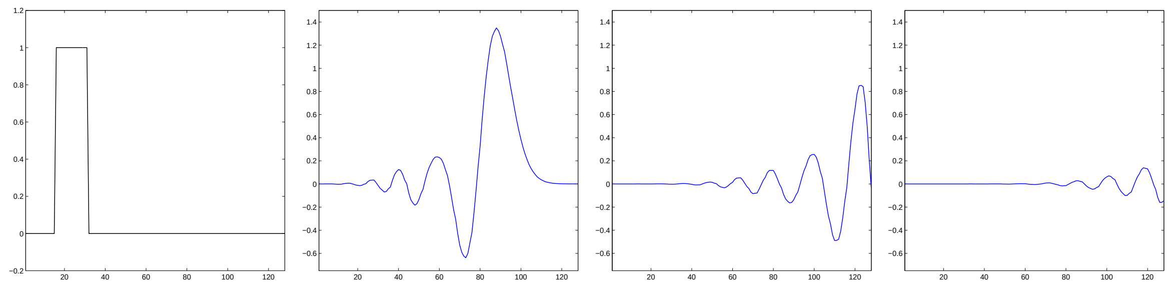

FIG. 1. *The numerical solution given by the standard multigrid at (a) time step = 0, (b) time step = 20, (c) time step = 40, (d) time step = 60.*

2. Linear wave equation. We consider the linear wave equation in one dimension:

$$
\begin{aligned}
u_t + u_x &= 0 \qquad\qquad 0 < x < 1, \\
u(0,t) &= 0, \quad u(x,0) = u_0(x).
\end{aligned}
\tag{1}
$$

We are interested in the steady state solution which, in this case, is $u \equiv 0$. We discretize the equations by the standard first order upwind scheme:

$$u_j^{n+1} = u_j^n - \lambda(u_j^n - u_{j-1}^n),$$

where $\lambda = \Delta t^h / \Delta x^h$ is the CFL number. If a single grid with N grid points is used, it will take

$$\frac{1}{\Delta t^h} = \frac{1}{\lambda} N$$

time steps to march to the steady state. The objective is to accelerate propagation on multiple grids while preserving monotonicity and being TVD for $\lambda \leq 1$.

We first describe a standard two-level algorithm for solving hyperbolic equations. Here are some notations. Given a fine grid $\{x_j^h\}$, $j = 0, 1, 2, \ldots, N$, the grid points with even indices are selected as coarse grid points $\{x_j^H\}$, $j = 0, 2, 4, \ldots, N$. The superscripts h and H denote functions on the fine and coarse grids, respectively, whereas the subscript j denotes the corresponding jth grid point.

In a standard two-level algorithm (FAS-cycle for nonlinear equations) [4, 17], we start with the current approximation u^n. The fine grid evolution consists of one step of upwinding:

$$\bar{u}_j^h = u_j^n - \lambda(u_j^n - u_{j-1}^n). \tag{2}$$

In our approach, we focus on fast propagation to reach the steady state and hence it is not important whether the upwinding step will damp the high frequencies (indeed it does for $0 < \lambda < 1$). Let $\mathcal{R}_u$ and $\mathcal{R}_r$ be the restriction operators for the solution and the residual, respectively. On the coarse grid, we compute the coarse grid approximation as follows:

$$\begin{array}{ll} \text{fine grid residual:} & \bar{r}_j^h = (\bar{u}_j - \bar{u}_{j-1}^h)/\Delta x^h \\ \text{restriction of } \bar{u}^h\text{:} & u_{2j}^H = \mathcal{R}_u \bar{u}_{2j}^h \\ \text{coarse grid RHS:} & f_{2j}^H = (u_{2j}^H - u_{2j-2}^H)/\Delta x^H - \mathcal{R}_r \bar{r}_{2j}^h \\ \text{coarse grid evolution:} & \bar{u}_{2j}^H = u_{2j}^H - \Delta t^H((u_{2j}^H - u_{2j-2}^H)/\Delta x^H - f_{2j}^H). \\ & \phantom{\bar{u}_{2j}^H} = u_{2j}^H - \lambda(u_{2j}^H - u_{2j-2}^H) + \Delta t^H f_{2j}^H. \end{array}$$

Finally, we interpolate the coarse grid correction to the fine grid and obtain the new iterate:

$$u^{n+1} = \bar{u}^h + \mathcal{P}(\bar{u}^H - \mathcal{R}_u \bar{u}^h), \tag{3}$$

where $\mathcal{P}$ is an interpolation operator. Thus, the crucial choices of the restriction and interpolation operators $\mathcal{R}_u$, $\mathcal{R}_r$, $\mathcal{P}$ will determine the effectiveness of the resulting multigrid method. As we shall see in the next sections, standard restriction and interpolation may lead to oscillations. We shall describe how we obtain monotonicity preserving and TVD multigrid by the use of upwind biased restriction and interpolation.

It is interesting to note [26] that, as opposed to elliptic multigrid, the restriction operators $\mathcal{R}_u$ and $\mathcal{R}_r$ should not be identical. In fact, the use of identical operators can lead to instability unless λ is small enough.

LEMMA 2.1. *The coarse grid evolution will be unstable if* $\mathcal{R}_u = \mathcal{R}_r$ *unless* $0 \le \lambda \le \frac{1}{2}$.

Proof. Consider the coarse grid evolution:

$$\begin{aligned} \bar{u}_j^H &= u_j^H - \lambda(u_j^H - u_{j-2}^H) + \Delta t^H f_j^H \\ &= u_j^H - \lambda(u_j^H - u_{j-2}^H) + \lambda(u_j^H - u_{j-2}^H) - \Delta t^H \mathcal{R}_r \bar{r}_j^h \\ &= \mathcal{R}_u \bar{u}_j^h - \Delta t^H \mathcal{R}_r \left(\frac{1}{\Delta x^h}(\bar{u}_j^h - \bar{u}_{j-1}^h)\right) \\ &= \mathcal{R}_r[\bar{u}_j^h - 2\lambda(\bar{u}_j^h - \bar{u}_{j-1}^h)]. \end{aligned}$$

Thus, $\bar{u}^H$ is essentially the restriction of the evolution of $\bar{u}^h$ with CFL number 2λ which is stable if and only if $0 \le \lambda \le \frac{1}{2}$. □

Hence, to allow large time steps, we must choose $\mathcal{R}_u$ different from $\mathcal{R}_r$. For the restriction of solution, we use injection, i.e.

$$(\mathcal{R}_u \bar{u}^h)_{2j} = \bar{u}_{2j}^h.$$

In the next section, we describe an appropriate residual restriction and interpolation.

2.1. Upwind Restriction and Interpolation. To achieve fast wave propagation without generating oscillations, Jameson proposed an upwind-biased residual restriction and interpolation. To define a conservative restriction operator, he used simple averaging restriction. There are two possible options:

(i) $\mathcal{R}_r \bar{r}_j^h = \frac{1}{2}(\bar{r}_j^h + \bar{r}_{j-1}^h)$, (ii) $\mathcal{R}_r \bar{r}_j^h = \frac{1}{2}(\bar{r}_j^h + \bar{r}_{j+1}^h)$.

Since the characteristics is from left to right, we select (i) as the residual restriction which collects information from behind; see Figure 2(a). This idea is essentially the same as the upwind schemes.

Similarly, we also have two possibilities for interpolating a coarse grid function v^H:

(i) $(\mathcal{P}v^H)_{2i} = (\mathcal{P}v^H)_{2i+1} = v_{2i}^H$. (ii) $(\mathcal{P}v^H)_{2i} = (\mathcal{P}v^H)_{2i-1} = v_{2i}^H$.

Again, based on the characteristics, we use the interpolation in (i) to predict information to the right; see Figure 2(b).

Interpolation based on characteristics have been used in [18, 24]. In [18], their restriction and interpolation are basically transpose of each other; ours are not. In elliptic multigrid, it is typical that $\mathcal{R}_r = \mathcal{P}^T$ since the resulting multigrid method will be symmetric. However, since the discretization matrix of hyperbolic equations is not symmetric in general, such constraint is unnecessary. In [24], their restriction and interpolation are basically the same as ours for the linear wave equation. However, our generalization to nonlinear equations is very different. Furthermore, we propose a new update formula for the coarse grid correction, which eliminates another source of numerical oscillations, as described in Section 2.3.

