

MULTIGRID METHODS FOR SYSTEMS OF HYPERBOLIC CONSERVATION LAWS*

SWATHI AMARALA[†] AND JUSTIN W. L. WAN[†]

Abstract. In this paper, we present total variation diminishing (TVD) multigrid methods for computing the steady state solutions for systems of hyperbolic conservation laws. An efficient multigrid method should avoid spurious numerical oscillations. This can be achieved by designing methods which preserve monotonicity and TVD properties through the use of upwind interpolation and restriction techniques. Such multigrid methods have been developed for scalar conservation laws in the literature. However, for hyperbolic systems, the upwinding directions are not apparent. We generalize the upwind techniques to systems by formulating the interpolation as solving a local Riemann problem. Upwind biased restriction is performed on the positive and negative components of the residual. This idea stems from the fact that the flux vector can be split into positive and negative components. For two-dimensional systems, we introduce a novel coarsening technique and extend the upwind interpolation and restriction techniques, which together capture the characteristics of the underlying system of hyperbolic equations. We provide a theoretical analysis to show that our two level method is TVD for one-dimensional linear systems. We also prove that both the additive and multiplicative multigrid schemes are consistent and convergent for one-dimensional linear systems. We demonstrate the effectiveness of our method by numerical examples including Euler equations.

Key words. multigrid methods, upwind biased interpolation and restriction, hyperbolic conservation laws, Euler equations

AMS subject classifications. 65D05, 65F10, 65F50, 65M12, 65M55

DOI. 10.1137/110851316

1. Introduction. Multigrid methods have been used as efficient solvers for a wide variety of problems [34, 1, 6]. The rate of convergence is often independent of the grid size. For elliptic partial differential equations (PDEs), the basic idea of multigrid is that high frequency errors are damped by the smoothers and low frequency errors are corrected by solving the problem on a coarse grid. While the standard theorems in the theory of multigrid methods generally assume ellipticity, it was first proposed by [27, 11] that it is possible to apply multigrid techniques to hyperbolic systems as well. The idea of this approach is to accelerate the wave propagation on multiple grids by using larger time steps on coarse grids without violating the Courant–Friedrichs–Lewy (CFL) condition. Thus the low frequency disturbances are rapidly expelled through the outer boundary while the high frequency modes are locally damped by the smoother. The smoother must be sufficiently dissipative to reduce the high frequency disturbances present in the initial approximation or introduced by the interpolation operator. Ni [27] performs smoothing by means of a control volume centered integration method with fluxes interpolated from corner values. In [16, 17, 2], Chima and Johnson apply the popular MacCormack scheme. Spekrijse [29] proposed a multigrid method for a second order accurate monotone upwind discretization of hyperbolic conservation laws. Dick [3] developed a multigrid method using flux difference splitting for discretization of the Euler equations.

*Received by the editors October 12, 2011; accepted for publication (in revised form) March 25, 2013; published electronically June 26, 2013. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

<http://www.siam.org/journals/mms/11-2/85131.html>

[†]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (samarala@cs.uwaterloo.ca, jwlwan@cs.uwaterloo.ca).

The time step of an explicit scheme is limited by the CFL condition. If one is interested only in the steady state solution of the problem, multigrid methods using fully implicit schemes are unconditionally stable, and they have been used with success for the Euler equations [9, 8, 13, 25, 30]. In [12], Jameson has shown that implicit schemes based on approximate factorization can be used with multigrid techniques to achieve rapidly convergent algorithms for computing the steady state solution of Euler equations. However, the implicit methods require the solution of the system at each time step. In [14], Jespersen gives an extensive review of the advantages and disadvantages of multigrid methods using implicit schemes.

The coarse grid corrections for multigrid methods using explicit time stepping schemes have an accelerating effect since they move the disturbances of the steady state over the distance of many mesh cells in one time step. Jespersen mathematically supported this argument by showing that, under certain assumptions, such multigrid time stepping schemes on an M -grid are 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 [15]. Further, the optimal speed of wave propagation is theoretically estimated to be $2^M - 1$ by [5, 24] using Fourier analysis.

Numerical oscillations can substantially delay the propagation of the wave [33]. Thus, it is desirable to design nonoscillatory multigrid schemes. Two multigrid time stepping schemes were proposed by Jameson [10] for computing the steady state solution of the one-dimensional linear wave equation. The idea is to capture the characteristics of the underlying PDE using upwind biased interpolation and residual restriction operators. The coarse grid update formula is modified to provide smooth results for the hyperbolic equations. Jameson proved that for two levels, these schemes preserve monotonicity. Wan and Jameson [33] extended the analysis to multilevel and proved that the multigrid methods satisfy the total variation diminishing (TVD) property as well. They also extended these schemes to nonlinear scalar conservation laws, which are able to capture discontinuities arising from shocks and rarefactions without violating the entropy conditions.

A number of variants of the characteristic multigrid method have been developed. Koren and Hemker [19] used an upwind prolongation operator and restriction operator as an approximate adjoint of the upwind prolongation operator. The numerical results show that the improved multigrid method performs significantly better than a standard nonlinear multigrid method. Leclercq and Stoufflet [21] emphasized that the use of characteristic transfer operators allows for efficient resolution of shocks. This is not possible with the classical multigrid approach. Grasso and Marini [4] proposed a multigrid technique based on directional grid transfer operators for hypersonic viscous flows.

In this paper, we propose to extend the multigrid schemes based on the characteristics approach [33] to systems of nonlinear hyperbolic conservation laws. However, a main challenge is that the system of equations is composed of a mixture of characteristics. In addition, the characteristic directions change from grid point to grid point. As a result, the upwind biased restriction and interpolation operators defined for scalar conservation laws cannot be directly extended to hyperbolic systems. We address these issues and develop novel upwind biased interpolation and restriction techniques for systems of equations which result in efficient multigrid methods. We extend these upwind biased techniques to two-dimensional systems and introduce a novel coarsening technique. We also provide an analysis of the two level time stepping scheme for one-dimensional linear systems and prove that it satisfies the TVD property. This is a desirable property for designing nonoscillatory multigrid schemes.

Consistency and convergence proofs for multilevel schemes are also provided.

In section 2, we present the flux vector splitting (FVS) scheme, a finite difference scheme used for the discretization of systems of hyperbolic conservation laws. In section 3, we describe the details of our multigrid method for one-dimensional hyperbolic conservation laws, and in section 4, we present an analysis for the TVD, consistency, and convergence properties of the proposed multigrid scheme. In section 5, we extend the multigrid scheme to two-dimensional hyperbolic systems. Finally, in section 6, we demonstrate the effectiveness of our multigrid scheme by testing it on a number of hyperbolic systems including Euler equations.

2. Systems of conservation laws. Consider a multidimensional hyperbolic system of conservative laws,

$$(2.1) \quad U_t + \sum_{s=1}^d F_s(U)_{x_s} = 0, \quad 0 < x_s < 1, \quad t > 0,$$

where $U = (u_1, \dots, u_m)^T \in \mathbb{R}^m$, d is the number of dimensions, $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, $F_s(U)$ is an m -dimensional vector of flux function, and t denotes time. We are interested in finding the steady state solution of the system of equations given the initial condition

$$U(x, 0) = U_0(x).$$

The boundary conditions are of either the Dirichlet or the Neumann type, which will be made more precise in the theoretical analysis in section 4 and in the numerical results in section 6.