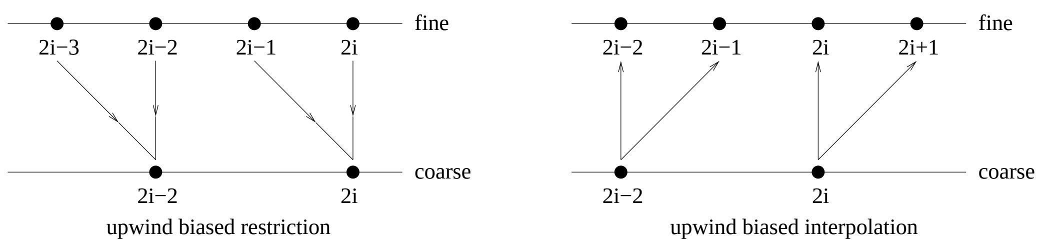

FIG. 2. *(a) Upwind biased restriction (b) Upwind biased interpolation.*

2.2. Higher order interpolation. In elliptic multigrid, linear interpolation is often used. It turns out that for the linear wave equation, the use of linear interpolation will, in fact, lead to oscillations. In Figure 3, we show the discrete functions u^n, $\bar{u}^h$, etc, as defined in Section 2. The current approximation u^n is defined as:

$$u_j^n = \begin{cases} 1 & j = 0, 1, 2. \\ 0 & j = 3, 4, \ldots, 8. \end{cases}$$

The functions $\bar{u}^h$, u^H, and $\bar{u}^H$ are computed based on the two-level algorithm with $\lambda = 1$. In the final step, we see that overshoot occurs at u_3^{n+1}: by (3),

$$\begin{aligned} u_3^{n+1} &= \bar{u}_3^h + \mathcal{P}_{\text{linear}}(\bar{u}^H - u^H)_3 \\ &= \bar{u}_3^h + \frac{\bar{u}_2^H + \bar{u}_4^H}{2} - \frac{u_2^H + u_4^H}{2} = 1 + 1 - \frac{1}{2} = \frac{3}{2}. \end{aligned}$$

As for comparisons, if the upwind biased interpolation is used, then

$$\begin{aligned} u_3^{n+1} &= \bar{u}_3^h + \mathcal{P}_{\text{upwind}}(\bar{u}^{n+1} - u^H)_3 \\ &= \bar{u}_3^h + \bar{u}_2^H - u_2^H = 1. \end{aligned}$$

See also Figure 4.

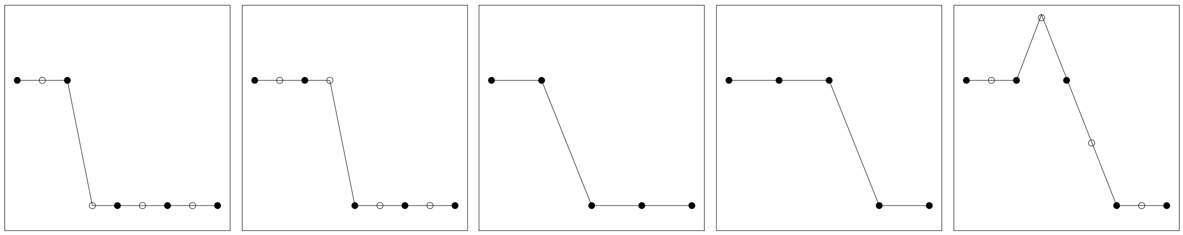

FIG. 3. *Oscillation caused by linear interpolation. From left to right:* u^n, $\bar{u}^h$, u^H, $\bar{u}^H$, *and* u^{n+1}. *The black dots denote coarse grid points.*

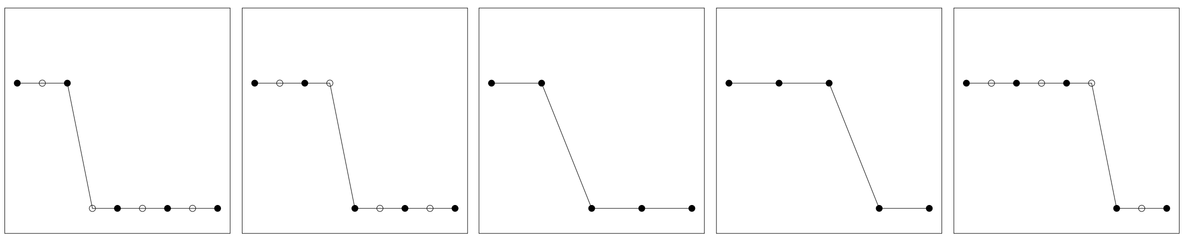

FIG. 4. *No oscillation using upwind biased interpolation. From left to right:* u^n, $\bar{u}^h$, u^H, $\bar{u}^H$, *and* u^{n+1}.

2.3. Coarse grid update. Unfortunately, we note that even with the use of upwind restriction and interpolation, oscillations can still occur. In Figure 5, we show a similar sequence of pictures as in Figure 3 with slightly different starting data u^n where the discontinuity is shifted by one grid point. We see that this time, undershoot occurs at u_3^{n+1}. Our fix is to replace the standard update formula by

$$u^{n+1} = \bar{u}^h + P(\bar{u}^H - \mathcal{R}_u u^n).$$

The idea is to compute the coarse grid error by the difference of the coarse grid evolved function $\bar{u}^H$ and the restriction of the original function u^n, instead of $\bar{u}^h$. As shown in Figure 6, the oscillation is eliminated. In fact, we can justify theoretically that oscillations will not occur in Section 4.

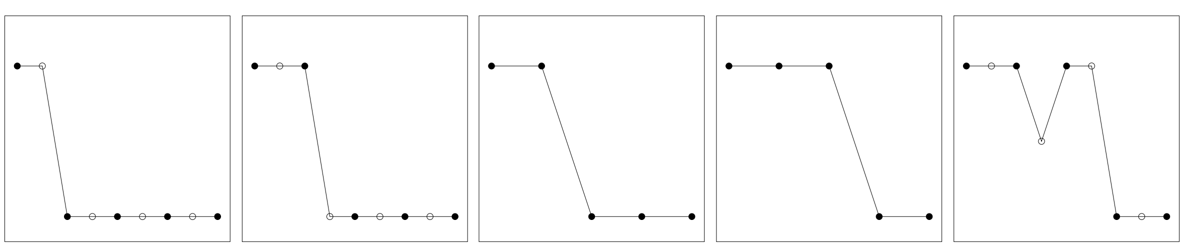

FIG. 5. *Oscillation caused by standard update. From left to right:* u^n, $\bar{u}^h$, u^H, $\bar{u}^H$, *and* u^{n+1}.

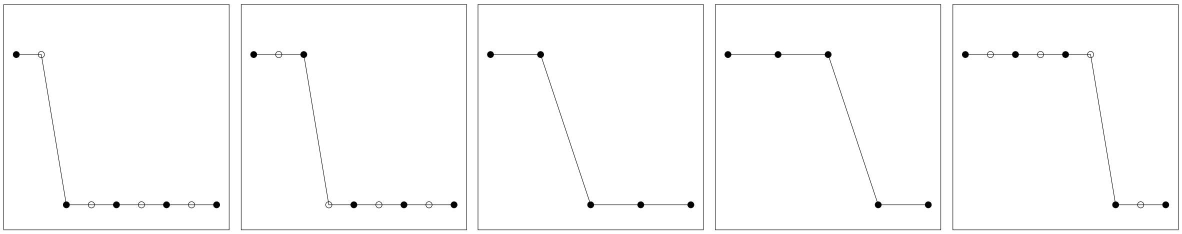

FIG. 6. *No oscillation using the new coarse grid update formula. From left to right:* u^n, $\bar{u}^h$, u^H, $\bar{u}^H$, *and* u^{n+1}.

2.4. Algorithms. We note that f_j^H can be further simplified in the linear wave problem:

$$\begin{aligned} f_j^H &= \frac{1}{\Delta x^H}(u_j^H - u_{j-2}^H) - \frac{1}{2}(\bar{r}_j^h + \bar{r}_{j-1}^h) \\ &= \frac{1}{2\Delta x^h}(\bar{u}_j^h - \bar{u}_{j-2}^h) - \frac{1}{2\Delta x^h}[(\bar{u}_j^h - \bar{u}_{j-1}^h) + (\bar{u}_{j-1}^h - \bar{u}_{j-2}^h)] \\ &= 0. \end{aligned}$$

The final algorithm of the two-level multigrid time stepping is:

Algorithm: Multiplicative Scheme (two-grid)

$$\begin{aligned} \bar{u}_j^h &= u_j^n - \lambda(u_j^n - u_{j-1}^n) \\ &\left.\begin{aligned} \bar{u}_j^H &= \bar{u}_j^h - \lambda(\bar{u}_j^h - \bar{u}_{j-2}^h) \\ \tilde{u}_j^h &= \bar{u}_j^H \\ \tilde{u}_{j-1}^h &= \bar{u}_{j-1}^h + \bar{u}_{j-2}^H - u_{j-2}^n \end{aligned}\right\} j = 0, 2, 4, \ldots \\ u_j^{n+1} &= \tilde{u}_j^h \end{aligned}$$

Since the coarse grid uses the most update information ($\bar{u}^h$) on the fine grid, this approach is also commonly known as the multiplicative scheme in the multigrid literature.

Another possibility is that we restrict and propagate u^n on all the coarse grids; thus all the computations can be carried out in parallel. Since this approach is similar in idea to the additive Schwarz method [33] in domain decomposition and the BPX algorithm [3], we call this the additive multigrid time stepping:

Algorithm: Additive Scheme (two-grid)

$$\begin{aligned} \bar{u}_j^h &= u_j^n - \lambda(u_j^n - u_{j-1}^n) \\ &\left.\begin{aligned} \bar{u}_j^H &= u_j^n - \lambda(u_j^n - u_{j-2}^n) \\ \tilde{u}_j^h &= \bar{u}_j^H \\ \tilde{u}_{j-1}^h &= \bar{u}_{j-1}^h + \bar{u}_{j-2}^H - u_{j-2}^n \end{aligned}\right\} j = 0, 2, 4, \ldots \\ u_j^{n+1} &= \tilde{u}_j^h \end{aligned}$$

Remark: The wave propagation by the multiplicative scheme is generally twice as fast as the additive scheme (cf. Gauss-Seidel vs Jacobi).

The multilevel scheme is straightforward; apply the two-level scheme recursively on the coarser grids. We denote functions on the first grid, which is also the finest grid, by superscript (1); the second grid by (2), and so on. The multilevel multiplicative and additive algorithms are as follows:

Algorithm: Multiplicative Scheme (multi-grid)

$u^{(1)} = u^n$

Define $\tilde{u}^{(k)}$=MG($u^{(k)}$) by:

if $k = L$,

$$\tilde{u}_j^{(k)} = u_j^{(k)} - \lambda(u_j^{(k)} - u_{j-2^{k-1}}^{(k)}) \qquad j = 0, 2^{k-1}, 2\cdot 2^{k-1}, \ldots.$$

else

$$\bar{u}_j^{(k)} = u_j^{(k)} - \lambda(u_j^{(k)} - u_{j-2^{k-1}}^{(k)}) \qquad j = 0, 2^{k-1}, 2\cdot 2^{k-1}, \ldots.$$

$$u_j^{(k+1)} = \bar{u}_j^{(k)} \qquad j = 0, 2^k, 2\cdot 2^k, \ldots.$$

$$\tilde{u}^{(k+1)} = \mathrm{MG}(u^{(k+1)})$$

$$\tilde{u}_j^{(k)} = \tilde{u}_j^{(k+1)} \qquad j = 0, 2^k, 2\cdot 2^k, \ldots.$$

$$\tilde{u}_{j-2^{k-1}}^{(k)} = \bar{u}_{j-2^{k-1}}^{(k)} + \tilde{u}_{j-2^k}^{(k+1)} - u_{j-2^k}^{(k)} \qquad j = 0, 2^k, 2\cdot 2^k, \ldots.$$

end

u^{n+1}=MG($u^{(1)}$)

Algorithm: Additive Scheme (multi-grid)

$u^{(1)} = u^n$

Define $\tilde{u}^{(k)}$=MG($u^{(k)}$) by:

if $k = L$,

$$\tilde{u}_j^{(k)} = u_j^n - \lambda(u_j^n - u_{j-2^{k-1}}^n) \qquad j = 0, 2^{k-1}, 2\cdot 2^{k-1}, \ldots.$$

else

$$\bar{u}_j^{(k)} = u_j^n - \lambda(u_j^n - u_{j-2^{k-1}}^n) \qquad j = 0, 2^{k-1}, 2\cdot 2^{k-1}, \ldots.$$

$$u_j^{(k+1)} = u_j^{(k)} \qquad j = 0, 2^k, 2\cdot 2^k, \ldots.$$

$$\tilde{u}^{(k+1)} = \mathrm{MG}(u^n,\ k+1)$$

$$\tilde{u}_j^{(k)} = \tilde{u}_j^{(k+1)} \qquad j = 0, 2^k, 2\cdot 2^k, \ldots.$$

$$\tilde{u}_{j-2^{k-1}}^{(k)} = \bar{u}_{j-2^{k-1}}^{(k)} + \tilde{u}_{j-2^k}^{(k+1)} - u_{j-2^k}^n \quad j = 0, 2^k, 2\cdot 2^k, \ldots.$$

end

u^{n+1}=MG($u^{(1)}$)

3. Nonlinear wave equation. In this section, we generalize the methodology for linear wave equations to nonlinear equations. We consider the model inviscid Burgers' equation in one dimension:

$$u_t + (\frac{1}{2}u^2)_x = 0 \qquad 0 < x < 1, \tag{4}$$

with appropriate boundary and initial conditions. Again, we are interested in obtaining the steady state solution fast.

We discretize the equations by the EO scheme [10]:

$$u_j^{n+1} = u_j^n - \lambda(F^{EO}(u_j^n, u_{j+1}^n) - F^{EO}(u_{j-1}^n, u_j^n)). \tag{5}$$

The numerical flux F^{EO} is defined as

$$F^{EO}(u_L, u_R) = \frac{1}{2}(u_L^+)^2 + \frac{1}{2}(u_R^-)^2,$$

where $u^+ \equiv \max(u, 0)$ and $u^- \equiv \min(u, 0)$. We note that the first order EO scheme is used for illustration purpose only; other schemes such as Godunov or more sophisticated high order schemes [25] can be used as well.

The basic format of the multigrid time stepping schemes, either multiplicative or additive, is essentially unchanged even in the nonlinear case; we smooth or propagate the wave on the fine grid, and accelerate the propagation on the coarse grids. We need, however, to make several modifications. For instance, the upwinding smoothing (2) is now substituted by the EO smoothing (5). The restriction and interpolation require more detail explanation which are described in the following.

3.1. Nonlinear Upwind Restriction and Interpolation. In the linear case, the characteristics are constantly from left to right at each grid point. For the nonlinear Burgers' equation, however, the characteristic directions changes from grid points to grid points. In addition, shocks and rarefaction waves can occur anywhere. Hence, we need to determine whether the coarse grid point should interpolate (restrict) to the left or right, and more importantly, how to handle shocks and rarefactions. Since restriction and interpolation essentially are based on the same principle, we shall describe the interpolation only.

Consider a two-level method. As shown in Figure 7, given the coarse grid values u_i^H and u_{i+1}^H at the coarse grid points x_i^H ($=x_{2i}^h$) and x_{i+1}^H ($=x_{2i+2}^h$), respectively, which value should we select at the fine grid point x_{2i+1}^h? If both values are positive (negative), the wave propagates to the right (left) locally and it resembles the linear case. Thus, we simply take u_i^H (u_{i+1}^H) for the value at x_{2i+1}^h. When u_i^H is positive and u_{i+1}^H is negative, i.e. a shock, we have information coming from both sides. Now, which one should we take? In the opposite case, when u_i^H is negative and u_{i+1}^H is positive, rarefaction occurs. This situation seems even worse since the information is now going away.