There are various finite difference methods [28, 31, 7, 20, 23, 32] for discretizing (2.1). We will use the FVS scheme to discretize the system of conservation laws. In section 3, we will also use it as a smoother for the multigrid method. FVS [32] is a scheme in which the discretization of the equations is performed according to the upwind direction, given by

$$(2.2) \quad U_J^{n+1} = U_J^n - \sum_{s=1}^d \lambda_s (F_{J+\frac{1}{2}e_s} - F_{J-\frac{1}{2}e_s}),$$

where $\lambda_s = \Delta t / \Delta x_s$ is the CFL number, J is the index of the grid point, Δt is the time step, Δx_s is the spatial step in the x_s direction, and e_s is the unit vector in the x_s direction. The flux differences are computed by the following approximation:

$$F_{J+\frac{1}{2}e_s} = F_J^+(U_J^n) + F_{J+e_s}^-(U_{J+e_s}^n).$$

The positive and negative fluxes are given by

$$F_J^\pm(U_J^n) = A^\pm U_J^n,$$

where A^+ and A^- are the positive and negative components of the Jacobian matrix $A = \frac{\partial F}{\partial U}$. Let K be the matrix of right eigenvectors of A . Then,

$$A^\pm = K \Lambda^\pm K^{-1},$$

where Λ^+ and Λ^- are the diagonal matrices whose entries are the positive and the negative eigenvalues of A , respectively.

For constant coefficient systems, the FVS scheme (2.2) can be rewritten as

$$(2.3) \quad U_J^{n+1} = U_J^n - \sum_{s=1}^d (\lambda_s A^+(U_J^n - U_{J-e_s}^n) - \lambda_s A^-(U_{J+e_s}^n - U_J^n)).$$

Note that we use the first order FVS scheme for illustration purposes only; other high resolution schemes, such as the positive schemes [23, 20], can also be used.

3. One-dimensional hyperbolic conservation laws. We design multigrid methods for hyperbolic systems based on the multigrid approach for the scalar conservation laws [33]. This multigrid method preserves monotonicity and is TVD for scalar conservation laws, which are two key properties in obtaining efficient multigrid methods. We first briefly review the multigrid approach for scalar conservation laws and then extend the idea for systems of conservation laws.

Consider a scalar conservation law,

$$(3.1) \quad u_t + f(u)_x = 0, \quad x \in (0, 1), \quad t > 0,$$

with appropriate boundary and initial conditions. The interval $[0, 1]$ is discretized uniformly by taking the N points, $x_j^h = j \times h$, $j = 0, 1, \dots, N-1$, where $h = 1/(N-1)$ denotes the fine grid. Let u_j^h be the approximation of the exact solution $u(x_j^h)$. The boundary values u_0^h and u_{N-1}^h are of either the Dirichlet or the Neumann type. Let b_j^n be the fine grid right-hand side (RHS) function which is initially zero. In general, the RHS function is nonzero for coarse grids. Given $\{x_j^h\}$, $j = 0, 1, \dots, N-1$, the grid points with even indices are selected as coarse grid points; i.e., $\{x_j^H\}$, $j = 0, 2, \dots, N-1$, where H denotes the coarse grid. In a two grid method, the fine grid solution is evolved by one time step using the FVS scheme (2.2), denoted by \bar{u}_j^h . The solution \bar{u}_j^h and the residual \bar{r}_j^h are restricted to the coarse grid Ω^H using the restriction operators \mathcal{R}_u and \mathcal{R}_r , respectively. \mathcal{R}_u is an injection operator. \mathcal{R}_r is an upwind restriction operator, the details of which are given in section 3.2. The coarse grid RHS function is denoted by b_j^H . The solution on the coarse grid is evolved by one time step, where $\Delta t^H = 2\Delta t^h$, to obtain \tilde{u}_j^H . The updated solution u_j^{n+1} is computed by correcting the fine grid solution \bar{u}_j^h with the interpolated coarse grid error $\mathcal{P}(\tilde{u}^H - \mathcal{R}_u u^n)_j$, where \mathcal{P} is the interpolation operator. To summarize, given the solution u_j^n on the fine grid, the two level algorithm for computing u_j^{n+1} is given below.

$$\begin{aligned} \text{Fine grid smoothing:} \quad \bar{u}_j^h &= u_j^n - \lambda(f_{j+\frac{1}{2}}^n - f_{j-\frac{1}{2}}^n) + \Delta t^h b_j^n, \\ & \quad j = 1, 2, \dots, N-2. \end{aligned}$$

$$\begin{aligned} \text{Fine grid residual:} \quad \bar{r}_j^h &= \frac{1}{\Delta x^h} (f_{j+\frac{1}{2}}^h - f_{j-\frac{1}{2}}^h) - b_j^n, \\ & \quad j = 1, 2, \dots, N-2. \end{aligned}$$

$$(3.2) \quad \begin{aligned} \text{Restriction of solution:} \quad u_j^H &= \mathcal{R}_u \bar{u}_j^h, \\ & \quad j = 2, 4, \dots, N-3. \end{aligned}$$

$$(3.3) \quad \begin{aligned} \text{Coarse grid RHS:} \quad b_j^H &= \frac{1}{\Delta x^H} (f_{j+\frac{1}{2}}^H - f_{j-\frac{1}{2}}^H) - \mathcal{R}_r \bar{r}_j^h, \\ & \quad j = 2, 4, \dots, N-3. \end{aligned}$$

$$(3.4) \quad \text{Coarse grid evolution:} \quad \tilde{u}_j^H = u_j^H - \lambda(f_{j+\frac{1}{2}}^H - f_{j-\frac{1}{2}}^H) + \Delta t^H b_j^H, \\ j = 2, 4, \dots, N-3.$$

$$(3.5) \quad \text{Fine grid update:} \quad \begin{cases} u_j^{n+1} = \tilde{u}_j^H, \\ u_{j-1}^{n+1} = \bar{u}_{j-1}^h + \mathcal{P}(\tilde{u}^H - \mathcal{R}_u u^n)_{j-1}, \end{cases} \\ j = 2, 4, \dots, N-1.$$

A multigrid algorithm is obtained by recursively applying the two level algorithm on the coarser grids. Since, on each coarse grid, we use the updated solution, \tilde{u}^h , from the finer grid, we call this method the multiplicative scheme. Another approach is to restrict and propagate u^n on all coarse grids; i.e., (3.2) is replaced by

$$u_j^H = u_j^n, \quad j = 2, 4, \dots, N-3.$$

The resulting algorithm is called the additive scheme. For the additive scheme, b_j^H can be computed in two ways. We can either use (3.3) or set it to zero. Both approaches have similar convergence rates. Setting $b_j^H = 0$ enables parallel computation of the algorithm.

We will extend the above multigrid methods to systems of hyperbolic conservation laws in one dimension,

$$(3.6) \quad U_t + F(U)_x = 0, \quad 0 < x < 1, \quad t > 0,$$

where U is an m -dimensional vector. Given the appropriate initial and boundary conditions, we are interested in computing the steady state solution of the system (3.6). The basic principle of the multiplicative and additive multigrid methods is essentially the same as those for scalar conservation laws. We perform smoothing on the fine grid, restrict the solution to the coarser grids, and accelerate the propagation on the coarser grids. However, the restriction and interpolation operators need to be redefined, as there is a mixture of characteristics in a system.