Our idea is based on characteristics and is motivated by the following key observation. We consider the problem shown in Figure 7 as a local two-point boundary value problem. For the linear wave equation (one boundary value is in fact redundant), it can be written as:

$$\begin{aligned} u_t + u_x &= 0 \qquad\qquad x_i^H < x < x_{i+1}^H \\ u(x_i^H, t) &= u_i^H \\ u(x, 0) &= \bar{u}^h. \end{aligned}$$

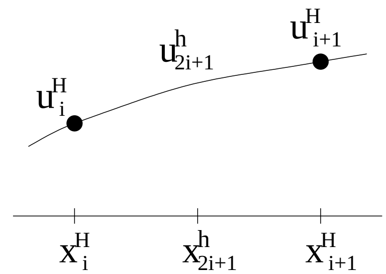

FIG. 7. *The interpolation value at the noncoarse grid point x^h_{2i+1} is given by the solution of a local PDE problem for the linear wave equation, and the Riemann solution for the Burgers' equation.*

We then define the interpolation value at x^h_{2i+1} as the solution of the local two-point boundary value problem. Such interpolation method turns out to be exactly the same as the upwind biased interpolation described in Section 2.1. In other word, we can view the upwind biased interpolation as solving local boundary value problems.

Generalizing this idea to Burgers' equation, we solve a local Riemann problem instead:

$$\begin{aligned} u_t + (\frac{1}{2}u^2)_x &= 0 \\ u(x,0) &= \begin{cases} u^H_i & x^H_i < x < x^h_{2i+1} \\ u^H_{i+1} & x^h_{2i+1} < x < x^H_{i+1}. \end{cases} \end{aligned}$$

Then, the interpolation value at x^h_{2i+1} is given by the Riemann solution. More precisely, if $u^H_i > u^H_{i+1}$, a shock occurs and its speed, $s = \frac{u^H_i + u^H_{i+1}}{2}$. If $s \geq 0$, then $u^h_{2i+1} = u^H_i$; if $s < 0$, then $u^h_{2i+1} = u^H_{i+1}$. If $u^H_i < u^H_{i+1}$, it is a rarefaction wave. If they are of the same sign, then $u^h_{2i+1} = u^H_i$ if they are positive, and $u^h_{2i+1} = u^H_{i+1}$ if they are negative. Finally, if $u^H_i < 0 < u^H_{i+1}$, then the rarefaction wave turns out to be zero at x^h_{2i+1} and hence $u^h_{2i+1} = 0$. Thus, the interpolation is completely determined by the local Riemann solutions. We note that the same interpolation would be obtained if we use an argument based on characteristics.

Applying the Riemann solutions to the multilevel multiplicative scheme, the interpolation value $\tilde{u}^{(k)}_{j-2^{k-1}}$ is defined as:

if $\bar{u}^{(k)}_{j-2^k} \geq 0$, $\bar{u}^{(k)}_j \geq 0$,

$$\tilde{u}^{(k)}_{j-2^{k-1}} = \bar{u}^{(k)}_{j-2^{k-1}} + \bar{u}^{(k+1)}_{j-2^k} - u^{(k+1)}_{j-2^k}$$

if $\bar{u}^{(k)}_{j-2^k} \leq 0$, $\bar{u}^{(k)}_j \leq 0$,

$$\tilde{u}^{(k)}_{j-2^{k-1}} = \bar{u}^{(k)}_{j-2^{k-1}} + \bar{u}^{(k+1)}_{j} - u^{(k+1)}_{j}$$

if $\bar{u}^{(k)}_{j-2^{k-1}} \geq 0 \geq \bar{u}^{(k)}_j$,

$$\tilde{u}^{(k)}_{j-2^{k-1}} = \begin{cases} \bar{u}^{(k)}_{j-2^{k-1}} + \bar{u}^{(k+1)}_{j-2^k} - u^{(k+1)}_{j-2^k} & \text{if } \bar{u}^{(k)}_{j-2^{k-1}} \geq 0 \\ \bar{u}^{(k)}_{j-2^{k-1}} + \bar{u}^{(k+1)}_{j} - u^{(k+1)}_{j} & \text{if } \bar{u}^{(k)}_{j-2^{k-1}} < 0 \end{cases}$$

if $\bar{u}^{(k)}_{j-2^k} < 0 < \bar{u}^{(k)}_j$,

$$\tilde{u}^{(k)}_{j-2^{k-1}} = \bar{u}^{(k)}_{j-2^{k-1}}.$$

For the additive scheme, the interpolation is defined similarly.

We remark that constructing interpolation by solving local linear boundary value problem has been used in multigrid methods for elliptic and convection-diffusion equations [14, 38]. Solving local Riemann problem has been extensively used in designing numerical schemes (e.g. Godunov schemes) for conservation laws [25]; however, it has not been used in the context of multigrid interpolation to the best of the authors' knowledge.

4. Analysis. In our construction of interpolation, restriction, and coarse grid update, the primary focus is to minimize oscillations. In this section, we justify this theoretically for the linear wave equation. Here, we consider the two multigrid methods as time stepping schemes, and each multigrid cycle as a time stepping. Like discretization schemes for conservation laws, one important measure of nonoscillatory schemes is whether it preserves monotonicity. More precisely, if u^n is a nonincreasing (nondecreasing) function, after one time stepping (in our case, one multigrid cycle), u^{n+1} must remain nonincreasing (nondecreasing). Another important measure is total variation diminishing (TVD), which requires that the total variation of u^{n+1} does not exceed that of u^n. Both concepts are fundamental to designing numerical schemes for conservation laws, but nevertheless, have never been used to analyze multigrid methods.

Hence, the primary focus of our analysis is on the monotonicity and total variation diminishing properties of the proposed schemes. In particular, we prove that both the two-level multiplicative and additive schemes preserve monotonicity and are TVD; and the same holds for multilevel additive scheme. These results are essentially due to the use of the upwind biased restriction and interpolation operators, which in turn leads to fast propagation on multiple grids. We first consider the two-level case.

THEOREM 4.1 (JAMESON). *Both two-level multiplicative and additive multigrid time stepping schemes preserve monotonicity.*

Proof. For j even, we have

$$\begin{aligned} u^{n+1}_j &= \tilde{u}^h_j = \bar{u}^H_j = \bar{u}^h_j - \lambda(\bar{u}^h_j - \bar{u}^h_{j-2}), \\ u^{n+1}_{j-1} &= \tilde{u}^h_{j-1} = \bar{u}^h_{j-1} + \bar{u}^H_{j-2} - u^n_{j-2} \\ &= \bar{u}^h_{j-1} + \bar{u}^h_{j-2} - \lambda(\bar{u}^h_{j-2} - \bar{u}^h_{j-4}) - u^n_{j-2} \\ &= \bar{u}^h_{j-1} - \lambda(u^n_{j-2} - u^n_{j-3}) - \lambda(\bar{u}^h_{j-2} - \bar{u}^h_{j-4}). \end{aligned}$$

Subtracting the two,

$$\begin{aligned} (6)\quad & u^{n+1}_j - u^{n+1}_{j-1} \\ =\; & (1-\lambda)(\bar{u}^h_j - \bar{u}^h_{j-1}) - \lambda(\bar{u}^h_{j-1} - \bar{u}^h_{j-2}) + \lambda(u^n_{j-2} - u^n_{j-3}) + \lambda(\bar{u}^h_{j-2} - \bar{u}^h_{j-3}) \\ & + \lambda(\bar{u}^h_{j-3} - \bar{u}^h_{j-4}) \\ =\; & (1-\lambda)^2(u^n_j - u^n_{j-1}) + \lambda(1-\lambda)(u^n_{j-1} - u^n_{j-2}) - \lambda(1-\lambda)(u^n_{j-1} - u^n_{j-2}) \end{aligned}$$

$$
\begin{aligned}
& - \lambda^2(u^n_{j-2} - u^n_{j-3}) + \lambda(u^n_{j-2} - u^n_{j-3}) + \lambda(1-\lambda)(u^n_{j-2} - u^n_{j-3}) + \lambda^2(u^n_{j-3} - u^n_{j-4}) \\
& + \lambda(1-\lambda)(u^n_{j-3} - u^n_{j-4}) + \lambda^2(u^n_{j-4} - u^n_{j-5}) \\
= \; & (1-\lambda)^2(u^n_j - u^n_{j-1}) + 2\lambda(1-\lambda)(u^n_{j-2} - u^n_{j-3}) + \lambda(u^n_{j-3} - u^n_{j-4}) \\
& + \lambda^2(u^n_{j-4} - u^n_{j-5}).
\end{aligned}
$$

Similarly, for $j-1$ being odd, we have,

$$
\begin{aligned}
u^{n+1}_{j-1} - u^{n+1}_{j-2} &= \tilde{u}^h_{j-1} - \tilde{u}^h_{j-2} = \bar{u}^h_{j-1} + \bar{u}^H_{j-2} - u^n_{j-2} - \bar{u}^H_{j-2} \\
&= \bar{u}^h_{j-1} - u^n_{j-2} \\
&= (1-\lambda)(u^n_{j-1} - u^n_{j-2}).
\end{aligned} \tag{7}
$$

For stability, it must hold that on each grid,

$$0 \le \lambda \le 1.$$

Thus, if u^n is monotone, for instance,

$$u^n_j - u^n_{j-1} \ge 0 \qquad \forall j,$$

then u^{n+1} is also monotone since

$$
\left.\begin{aligned}
u^{n+1}_j - u^{n+1}_{j-1} &\ge 0 \\
u^{n+1}_{j-1} - u^{n+1}_{j-2} &\ge 0
\end{aligned}\right\} \qquad j = 0, 2, 4, \ldots
$$

for $0 \le \lambda \le 1$.

For the additive scheme, the calculations are similar. For even j, we have

$$
\begin{aligned}
u^{n+1}_j &= \tilde{u}^h_j = \bar{u}^H_j = u^n_j - \lambda(u^n_j - u^n_{j-2}), \\
u^{n+1}_{j-1} &= \tilde{u}^h_{j-1} = \bar{u}^h_{j-1} + \bar{u}^H_{j-2} - u^n_{j-2} \\
&= u^n_{j-1} - \lambda(u^n_{j-1} - u^n_{j-2}) - \lambda(u^n_{j-2} - u^n_{j-4}).
\end{aligned}
$$

Thus

$$
\begin{aligned}
& u^{n+1}_j - u^{n+1}_{j-1} \\
= \; & u^n_j - u^n_{j-1} - \lambda(u^n_j - u^n_{j-2}) + \lambda(u^n_{j-1} - u^n_{j-2}) + \lambda(u^n_{j-2} - u^n_{j-4}) \\
= \; & (1-\lambda)(u^n_j - u^n_{j-1}) + \lambda(u^n_{j-2} - u^n_{j-3}) + \lambda(u^n_{j-3} - u^n_{j-4}),
\end{aligned} \tag{8}
$$

and

$$
\begin{aligned}
& u^{n+1}_{j-1} - u^{n+1}_{j-2} \\
= \; & u^n_{j-1} - u^n_{j-2} - \lambda(u^n_{j-1} - u^n_{j-2}) - \lambda(u^n_{j-2} - u^n_{j-4}) + \lambda(u^n_{j-2} - u^n_{j-4}) \\
= \; & (1-\lambda)(u^n_{j-1} - u^n_{j-2}).
\end{aligned} \tag{9}
$$

Therefore, monotonicity is preserved for $0 \le \lambda \le 1$. □

As a direct consequence, we can easily show that these two schemes are also TVD.

THEOREM 4.2. *The two-level multiplicative and additive multigrid time stepping schemes are TVD.*

Proof. Denote the total variation of a function u by $\mathrm{TV}(u)$. For the multiplicative scheme, using the formulae in (6) and (7),

$$\begin{aligned}
& \mathrm{TV}(u^{n+1}) \\
= & \sum_j |u_j^{n+1} - u_{j-1}^{n+1}| \\
= & \sum_{\text{even } j} |u_j^{n+1} - u_{j-1}^{n+1}| + |u_{j-1}^{n+1} - u_{j-2}^{n+1}| \\
= & \sum_{\text{even } j} |(1-\lambda)^2(u_j^n - u_{j-1}^n) + 2\lambda(1-\lambda)(u_{j-2}^n - u_{j-3}^n) + \lambda(u_{j-3}^n - u_{j-4}^n) \\
& \qquad + \lambda^2(u_{j-4}^n - u_{j-5}^n)| + |(1-\lambda)(u_{j-1}^n - u_{j-2}^n)|. \\
\leq & \sum_{\text{even } j} (1-\lambda)^2|u_j^n - u_{j-1}^n| + (1-\lambda)^2|u_{j-1}^n - u_{j-2}^n| \\
& \qquad + (1-\lambda-(1-\lambda)^2)|u_{j-1}^n - u_{j-2}^n| + 2\lambda(1-\lambda)|u_{j-2}^n - u_{j-3}^n| \\
& \qquad + \lambda|u_{j-3}^n - u_{j-4}^n| + \lambda^2|u_{j-4}^n - u_{j-5}^n| \\
= & (1-\lambda)^2\ \mathrm{TV}(u^n) + \sum_{\text{even } j} \lambda(1-\lambda)|u_{j-1}^n - u_{j-2}^n| + \lambda(1-\lambda)|u_{j-2}^n - u_{j-3}^n| \\
& + \lambda(1-\lambda)|u_{j-2}^n - u_{j-3}^n| + \lambda|u_{j-3}^n - u_{j-4}^n| + \lambda^2|u_{j-4}^n - u_{j-5}^n| \\
= & ((1-\lambda)^2 + \lambda(1-\lambda))\ \mathrm{TV}(u^n) + \sum_{\text{even } j} \lambda(1-\lambda)|u_{j-2}^n - u_{j-3}^n| \\
& + \lambda(1-\lambda)|u_{j-3}^n - u_{j-4}^n| + (\lambda - \lambda(1-\lambda))|u_{j-3}^n - u_{j-4}^n| + \lambda^2|u_{j-4}^n - u_{j-5}^n| \\
= & ((1-\lambda)^2 + 2\lambda(1-\lambda))\ \mathrm{TV}(u^n) + \sum_{\text{even } j} \lambda^2|u_{j-3}^n - u_{j-4}^n| + \lambda^2|u_{j-4}^n - u_{j-5}^n| \\
= & ((1-\lambda)^2 + 2\lambda(1-\lambda) + \lambda^2)\ \mathrm{TV}(u^n) \\
= & \mathrm{TV}(u^n)
\end{aligned}$$

Similarly, for the additive scheme, we have

$$\begin{aligned}
& \mathrm{TV}(u^{n+1}) \\
= & \sum_{\text{even } j} |u_j^{n+1} - u_{j-1}^{n+1}| + |u_{j-1}^{n+1} - u_{j-2}^{n+1}| \\
= & \sum_{\text{even } j} |(1-\lambda)(u_j^n - u_{j-1}^n) + \lambda(u_{j-2}^n - u_{j-3}^n) + \lambda(u_{j-3}^n - u_{j-4}^n)| \\
& \qquad + |(1-\lambda)(u_{j-1}^n - u_{j-2}^n)| \\
\leq & \sum_{\text{even } j} (1-\lambda)|u_j^n - u_{j-1}^n| + (1-\lambda)|u_{j-1}^n - u_{j-2}^n| + \lambda|u_{j-2}^n - u_{j-3}^n| \\
& \qquad + \lambda|u_{j-3}^n - u_{j-4}^n| \\
= & \mathrm{TV}(u^n).
\end{aligned}$$

□

For the multilevel algorithms, it turns out that the multiplicative algorithm does not preserve monotonicity in general. However, the oscillations appear to be very small and do not seem to affect the fast wave propagation; see the numerical results in Section 5. For the multilevel additive scheme, it still has the monotonicity preserving and TVD result. We first extend the formulae in (8) and (9) to the multilevel case.