3.1. Interpolation. The idea of upwind biased interpolation is to interpolate values based on the direction of propagation of information. Consider the linear wave equation $u_t + u_x = 0$. In a two level method, given the coarse grid values \tilde{u}_{j-2}^H and \tilde{u}_j^H , $j = 2, 4, \dots, N-1$, we want to interpolate the solution at x_{j-1}^h . Since the wave is propagating from left to right, the interpolated value is given by $\tilde{u}_{j-1}^h = \tilde{u}_{j-2}^H$. For scalar hyperbolic conservation laws, the direction of propagation of information can be determined using the sign of the characteristic speeds. If the sign is positive on both coarse grid points, the information is propagating from left to right and the fine grid solution is interpolated from the left. Conversely, if the sign is negative, the interpolated value is from the right. For mixed signs (e.g., shocks and rarefactions), upwind interpolation can also be defined [33].

Schemes for applying upwind biased interpolation to systems of hyperbolic equations, however, are not apparent since the characteristics for some characteristic variables can be to the right while others can be to the left. This is equivalent to the fact that some eigenvalues of the Jacobian of F can be positive and some can be negative. One possible solution is to apply upwind interpolation to each characteristic variable explicitly. This method works well when the right eigenvectors (i.e., K) are constant; e.g., $F(U) = AU$ where A is a constant matrix. In this case, let $W = K^{-1}U$. Then (3.6) becomes a decoupled system in the new characteristic variables W . One can

easily apply the upwind interpolation to this system. In general, however, the right eigenvectors are not constant. Then it would not be a good idea to apply upwind interpolation to the characteristic variables ($W_j = K_j^{-1}U_j$) from one grid point to interpolate the values of the characteristic variables ($W_{j-1} = K_{j-1}^{-1}U_{j-1}$) at another grid point when K_j and K_{j-1} are different.

There is another more subtle issue of using characteristics explicitly for defining upwind interpolation. Recall that the coarse grid correction step updates the fine grid solution by interpolating the error from the coarse grid as given by (3.5). When (3.5) is directly applied to systems, we have the fine grid update, \tilde{U}_{j-1}^h , as

$$\tilde{U}_{j-1}^h = \bar{U}_{j-1}^h + \mathcal{P}(\tilde{U}^H - \mathcal{R}_u U^h)_{j-1}, \quad j = 2, 4, \dots, N - 1.$$

For nonlinear equations, upwind interpolation based on characteristics will depend on the solution being interpolated. Thus, if one tries to interpolate the coarse grid error to the fine grid, the characteristics of the error will be used to define the upwind direction, which may not necessarily capture the behavior of the solution. One may interpolate \tilde{U}^H by \mathcal{P}_1 based on the characteristics of \tilde{U}^H and separately interpolate $\mathcal{R}_u U^h$ by \mathcal{P}_2 based on the characteristics of $\mathcal{R}_u U^h$. In general \mathcal{P}_1 and \mathcal{P}_2 are not the same, and hence a different interpolation will be performed at the same grid point for \tilde{U}^H and $\mathcal{R}_u U^h$. In our experience, this inconsistency leads to slow convergence in general. One may pick one of the two interpolation operators. In [33], the authors apply \mathcal{P}_1 to interpolate $\mathcal{R}_u U^h$. However, in general, it is not clear whether one should choose \mathcal{P}_1 or \mathcal{P}_2 , and whether the upwind interpolation based on one solution would be good for the other.

To address this issue, we note that upwind interpolation can be formulated as solving a local Riemann problem [33]. Consider the linear wave equation and the following local boundary value problem:

$$(3.7) \quad \begin{aligned} e_\tau + e_x &= 0, & x_{j-2}^h < x < x_j^h, \\ e(x, 0) &= \begin{cases} e_{j-2}^H, & x_{j-2}^h < x < x_{j-1}^h, \\ e_j^H, & x_{j-1}^h < x < x_j^h, \end{cases} \end{aligned}$$

where τ is an artificial time, and e_{j-2}^H and e_j^H are two given constant values as shown in Figure 1. The steady state solution of (3.7) at x_{j-1}^h gives the interpolated value. In this case, the value is e_{j-2}^H , which is the same as the value given by the upwind biased interpolation.

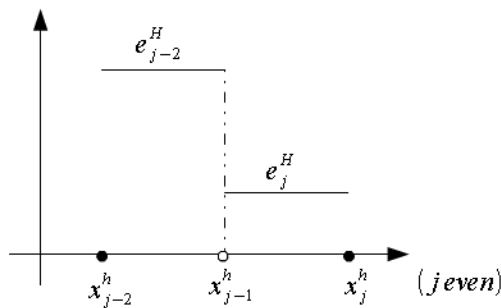


FIG. 1. The initial data for a Riemann problem for defining an upwind interpolation. The data consist of two constant states separated by a discontinuity at x_{j-1}^h .

We extend the local Riemann formulation for upwind interpolation to systems of hyperbolic equations. In this case, we do not interpolate the coarse grid error onto the fine grid and then update the fine grid solution. Instead, we update the fine grid solution at the coarse grid points using the coarse grid solution. At the noncoarse grid points, we determine the values by upwind interpolation. More precisely, for $j = 2, 4, \dots, N - 1$, we solve the following local Riemann problem:

$$(3.8) \quad \begin{aligned} U_\tau + F(U)_x &= 0, & x_{j-2}^h < x < x_j^h, \\ U(x, 0) &= \begin{cases} \tilde{U}_{j-2}^h, & x_{j-2}^h < x < x_{j-1}^h, \\ \tilde{U}_j^h, & x_{j-1}^h < x < x_j^h, \end{cases} \end{aligned}$$

where \tilde{U}_{j-2}^h and \tilde{U}_j^h are the fine grid values of U at x_{j-2}^h and x_j^h , respectively, after coarse grid correction. We compute the steady state solution by using the FVS scheme (2.2) for each noncoarse grid point. The upwind interpolation is essentially carried out by the upwind discretization of the FVS scheme. Since the local Riemann problem involves only one grid point, the computation is very efficient.

We note that the interpolation can be performed by computing the exact solution to the local Riemann problem. For the linear wave equation example, the exact solution at x_{j-1}^h is \tilde{u}_{j-2}^H , which is the same as (3.9). In general, however, not all the hyperbolic equations have analytical solutions. In order for the method to be applicable to general hyperbolic systems, we describe the interpolation procedure as solving a local Riemann problem numerically to steady state.

The proposed interpolation approach resolves the two issues addressed above. The FVS scheme captures the different characteristics through the two flux functions. Thus, we do not need to determine the characteristic direction for each characteristic variable in order to apply the upwind interpolation. Second, the local Riemann problem is based on the fine grid values of U , and hence we avoid the inconsistency inherent in choosing whether to apply the upwind interpolation to \tilde{U}^H or $\mathcal{R}_u U^h$.

We remark that this interpolation approach results in a different coarse grid correction. Consider the one-dimensional linear wave equation $u_t + u_x = 0$. In [33], the coarse grid update at a noncoarse grid point is

$$\tilde{u}_{j-1}^h = \bar{u}_{j-1}^h + (\tilde{u}_{j-2}^H - u_{j-2}^h), \quad j = 2, 4, \dots, N - 1.$$

The new approach performs upwind interpolation directly on \tilde{u}^h from coarse grid points to noncoarse grid points. Thus,

$$(3.9) \quad \tilde{u}_{j-1}^h = \tilde{u}_{j-2}^H, \quad j = 2, 4, \dots, N - 1.$$

We will show in section 4 that the new interpolation approach will still result in a multigrid scheme which is TVD.

3.2. Residual restriction. For scalar hyperbolic equations, an upwind biased residual restriction can be defined based on the characteristic direction [33]. Consider the scalar conservation law (3.1) again. The characteristic direction can be determined by the sign of the derivative of the flux, $f'(u)$. Given the fine grid residual, r_j^h , we want to compute the residual on the coarse grid, r_j^H . Suppose $f'_j(u_{j-1})$, $f'_j(u_j)$, and $f'_j(u_{j+1})$ are all positive; then the information is propagating from left to right, and hence averaging for the residual is performed from the grid point towards its left as given by

$$(3.10) \quad r_j^H = 0.5[r_j^h + r_{j-1}^h], \quad j = 2, 4, \dots, N - 1,$$

and $r_0^H = 0$. Similarly, when the wave is locally propagating to the left, averaging for residual is performed from the right. For details on special situations such as shocks and rarefactions, we refer the reader to [33].

The upwind residual restriction technique described above works well since the scalar unknown is either moving forward or backward at each grid point. However, a system of hyperbolic conservation laws is composed of a mixture of characteristics; i.e., it contains both forward and backward moving components. Thus, this approach cannot be directly extended to the system case. The main issue is that it is not apparent what the upwind side is at each grid point for systems, and hence averaging for the residual based on the upwind direction becomes infeasible.

Nevertheless, the concept of propagation direction of information can still be exploited for systems of equations. The FVS intercell numerical flux is composed of two contributions,

$$(3.11) \quad F_{j+\frac{1}{2}} = F_j^+(U_j^h) + F_{j+1}^-(U_{j+1}^h), \quad j = 1, 2, \dots, N - 2,$$

where one comes from the forward component of the flux, F_j^+ , and the other comes from the backward component of the flux, F_{j+1}^- , as shown in Figure 2. Using the idea of the FVS numerical flux splitting (3.11), we will split the fine grid residual, $R_j^h, j = 1, 2, \dots, N - 2$, similarly into two parts, based on the forward and backward flux components. More precisely, let $(R_j^h)^+$ be the residual from the forward component of fluxes, and let $(R_j^h)^-$ be the residual from the backward component of fluxes. Then $R_j^h = (R_j^h)^+ + (R_j^h)^-$, where

$$\begin{aligned} (R_j^h)^+ &= \frac{1}{\Delta x^h} [F_j^+(U_j^h) - F_{j-1}^+(U_{j-1}^h)] - (B^h)_j^+, \\ (R_j^h)^- &= \frac{1}{\Delta x^h} [F_{j+1}^-(U_{j+1}^h) - F_j^-(U_j^h)] - (B^h)_j^-, \\ & \quad j = 1, 2, \dots, N - 2. \end{aligned}$$

We note that $R_0^h = R_{N-1}^h = 0$. $(B^h)_j^+$ and $(B^h)_j^-$ are the positive and negative RHS functions which will be made more precise later in this section.

In order to define the upwind restriction for the residual, we note that the Jacobian matrix, $A^+ = \frac{\partial F^+}{\partial U}$, is positive semidefinite and $A^- = \frac{\partial F^-}{\partial U}$ is negative semidefinite. The eigenvalues determine the characteristic direction of the system. A^+ has positive or zero eigenvalues, and the information is propagating from left to right. Similarly,

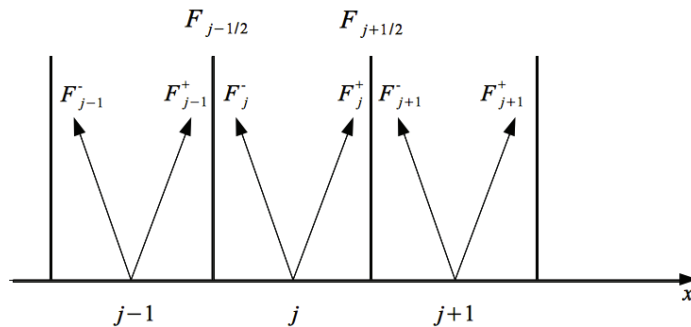


FIG. 2. The forward and backward flux components of the intercell flux functions $F_{j-\frac{1}{2}}$ and $F_{j+\frac{1}{2}}$.

A^- has negative or zero eigenvalues, and the information is propagating from right to left. We exploit these directionality properties of the FVS scheme to perform upwind biased residual restriction. For the positive component of the residual, averaging is performed from the grid point towards its left as given by

$$(R^H)_j^+ = 0.5[(R^h)_j^+ + (R^h)_{j-1}^+], \quad j = 2, 4, \dots, N-3.$$

For the negative component of the residual, averaging is performed from the grid point towards its right,

$$(R^H)_j^- = 0.5[(R^h)_j^- + (R^h)_{j+1}^-], \quad j = 2, 4, \dots, N-3.$$

The coarse grid residual, R_j^H , is then given by the sum of the positive and negative coarse grid residuals,

$$(3.12) \quad \begin{aligned} R_j^H &= (R^H)_j^+ + (R^H)_j^-, & j &= 2, 4, \dots, N-3, \\ R_j^H &= 0, & j &= 0, N-1. \end{aligned}$$

Following a similar idea, the coarse grid RHS, B_j^H , is then defined by

$$\begin{aligned} B_j^H &= (B^H)_j^+ + (B^H)_j^-, & j &= 2, 4, \dots, N-3, \\ B_j^H &= 0, & j &= 0, N-1, \end{aligned}$$

where

$$\begin{aligned} (B^H)_j^+ &= \frac{1}{\Delta x} [F_j^+(U_j^H) - F_{j-2}^+(U_{j-2}^H)] - 0.5[(R^h)_j^+ + (R^h)_{j-1}^+], \\ (B^H)_j^- &= \frac{1}{\Delta x} [F_{j+2}^-(U_{j+2}^H) - F_j^-(U_j^H)] - 0.5[(R^h)_j^- + (R^h)_{j+1}^-], \\ & & j &= 2, 4, \dots, N-3. \end{aligned}$$

This residual restriction approach extends the previous approach for scalar conservation laws. Consider the scalar equation (3.1) again. Suppose the wave is locally propagating to the right. Then the residual restriction based on characteristics is given by (3.10). In our new approach, $r_j^H = (r^H)_j^+ + (r^H)_j^-$, $j = 2, 4, \dots, N-1$. However, since the sign of the characteristic speed is positive, the backward moving component of the flux is zero; i.e., $(r^H)^- = (r^h)^- = 0$. As a result, $r_j^H = (r^H)_j^+ = 0.5[(r^h)_j^+ + (r^h)_{j-1}^+] = 0.5[(r^h)_j + (r^h)_{j-1}]$, which is the same as (3.10). By a similar calculation, for other characteristic directions, including cases of shocks and rarefactions, it can be shown that the new approach recovers the same formulas as given by the upwind restriction technique defined for the scalar case. This generalization has the advantage that it does not require explicit knowledge of the characteristics as in [33], and hence can be easily applied to the system case.