LEMMA 4.3. *For the k-level additive multigrid time stepping scheme, and $j = 2^{k-1}$, it holds that*

$$\begin{aligned} u_j^{n+1} - u_{j-1}^{n+1} &= (1-\lambda)(u_j^n - u_{j-1}^n) + \lambda(u_{j-2^{k-1}}^n - u_{j-2^k}^n) \\ u_{j-m}^{n+1} - u_{j-m-1}^{n+1} &= (1-\lambda)(u_{j-m}^n - u_{j-m-1}^n) \qquad m = 1, \ldots, 2^{k-1}-1. \end{aligned}$$

Proof. We prove by induction. Suppose L is the number of multigrid levels. For $L = 2$, it is proved in the proof of Theorem 4.1. Suppose it is true for $L = k$. Now we consider the case $L = k+1$. For j=a constant multiple of 2^{k+1},

$$\begin{aligned} u_j^{n+1} &= \tilde{u}_j^{(2)} \\ u_{j-1}^{n+1} &= \bar{u}_{j-1}^{(1)} + \tilde{u}_{j-2}^{(2)} - u_{j-2}^n, \end{aligned}$$

where the superscript (1) denotes functions on the first grid, which is also the finest grid, and (2) functions on the second grid. Subtracting the two,

$$u_j^{n+1} - u_{j-1}^{n+1} = \tilde{u}_j^{(2)} - \tilde{u}_{j-2}^{(2)} - (1-\lambda)(u_{j-1}^n - u_{j-2}^n).$$

Since we use the previous values u^n on all the coarse grids, the update of $\tilde{u}_j^{(2)}$ from $u_j^{(2)}$, $j = 0, 2, \ldots$, is completely independent of the update on the finest grid. In other words, $\tilde{u}_j^{(2)}$ can be thought of being obtained by applying the level k additive scheme to u^n on the even points. By induction hypothesis, we obtain

$$\begin{aligned} u_j^{n+1} - u_{j-1}^{n+1} &= (1-\lambda)(u_j^n - u_{j-2}^n) + \lambda(u_{j-2^k}^n - u_{j-2^{k+1}}^n) - (1-\lambda)(u_{j-1}^n - u_{j-2}^n) \\ &= (1-\lambda)(u_j^n - u_{j-1}^n) + \lambda(u_{j-2^k}^n - u_{j-2^{k+1}}^n). \end{aligned}$$

Similarly, since $u_{j-2}^{n+1} = \tilde{u}_{j-2}^{(2)}$, we have

$$u_{j-1}^{n+1} - u_{j-2}^{n+1} = \bar{u}_{j-1} + \tilde{u}_{j-2}^{(2)} - u_{j-2}^n - \tilde{u}_{j-2}^{(2)} = (1-\lambda)(u_{j-1}^n - u_{j-2}^n).$$

We compute one more difference:

$$\begin{aligned} u_{j-2}^{n+1} - u_{j-3}^{n+1} &= \tilde{u}_{j-2}^{(2)} - (\bar{u}_{j-3}^{(1)} - \tilde{u}_{j-4}^{(2)} - u_{j-4}^n) \\ &= (1-\lambda)(u_{j-2}^n - u_{j-4}^n) - (1-\lambda)(u_{j-3}^n - u_{j-4}^n) \\ &= (1-\lambda)(u_{j-2}^n - u_{j-3}^n), \end{aligned}$$

by induction hypothesis on $\tilde{u}_{j-2}^{(2)} - \tilde{u}_{j-4}^{(2)}$. The rest is essentially the same and we shall omit the calculations. In conclusion, the formulae are also true for $L = k+1$. □

Now, the main result for the multilevel additive scheme follows directly from Lemma 4.3.

THEOREM 4.4. *The multilevel additive multigrid time stepping scheme preserves monotonicity and are TVD.*

Proof. The monotonicity preserving property is clear from the formulae given by Lemma 4.3. The TVD property can be seen by (assuming k-level)

$$\begin{aligned}
& \mathrm{TV}(u^{n+1}) \\
= & \sum_j |u_j^{n+1} - u_{j-1}^{n+1}| \\
= & \sum_{j=\text{multiple of } 2^{k-1}} \left(|u_j^{n+1} - u_{j-1}^{n+1}| + |u_{j-1}^{n+1} - u_{j-2}^{n+1}| + \cdots + |u_{j-2^{k-1}+1}^{n+1} - u_{j-2^{k-1}}^{n+1}| \right) \\
= & \sum_{j=\text{multiple of } 2^{k-1}} \Big[\, |(1-\lambda)(u_j^n - u_{j-1}^n) + \lambda(u_{j-2^{k-1}}^n - u_{j-2^k}^n)| \\
& \qquad + (1-\lambda)|u_{j-1}^n - u_{j-2}^n| + \cdots + (1-\lambda)|u_{j-2^{k-1}+1}^n - u_{j-2^{k-1}}^n| \, \Big] \\
\leq & (1-\lambda)\mathrm{TV}(u^n) + \lambda \mathrm{TV}(u^n) \\
= & \mathrm{TV}(u^n).
\end{aligned}$$

□

5. Numerical results. In this section, we verify numerically that the proposed schemes are nonoscillatory and the steady state solution can be reached in small number of multigrid cycles by considering the linear wave equation (1) and the nonlinear Burgers' equation (4).

For the linear wave equation, the boundary condition is chosen as 0 and the initial condition is a square wave. Therefore, the steady state solution $u \equiv 0$. We apply the 3-level multiplicative and additive schemes to solving the equation with CFL number $\lambda = 1.0$. Thus, the fine grid evolution given by the upwinding is in fact exact. The results are shown in Figure 8 and 9, respectively. First, in contrast with the standard multigrid approach (see Figure 1), the initial square wave remains as a square wave as it propagates. We remark that the multiplicative scheme in general does not preserve monotonicity and is not TVD when the number of grids is more than 2, but with the special choice of CFL number, $\lambda = 1.0$, there are no oscillations.

Secondly, the theoretically results by Jespersen [21] and Gustafsson and Lötstedt [13] estimate that the optimal speed of propagation is:

$$\text{optimal speed of multigrid} = 2^k - 1,$$

where k is the number of grids. By careful inspection, we can see that the speed of propagation of the wave given by the multiplicative scheme is precisely $2^3 - 1 = 7$ times than the speed of upwinding on a single grid, since the wave takes $128/7 \approx 18$ multigrid cycles to propagate out of the boundary. Thus, it achieves the theoretical optimal speed, while the oscillations produced by the standard multigrid method delay the convergence.

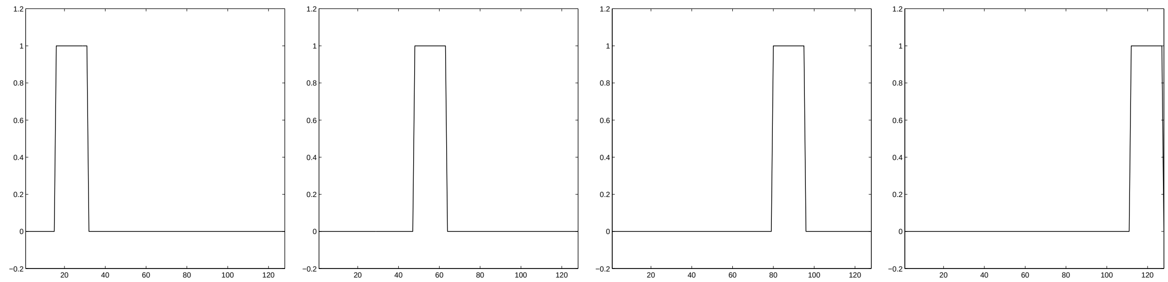

FIG. 8. *The numerical solution given by the 3-level multiplicative scheme,* $\lambda = 1.0$*, at (a) time step = 0, (b) time step = 4, (c) time step = 8, (d) time step = 12.*

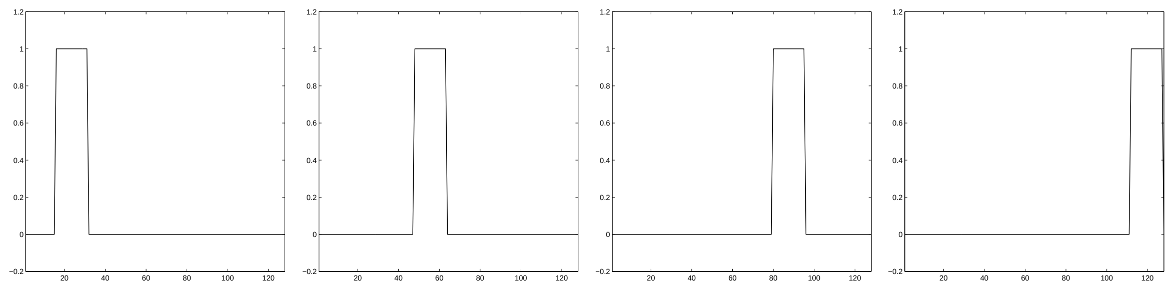

FIG. 9. *The numerical solution given by the 3-level additive scheme,* $\lambda = 1.0$*, at (a) time step = 0, (b) time step = 8, (c) time step = 16, (d) time step = 24.*

Finally, comparing the propagation speeds of the multiplicative and the additive schemes, the former is about twice as fast as the latter (cf Gauss-Seidel vs Jacobi).

When the CFL number $\neq 1.0$, e.g. $\lambda = 0.5$, the multiplicative scheme introduces oscillations. In Figure 10, second left plot, we can see a small dip on the top of the solution; i.e. the scheme does not preserve monotonicity and is not TVD. However, we note that the oscillations appear to be very minimal and do not seem to affect the fast wave propagation. The additive scheme remains nonoscillatory, as is proved in Section 4; see Figure 11.

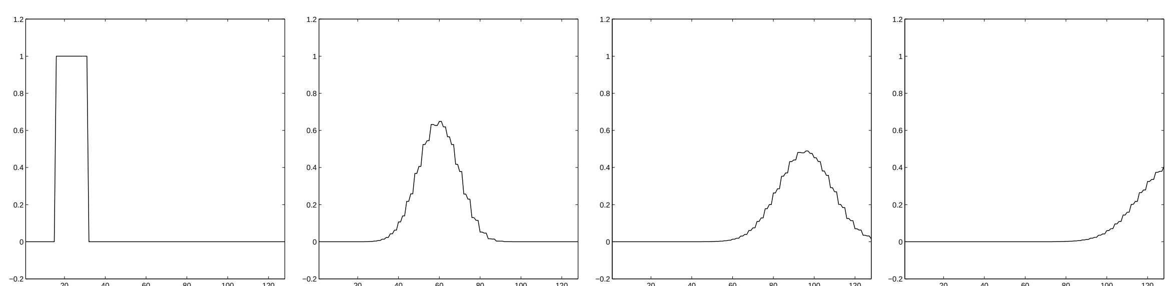

FIG. 10. *The numerical solution given by the 3-level multiplicative scheme,* $\lambda = 0.5$*, at (a) time step = 0, (b) time step = 8, (c) time step = 16, (d) time step = 24.*

Next, we apply the two schemes together with nonlinear upwind restriction and interpolation to solve the Burgers' equation. We start with a shock problem, which has also been tested by Ferm and Lötstedt [11], with boundary conditions: $u(0,t) = 1$ and

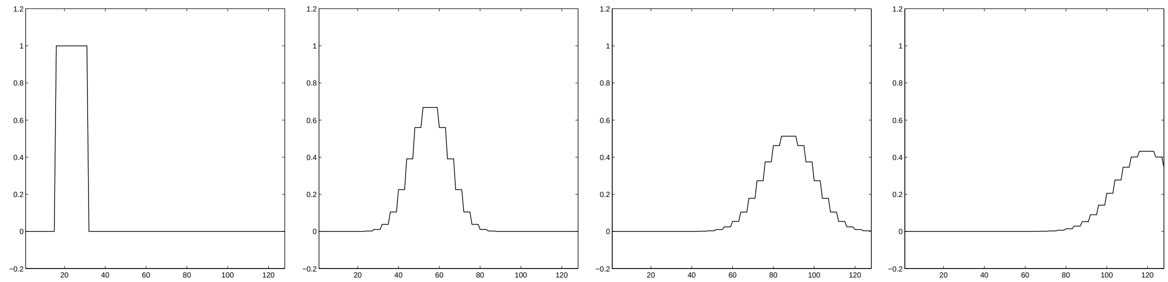

FIG. 11. *The numerical solution given by the 3-level additive scheme, $\lambda = 0.5$, at (a) time step = 0, (b) time step = 16, (c) time step = 32, (d) time step = 48.*

$u(1,t) = -1$, and piecewise constant initial condition as shown in Figure 12 and 13. Thus the steady state solution is

$$u = \begin{cases} 1 & \text{if } 0 \leq x < 0.5 \\ -1 & \text{if } 0.5 < x \leq 1. \end{cases}$$

with a discontinuity at $x = 0.5$. The intermediate solutions given by the four-level multiplicative and additive schemes are shown in Figure 12 and 13. On a single grid, EO scheme alone takes 81 time steps to reach the steady state. The optimal linear speedup for a four-level method is 2^4-1=15, and hence the optimal number of multigrid time steps is $81/15 \approx 6$, which is achieved by the multiplicative scheme. The four-level additive scheme takes 10 multigrid time steps, which is about half the speedup of the multiplicative scheme, as expected.

In the previous example, only shocks are developed. We also test the schemes with the initial condition being $u(x,0) = \cos(5\pi x)$, thus both shocks and rarefactions are developed during time steppings. The steady state solution, however, is the same as the previous example. We again apply the four-level schemes and the results are shown in Figure 14 and 15. The single-grid EO scheme takes 143 time steps whereas the multiplicative and additive schemes take 10 and 19 multigrid time steps, respectively. As before, the multiplicative scheme nearly achieves the optimal speedup of 15, and the additive scheme is about half as efficient.

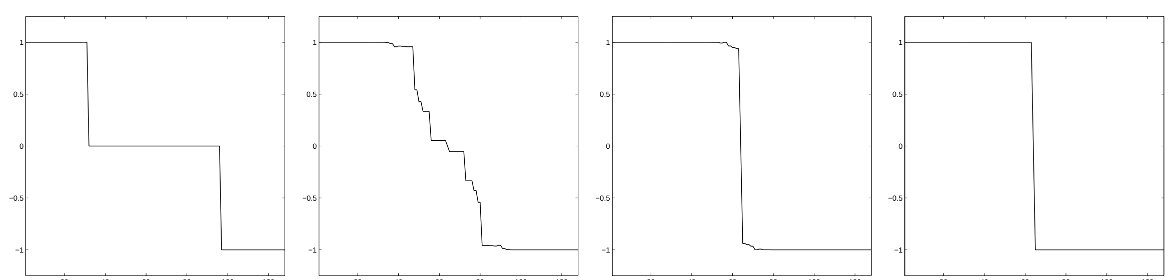

FIG. 12. *The numerical solution given by the 4-level multiplicative scheme, $\lambda = 1.0$, at (a) time step = 0, (b) time step = 2, (c) time step = 4, (d) time step = 6.*

REFERENCES

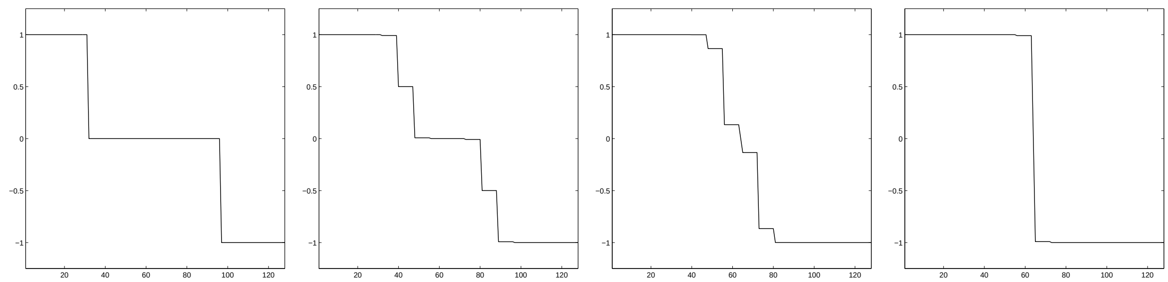

FIG. 13. *The numerical solution given by the 4-level additive scheme, $\lambda = 1.0$, at (a) time step = 0, (b) time step = 3, (c) time step = 6, (d) time step = 9.*

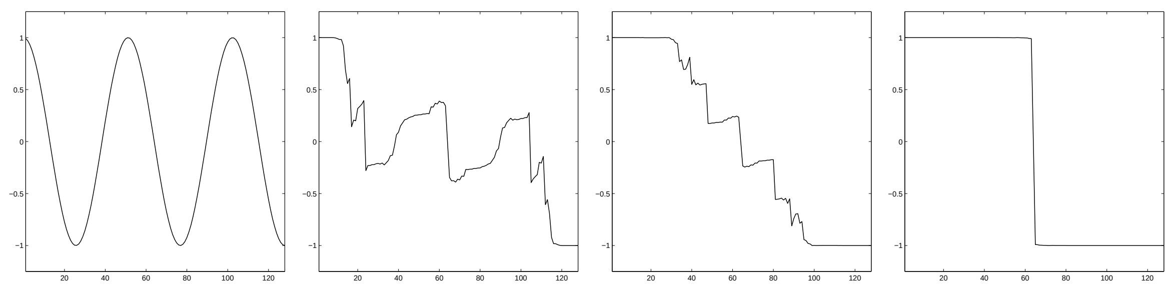

FIG. 14. *The numerical solution given by the 4-level multiplicative scheme, $\lambda = 1.0$, at (a) time step = 0, (b) time step = 3, (c) time step = 6, (d) time step = 9.*

[1] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput.*, 2(4):430–454, 1981.