3.3. Algorithm. The multiplicative multigrid time stepping scheme for one-dimensional systems of hyperbolic conservation laws is given in Algorithm 1. We denote the functions on the finest grid by (1), the second grid by (2), and so on. We denote the functions on the coarsest grid by (L). FVS scheme (2.2) is used as the smoother. The interpolation operator \mathcal{P} solves a local Riemann problem to the steady state; see (3.8).

ALGORITHM 1. Multiplicative multigrid scheme for one-dimensional systems of hyperbolic conservation laws.

```

 $U^{(1)} = U^n, B^{(1)} = 0$ 
Define  $\tilde{U}^{(k)} = \text{MG}(U^{(k)}, B^{(k)})$ :
if  $k = L$  then
     $\tilde{U}_j^{(k)} = U_j^{(k)} - \lambda(F_{j+\frac{1}{2}}^{(k)} - F_{j-\frac{1}{2}}^{(k)}) + \Delta t^{(k)} B_j^{(k)}$     ( $j = 2^{k-1}, 2 \cdot 2^{k-1}, \dots$ )
else
     $\bar{U}_j^{(k)} = U_j^{(k)} - \lambda(F_{j+\frac{1}{2}}^{(k)} - F_{j-\frac{1}{2}}^{(k)}) + \Delta t^{(k)} B_j^{(k)}$     ( $j = 2^{k-1}, 2 \cdot 2^{k-1}, \dots$ )
     $\bar{R}_j^{(k)} = \frac{1}{\Delta x^{(k)}} (\bar{F}_{j+\frac{1}{2}}^{(k)} - \bar{F}_{j-\frac{1}{2}}^{(k)}) - B_j^{(k)}$     ( $j = 2^{k-1}, 2 \cdot 2^{k-1}, \dots$ )
     $U_j^{(k+1)} = \mathcal{R}_u \bar{U}_j^{(k)}$     ( $j = 2^k, 2 \cdot 2^k, \dots$ )
     $B_j^{(k+1)} = \frac{1}{\Delta x^{(k+1)}} (F_{j+\frac{1}{2}}^{(k+1)} - F_{j-\frac{1}{2}}^{(k+1)}) - \mathcal{R}_r \bar{R}_j^{(k)}$     ( $j = 2^k, 2 \cdot 2^k, \dots$ )
     $\tilde{U}^{(k+1)} = \text{MG}(U^{(k+1)}, B^{(k+1)})$ 
     $\tilde{U}_j^{(k)} = \tilde{U}_j^{(k+1)}$     ( $j = 2^k, 2 \cdot 2^k, \dots$ )
     $\tilde{U}_{j-2^{k-1}}^{(k)} = \mathcal{P}(\tilde{U}_{j-2 \cdot 2^{k-1}}^{(k+1)}, \bar{U}_{j-2^{k-1}}^{(k)}, \tilde{U}_j^{(k+1)})$     ( $j = 2^k, 2 \cdot 2^k, \dots$ )
end if
 $U^{n+1} = \text{MG}(U^{(1)}, B^{(1)})$ 

```

On each coarse grid k , we restrict and propagate the most recently updated solution $\bar{U}^{(k-1)}$ from the finer grid in the multiplicative scheme as given in Algorithm 1. If we restrict and propagate $U^{(k)}$ on all the coarse grids as

$$U_j^{(k+1)} = U_j^{(k)} \quad (j = 2^k, 2 \cdot 2^k, \dots),$$

then we obtain the additive multigrid time stepping scheme.

In the next section, we prove that both the two level multiplicative and additive time stepping schemes satisfy the TVD, consistency, and convergence properties for constant coefficient linear systems of hyperbolic conservation laws. In Appendix A.1, we prove that the two level schemes for the linear wave equation preserve monotonicity. These properties are important in ensuring that the multigrid methods are nonoscillatory and in turn result in smooth and fast convergence of the multigrid solution.

4. Linear analysis. In this section, we analyze the TVD, consistency, and convergence properties of multigrid time stepping schemes for solving an $m \times m$ hyperbolic system of linear conservation laws given by

$$(4.1) \quad U_t + AU_x = 0, \quad 0 < x < 1, t > 0,$$

where A is an $m \times m$ constant matrix. Without loss of generality, we assume a zero boundary condition and a zero RHS. We use the FVS scheme (2.2) as the smoother for the multigrid algorithm. We first define the TVD and consistency preserving properties.

DEFINITION 4.1. *Given any set of data U^n , let U^{n+1} be computed by one multigrid cycle. Then the multigrid method is called TVD [22] if*

$$TV(U^{n+1}) \leq TV(U^n),$$

where the total variation of a function U is defined in terms of its vector norm as

$$(4.2) \quad TV(U) = \sum_j \|U_j - U_{j-1}\|.$$

DEFINITION 4.2. Let ϕ be any numerical scheme for solving a PDE. $U^{n+1} = \phi(U^n)$ is a consistent scheme if $U^* = \phi(U^*)$, where U^* is the exact solution.

Before we present the analysis, we make a useful observation about the coarse grid RHS B_j^H .

LEMMA 4.3. The coarse grid RHS function is zero for constant coefficient linear systems.

Proof. For the two level multiplicative scheme given by Algorithm 1, we rewrite B_j^H with $j = 2, 4, \dots, N - 3$ for constant coefficient linear systems as

$$\begin{aligned} B_j^H &= \frac{1}{\Delta x^H} (A^+(U_j^H - U_{j-2}^H) + A^-(U_{j+2}^H - U_j^H)) - \frac{1}{2} ((\bar{R}_j^h)^+ + (\bar{R}_{j-1}^h)^+) \\ &\quad - \frac{1}{2} ((\bar{R}_{j-1}^h)^- + (\bar{R}_j^h)^-) \\ &= \frac{1}{2\Delta x^h} (A^+(\bar{U}_j^h - \bar{U}_{j-2}^h) + A^-(\bar{U}_{j+2}^h - \bar{U}_j^h)) - \frac{1}{2\Delta x^h} (A^+(\bar{U}_j^h - \bar{U}_{j-1}^h) \\ &\quad + A^+(\bar{U}_{j-1}^h - \bar{U}_{j-2}^h)) - \frac{1}{2\Delta x^h} (A^-(\bar{U}_{j+1}^h - \bar{U}_j^h) + A^-(\bar{U}_{j+2}^h - \bar{U}_{j+1}^h)) \\ &= 0. \quad \square \end{aligned}$$

For constant coefficient linear systems, it is possible to prove stability by measuring the total variation in terms of wave strengths instead of using the standard vector norms [22]. Let K be the matrix of right eigenvectors of A and define $W = K^{-1}U$. The total variation for constant coefficient linear systems can then be computed in terms of its characteristic variables.

LEMMA 4.4 (see [22]). The total variation of an m -dimensional vector U can be computed in terms of W as

$$TV(U) = \sum_{i=1}^m TV(W_i),$$

where $TV(W_i)$ is the total variation of the i th characteristic component.

We assume that A has m_1 positive eigenvalues and m_2 negative eigenvalues, where $m_1 + m_2 = m$. Without loss of generality, we assume that the vector of positive eigenvalues, α^+ , of A is ordered before the vector of negative eigenvalues, α^- . Thus we write W as

$$(4.3) \quad W = \begin{bmatrix} p \\ q \end{bmatrix},$$

where $p = [p_1, p_2, \dots, p_{m_1}]$ is an m_1 -dimensional vector with positive sign for the characteristic speeds and $q = [q_{m_1+1}, q_{m_1+2}, \dots, q_m]$ is an m_2 -dimensional vector with negative sign for the characteristic speeds.