[2] J. Bramble. *Multigrid Methods.* Longman Scientific & Technical, Essex, UK, 1993.

[3] J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55:1–22, 1990.

[4] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:333–390, 1977.

[5] T. F. Chan and W. L. Wan. Robust multigrid methods for elliptic linear systems. *J. Comput. Appl. Math.*, 123:323–352, 2000.

[6] J. E. Dendy, Jr. Black box multigrid. *J. Comp. Phys.*, 48:366–386, 1982.

[7] J. E. Dendy, Jr., M. P. Ida, and J. M. Rutledge. A semi-coarsening multigrid algorithm for SIMD machines. *SIAM J. Sci Stat. Comput.*, 13:1460–1469, 1992.

[8] J. E. Dendy, Jr., S. F. McCormick, J. W. Ruge, T. F. Russell, and S. Schaffer. Multigrid methods for three-dimensional petroleum reservoir simulation. In *10th Symposium on Reservoir Simulation, Houston*, pages 19–25. Society of Petroleum Engineers (SPE), Inc., February 6–8 1989.

[9] P. Eliasson. *Dissipation Mechanisms and Multigrid Solutions in a Multiblock Solver for Compressible Flow.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1993.

[10] B. Engquist and S. Osher. Stable and entropy satisfying approximations for transonic flow calculations. *Math. Comp.*, 34:45–75, 1980.

[11] L. Ferm and P. Lötstedt. Two-grid solution of shock problems. *SIAM J. Sci. Comput.*, 18:1533–1552, 1997.

[12] F. Grasso and M Marini. Solutions of hypersonic viscous flows with total variation diminishing multigrid techniques. *Comput. & Fluids*, 24:571–592, 1995.

[13] B. Gustafsson and P. Lötstedt. Analysis of the multigrid method applied to first order systems. In J. Mandel et al., editor, *Proceedings of the Fourth Copper Mountain Conference on Multigrid*

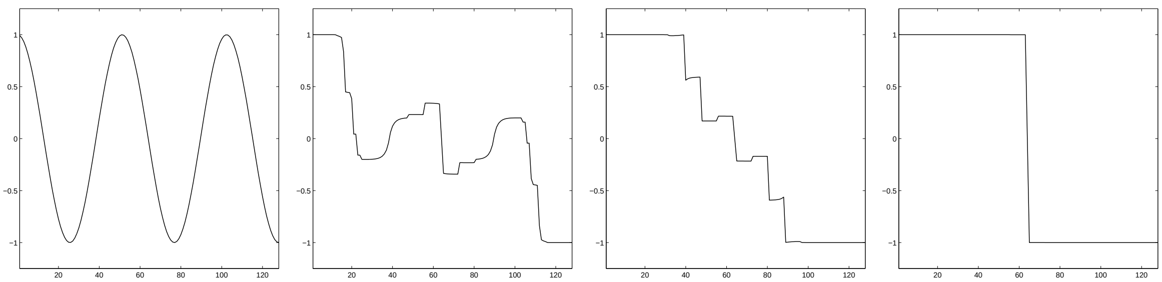

FIG. 15. *The numerical solution given by the 4-level additive scheme, $\lambda = 1.0$, at (a) time step = 0, (b) time step = 6, (c) time step = 12, (d) time step = 18.*

Methods, pages 181–233, Philadelphia, PA, 1989. SIAM.

[14] W. Hackbusch. *Multi-grid Methods and Applications.* Springer-Verlag, Berlin, 1985.

[15] M. G. Hall. Cell vertex multigrid schemes for solution of the Euler equations. In *IMA Conference on Numerical Methods for Fluid Dynamics*, Reading, April 1985.

[16] P. W. Hemker. Multigrid methods for problems with a small parameter. In D.F. Griffiths, editor, *Proceedings, Dundee, June–July 1983*, Lecture Notes in Mathematics 1066, pages 106–121, Berlin, 1983. Springer-Verlag.

[17] P. W. Hemker and G. M. Johnson. Multigrid approaches to the Euler equations. In S. McCormick, editor, *Multigrid Methods*, pages 57–72. SIAM, 1987.

[18] P. W. Hemker and S. P. Spekreijse. Multigrid solution of the steady Euler equations. In *Oberwolfach Meeting on Multigrid Methods*, December 1984.

[19] A. Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Appl. Math. Comp.*, 13:327–355, 1983.

[20] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA paper 81-1259*, 1981.

[21] D. C. Jespersen. Recent developments in multigrid methods for the steady Euler equations. In *Lecture Notes for Lecture Series on Computational Fluid Dynamics.* von Karman Institute for Fluid Dynamics, Rhode–St.–Genese, Belgium, 1984.

[22] D. C. Jespersen. A time-accurate multiple-grid algorithm. *AIAA paper 85-1493-CP*, 1985.

[23] B. Koren and P. W. Hemker. Damped, direction-dependent multigrid for hypersonic flow computations. *Appl. Numer. Math.*, 7:309–328, 1991.

[24] M. P. Leclercq and B. Stoufflet. Characteristic multigrid method application to solve the Euler equations with unstructured and unnested grids. *J. Comput. Phys.*, 104:329–346, 1993.

[25] R. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser, Zürich, 1992.

[26] X. Liu. Private communication. 2000.

[27] P. Lötstedt and B. Gustafsson. Fourier analysis of multigrid methods for general systems of PDEs. *Math. Comp.*, 60:473–493, 1993.

[28] W. Mulder. A new multigrid approach to convection problems. *J. Comp. Phys.*, 83:303, 1989.

[29] W. A. Mulder. Multigrid for the one-dimensional invscid burgers equations. *SIAM J. Sci. Comput.*, 11:33–50, 1990.

[30] R. H. Ni. A multiple-grid scheme for solving the euler equations. *AIAA*, 20:1565–1571, 1982.

[31] A. Reusken. Multigrid with matrix-dependent transfer operators for a singular perturbation problem. *Computing*, 50:199–211, 1993.

[32] J. W. Ruge and K. Stüben. Algebraic multigrid. In S. McCormick, editor, *Multigrid Methods*, pages 73–130. SIAM, 1987.

[33] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, Cambridge, 1996.

[34] R. A. Smith and A. Weiser. Semicoarsening multigrid on a hypercube. *SIAM J. Sci Stat. Comput.*, 13:1314–1329, 1992.

[35] W. P. Tang and W. L. Wan. Sparse approximate inverse smoother for multi-grid. *SIAM J. Matrix*

Anal. Appl., 21:1236–1252, 2000.

[36] W. L. Wan. Interface preserving coarsening multigrid for elliptic problems with highly discontinuous coefficients. *Numer. Lin. Alg. Appl.*, 7:727–741, 2000.

[37] W. L. Wan, T. F. Chan, and B. Smith. An energy-minimizing interpolation for robust multigrid. *SIAM J. Sci. Comput.*, 21:1632–1649, 2000.

[38] P. Wesseling. *An Introduction to Multigrid Methods.* Wiley, Chichester, 1992.

[39] G. Wittum. On the robustness of ilu smoothing. *SIAM J. Sci. Stat. Comput.*, 10:699–717, 1989.

[40] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34:581–613, 1992.

[41] P. M. De Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comp. Appl. Math.*, 33:1–27, 1990.