In the following theorem, we analyze the TVD properties of the multigrid method for the system of equations (4.1).

THEOREM 4.5. Let U^n be the solution at time step n ; then,

$$TV(U^{n+1}) \leq TV(U^n)$$

for both the two level multiplicative and additive multigrid time stepping schemes.

Proof. From Lemma 4.4 and (4.3), we have

$$(4.4) \quad TV(U^{n+1}) = \sum_{i=1}^{m_1} TV(p_i^{n+1}) + \sum_{i=m_1+1}^m TV(q_i^{n+1}).$$

We now compute $TV(p_i^{n+1})$, $i = 1, 2, \dots, m_1$, using Lemma A.4. Let $\mathcal{J} = \{2, 4, \dots, N - 1\}$. We assume zero boundary conditions and $(p_i)_j^n = 0$, $j \leq 0$. From the CFL condition, we have $0 \leq |\lambda\alpha_i^+| \leq 1$, and

$$(4.5) \quad \begin{aligned} TV(p_i^{n+1}) &= \sum_{j \in \mathcal{J}} |(p_i)_j^{n+1} - (p_i)_{j-1}^{n+1}| + |(p_i)_{j-1}^{n+1} - (p_i)_{j-2}^{n+1}| \\ &\leq (1 - \lambda\alpha_i^+)^2 |((p_i)_j^n - (p_i)_{j-1}^n)| + (1 - \lambda\alpha_i^+)^2 |((p_i)_{j-1}^n - (p_i)_{j-2}^n)| \\ &\quad + \lambda\alpha_i^+(1 - \lambda\alpha_i^+) |((p_i)_{j-1}^n - (p_i)_{j-2}^n)| + \lambda\alpha_i^+(1 - \lambda\alpha_i^+) |((p_i)_{j-2}^n - (p_i)_{j-3}^n)| \\ &\quad + \lambda\alpha_i^+(1 - \lambda\alpha_i^+) |((p_i)_{j-2}^n - (p_i)_{j-3}^n)| + \lambda\alpha_i^+(1 - \lambda\alpha_i^+) |((p_i)_{j-3}^n - (p_i)_{j-4}^n)| \\ &\quad + (\lambda\alpha_i^+)^2 |((p_i)_{j-3}^n - (p_i)_{j-4}^n)| + (\lambda\alpha_i^+)^2 |((p_i)_{j-4}^n - (p_i)_{j-5}^n)| \\ &\leq ((1 - \lambda\alpha_i^+)^2 + 2\lambda\alpha_i^+(1 - \lambda\alpha_i^+) + (\lambda\alpha_i^+)^2) TV(p_i^n) \\ &= TV(p_i^n). \end{aligned}$$

Similarly $TV(q_i^{n+1})$, where $i = m_1 + 1, m_1 + 2, \dots, m$ is computed using Lemma A.4. We note that with zero boundary conditions, $(q_i)_j^n = 0$, $j \geq N - 1$.

$$(4.6) \quad \begin{aligned} TV(q_i^{n+1}) &= \sum_{j \in \mathcal{J}} |(q_i)_j^{n+1} - (q_i)_{j-1}^{n+1}| + |(q_i)_{j-1}^{n+1} - (q_i)_{j-2}^{n+1}| \\ &= \sum_{j \in \mathcal{J}} |(\lambda\alpha_i^-)^2 ((q_i)_{j+3}^n - (q_i)_{j+2}^n) + (\lambda\alpha_i^-)^2 ((q_i)_{j+2}^n - (q_i)_{j+1}^n) \\ &\quad - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) ((q_i)_{j+2}^n - (q_i)_{j+1}^n) - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) ((q_i)_{j+1}^n - (q_i)_j^n) \\ &\quad - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) ((q_i)_{j+1}^n - (q_i)_j^n) - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) ((q_i)_j^n - (q_i)_{j-1}^n) \\ &\quad + (1 + \lambda\alpha_i^-)^2 ((q_i)_j^n - (q_i)_{j-1}^n) + (1 + \lambda\alpha_i^-)^2 ((q_i)_{j-1}^n - (q_i)_{j-2}^n)|. \end{aligned}$$

We have the CFL condition as $0 \leq |\lambda\alpha_i^-| \leq 1$. Since $\alpha_i^- < 0$, (4.6) can be rewritten as

$$(4.7) \quad \begin{aligned} TV(q_i^{n+1}) &\leq \sum_{j \in \mathcal{J}} (\lambda\alpha_i^-)^2 |((q_i)_{j+3}^n - (q_i)_{j+2}^n)| + (\lambda\alpha_i^-)^2 |((q_i)_{j+2}^n - (q_i)_{j+1}^n)| \\ &\quad - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) |((q_i)_{j+2}^n - (q_i)_{j+1}^n)| - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) |((q_i)_{j+1}^n - (q_i)_j^n)| \\ &\quad - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) |((q_i)_{j+1}^n - (q_i)_j^n)| - \lambda\alpha_i^-(1 + \lambda\alpha_i^-) |((q_i)_j^n - (q_i)_{j-1}^n)| \\ &\quad + (1 + \lambda\alpha_i^-)^2 |((q_i)_j^n - (q_i)_{j-1}^n)| + (1 + \lambda\alpha_i^-)^2 |((q_i)_{j-1}^n - (q_i)_{j-2}^n)| \\ &\leq ((\lambda\alpha_i^-)^2 - 2\lambda\alpha_i^-(1 + \lambda\alpha_i^-) + (1 + \lambda\alpha_i^-)^2) TV(q_i^n) \\ &= TV(q_i^n). \end{aligned}$$

Hence by (4.4), (4.5), and (4.7), the multiplicative multigrid time stepping scheme is TVD for systems of constant coefficient linear hyperbolic conservation laws.

Similarly for a two level additive scheme, $TV(p_i^{n+1})$, where $i = 1, 2, \dots, m_1$ is computed using Lemma A.5,

$$\begin{aligned} TV(p_i^{n+1}) &\leq \sum_{j \in \mathcal{J}} (1 - \lambda \alpha_i^+) |((p_i)_j^n - (p_i)_{j-1}^n)| + (1 - \lambda \alpha_i^+) |((p_i)_{j-1}^n - (p_i)_{j-2}^n)| \\ &\quad + \lambda \alpha_i^+ |((p_i)_{j-2}^n - (p_i)_{j-3}^n)| + \lambda \alpha_i^+ |((p_i)_{j-3}^n - (p_i)_{j-4}^n)| \\ &\leq (1 - \lambda \alpha_i^+ + \lambda \alpha_i^+) TV(p_i^n) \\ (4.8) \quad &= TV(p_i^n). \end{aligned}$$

$TV(q_i^{n+1})$, $i = m_1 + 1, m_1 + 2, \dots, m$, is obtained from Lemma A.5 as

$$\begin{aligned} TV(q_i^{n+1}) &\leq \sum_{j \in \mathcal{J}} -\lambda \alpha_i^- |((q_i)_{j+2}^n - (q_i)_{j+1}^n)| - \lambda \alpha_i^- |((q_i)_{j+1}^n - (q_i)_j^n)| \\ &\quad + (1 + \lambda \alpha_i^-) |((q_i)_j^n - (q_i)_{j-1}^n)| + (1 + \lambda \alpha_i^-) |((q_i)_{j-1}^n - (q_i)_{j-2}^n)| \\ &\leq (-\lambda \alpha_i^- + 1 + \lambda \alpha_i^-) TV(q_i^n) \\ (4.9) \quad &= TV(q_i^n). \end{aligned}$$

Thus, by (4.4), (4.8), and (4.9) the additive scheme for constant coefficient linear systems also satisfies the TVD property. \square

The analysis for consistency and convergence properties is presented in the following theorems.

THEOREM 4.6. *Both the multiplicative and additive time stepping schemes for constant coefficient linear systems are consistent.*

Proof. The characteristic form of the linear system (4.1) is given by

$$(4.10) \quad W_t + \Lambda W_x = 0, \quad 0 < x < 1, t > 0.$$

We rewrite (4.10) as two separate equations: one for vector p associated with positive eigenvalues α^+ , and the other for vector q associated with negative eigenvalues α^- using (4.3):

$$(4.11) \quad p_t + \alpha^+ p_x = 0,$$

$$(4.12) \quad q_t + \alpha^- q_x = 0.$$

Suppose $p^n = p^*$ is the exact steady state solution of (4.11), which is given by $p_j^* = p_0^*$, $j = 0, 1, \dots, N - 1$. We then compute p_j^{n+1} , $j = 1, 2, \dots, N - 1$, using a k -level additive scheme. By Lemma A.5,

$$p_j^{n+1} = (1 - \lambda \alpha^+ + \lambda \alpha^+) p_0^* = p_0^*.$$

Similarly, if $q^n = q^*$ is the exact steady state solution of (4.12), which is given by $q_j^* = q_{N-1}^*$, $j = 0, 1, \dots, N - 1$, then q_j^{n+1} , $j = 0, 1, \dots, N - 2$, is computed using Lemma A.5,

$$q_j^{n+1} = (1 + \lambda \alpha^- - \lambda \alpha^-) q_{N-1}^* = q_{N-1}^*.$$

By Definition 4.2, the k -level additive scheme is consistent for each scalar conservation law in the W -space, and hence in the U -space as well. The consistency of the multilevel multiplicative scheme can be proved using similar reasoning; we omit the details. \square

THEOREM 4.7. *Both the additive and multiplicative time stepping schemes for constant coefficient linear systems are convergent.*

Proof. Consider the linear system (4.1). Without loss of generality, we assume that the RHS and boundary conditions are zero. For the two level multiplicative scheme, $(p_i)^{n+1}$, $j = 2, 3, \dots, N - 1$, is given by Lemma A.4. In matrix notation, we have $p_i^{n+1} = M_{\text{Mult}}^i p_i^n$, where M_{Mult}^i is an $(N - 2) \times (N - 2)$ iteration matrix. For notational convenience, we define $c = \lambda\alpha_i^+$. The iteration matrix is then given by

$$M_{\text{Mult}}^i = \begin{bmatrix} (1-c)^2 & & & & & & & & & \\ (1-c)^2 & 0 & & & & & & & & \\ c(1-c) & c(1-c) & (1-c)^2 & & & & & & & \\ c(1-c) & c(1-c) & (1-c)^2 & 0 & & & & & & \\ 0 & c^2 & c(1-c) & c(1-c) & (1-c)^2 & & & & & \\ & c^2 & c(1-c) & c(1-c) & (1-c)^2 & 0 & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & c^2 & c(1-c) & c(1-c) & (1-c)^2 & 0 & & \\ 0 & & & & 0 & c^2 & c(1-c) & c(1-c) & (1-c)^2 & \end{bmatrix}$$

Note that M_{Mult}^i is a lower triangular matrix, and hence its eigenvalues $(\mu_{\text{Mult}}^i)_j$, $j = 1, 2, \dots, N - 2$, are given by the entries on the diagonal; i.e., they are equal to either 0 or $(1 - \lambda\alpha_i^+)^2$. With $0 < |\lambda\alpha_i^+| \leq 1$, the eigenvalues of M_{Mult}^i satisfy $|(\mu_{\text{Mult}}^i)_j| < 1$. Similar results are obtained for the scalar equations with negative characteristic speeds using Lemma A.4. Hence, the two level multiplicative scheme is convergent. We note that convergence can also be proved for the multilevel case, whose details are omitted due to space limitations.

For a k -level additive scheme, we have $p_i^{n+1} = M_{\text{Add}}^i p_i^n$, where M_{Add}^i is an $(N - 2^{k-1}) \times (N - 2^{k-1})$ iteration matrix constructed from Lemma A.5. M_{Add}^i is a lower triangular matrix with eigenvalues, $(\mu_{\text{Add}}^i)_j$, $j = 1, 2, \dots, N - 2^{k-1}$, equal to either 0 or $(1 - \lambda\alpha_i^+)$. The CFL condition $0 < |\lambda\alpha_i^+| \leq 1$ implies that $|(\mu_{\text{Add}}^i)_j| < 1$. Therefore, the multilevel additive scheme is convergent for all the scalar characteristic equations with positive characteristic speeds. It can be proved in a similar way that the scheme is convergent for the scalar equations with negative characteristic speeds using Lemma A.5. The multilevel additive scheme is convergent in the W -space, and hence convergent in the U -space. \square

5. Two-dimensional systems of hyperbolic conservation laws. A two-dimensional system of hyperbolic conservation laws can be written as

$$(5.1) \quad U_t + F(U)_x + G(U)_y = 0, \quad 0 < x, y < 1, t > 0,$$

where $F(U)$ and $G(U)$ are m -dimensional flux vectors in the x and y dimensions, respectively, and U is an m -dimensional vector of conserved quantities. We will apply the idea of the one-dimensional multigrid time stepping scheme to two dimensions. In particular, we will describe the details of the construction of the coarse grids, the interpolation, and the restriction operators.

Standard multigrid methods for two-dimensional problems use either the typical fully coarsened grid or a semicoarsened grid. There are a few inherent issues with these types of coarse grids for hyperbolic problems which may result in slow convergence of the multigrid method. Suppose we are solving the scalar wave equation $u_t + u_x = 0$ in two dimensions with initial conditions, as given in Figure 3(a), using a standard

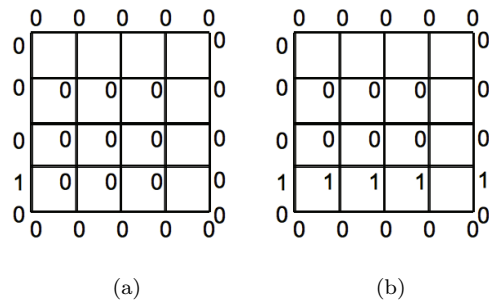


FIG. 3. Demonstration of the scenario in which the multigrid method using a standard fully coarsened grid would fail. (a) The initial and boundary conditions for $u_t + u_x = 0$. (b) Steady state solution is nonzero only along a noncoarse grid line.

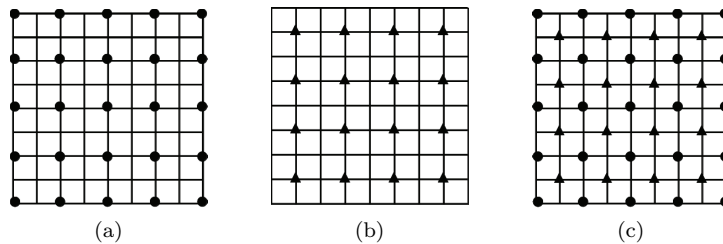


FIG. 4. The coarse grids used in the two-dimensional multigrid time stepping scheme. (a) The dots represent the coarse grid points on $\Omega^{1,H}$. (b) The triangles represent the coarse grid points on $\Omega^{2,H}$. (c) Ω^H , the grid obtained when the two coarse grids $\Omega^{1,H}$ and $\Omega^{2,H}$ are put together.

fully coarsened grid. Note that the steady state solution, shown in Figure 3(b), is nonzero only along a noncoarse grid line and is zero elsewhere. If injection is used for restriction of the solution, then this information on the noncoarse grid line will go completely unseen on the coarse grid. Hence, the coarse grid evolution of the solution would not help in accelerating the wave propagation. Full weighting restriction would spread out the solution so that the solution on the coarse grid is not zero, but it would still not resolve the fundamental issue.

One possible way to avoid the problem of missing information is to use a semicoarsened grid [33, 26]. However, with the use of explicit time marching schemes for hyperbolic equations, it is necessary to satisfy the CFL condition, which limits the size of the time step by the minimum of the spatial grid size. In semicoarsening, the spatial grid size increases by a factor of two along one dimension, while the spatial grid size in the other dimension remains unchanged from that of the fine grid. Hence, the time step size on the coarse grid cannot be larger than the time step size on the fine grid. As a result, using a semicoarsened grid would not accelerate wave propagation, as there is no increase in the size of the time step on the coarse grid.

To address this issue, we propose using two fully coarsened grids, as shown in Figures 4(a) and 4(b). Given a fine grid (x_j^h, y_l^h) , $j, l = 0, 1, \dots, N-1$, the coarse grid shown in Figure 4(a) is defined such that the grid points with even indices for j and l are selected as coarse grid points: $(x_j^{1,H}, y_l^{1,H})$, $j, l = 0, 2, \dots, N-1$. The other coarse grid, as shown in Figure 4(b), is defined such that the grid points with odd indices for j and l are selected as coarse grid points: $(x_j^{2,H}, y_l^{2,H})$, $j, l = 1, 3, \dots, N-2$. The superscripts h , $\{1, H\}$, and $\{2, H\}$ represent functions on the fine grid and the two coarse grids, respectively.

The grid size for the coarse grids $\Omega^{1,H}$ and $\Omega^{2,H}$ increases by a factor of two from that of the fine grid in both dimensions, as shown in Figures 4(a) and 4(b). This means that a time step of size $2\Delta t^h$ can be used for both of these coarse grids without violating the CFL condition, in contrast with the semicoarsening approach. Furthermore, from the combined coarse grid, Ω^H , shown in Figure 4(c), we can see that coarse grid points are selected in such a way that coarse grid points exist on every grid line in both the x and y directions. This ensures that there is no loss of information due to disturbances along the noncoarse grid lines, as in the case of full coarsening.

Further coarse grids are obtained by recursively applying the coarsening technique described above to each of the grids $\Omega^{1,H}$ and $\Omega^{2,H}$. Figure 5 shows the coarsening of the grid $\Omega^{1,H}$. The computational complexity of our approach is the same as that of a multigrid method using a semicoarsened grid.

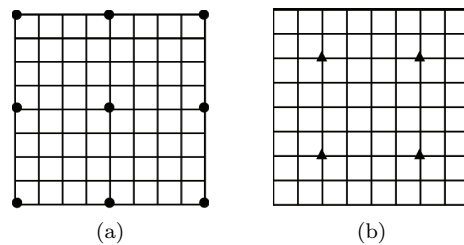


FIG. 5. The grids obtained from coarsening the grid $\Omega^{1,H}$.

5.1. Interpolation. In this section, we describe how a coarse grid solution is interpolated to the fine grid. At the coarse grid points, we just copy the values to the fine grid, as shown in Figure 4(c). For interpolation of the solution at the noncoarse grid points, (x_j^h, y_l^h) , we will apply the same idea as in one dimension. More precisely, we solve a local boundary value problem at (x_j^h, y_l^h) ; see Figure 6. Notice that all the neighboring grid points of (x_j^h, y_l^h) are coarse grid points whose values are known from the coarse grid solution. The interpolated value of $U_{j,l}^h$ is then given by the steady state solution of (5.1).

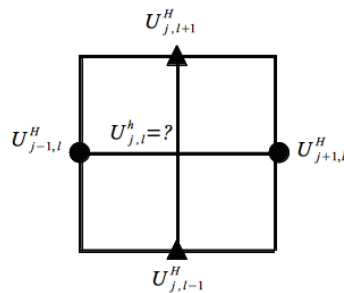


FIG. 6. Construction of a local boundary value problem for interpolation of the solution at (x_j^h, y_l^h) .

5.2. Residual restriction. The idea for residual restriction in two dimensions is the same as that described for one-dimensional systems in section 3.2. We will split the residual into forward and backward components and then perform upwind

Downloaded 11/19/14 to 129.97.140.11. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

restriction on the corresponding components accordingly. On the fine grid, for $j, l = 1, 2, \dots, N - 2$, the forward and backward components of the residual are defined as

$$(R^h)_{j,l}^+ = \frac{1}{\Delta x^h} [F_{j,l}^+(U_{j,l}^h) - F_{j-1,l}^+(U_{j-1,l}^h)] + \frac{1}{\Delta y^h} [F_{j,l}^+(U_{j,l}^h) - F_{j,l-1}^+(U_{j,l-1}^h)] - (B^h)_{j,l}^+,$$

$$(R^h)_{j,l}^- = \frac{1}{\Delta x^h} [F_{j+1,l}^-(U_{j+1,l}^h) - F_{j,l}^-(U_{j,l}^h)] + \frac{1}{\Delta y^h} [F_{j,l+1}^-(U_{j,l+1}^h) - F_{j,l}^-(U_{j,l}^h)] - (B^h)_{j,l}^-,$$

where B^h is the RHS on the fine grid, F^+ and F^- are the forward and backward flux components, and Δx^h and Δy^h are the fine grid sizes in the x and y dimensions, respectively. We note that $(R^h)_{j,l}^+ = (R^h)_{j,l}^- = 0$, $j, l = 0, N - 1$.

We first consider the restriction of the residual on the coarse grid, $\Omega^{1,H}$. For the forward component of the residual, the information is propagating in the positive x and y dimensions. We compute the residual on the coarse grid by averaging along the upwind directions:

$$(R^{1,H})_{j,l}^+ = 0.25[(R^h)_{j,l}^+ + (R^h)_{j-1,l}^+] + 0.25[(R^h)_{j,l}^+ + (R^h)_{j,l-1}^+], \quad j, l = 2, 4, \dots, N - 3.$$

Similarly, we compute the backward component of the residual on the coarse grid by averaging along the right and top sides:

$$(R^{1,H})_{j,l}^- = 0.25[(R^h)_{j,l}^- + (R^h)_{j+1,l}^-] + 0.25[(R^h)_{j,l}^- + (R^h)_{j,l+1}^-], \quad j, l = 2, 4, \dots, N - 3.$$

Combining the two components gives the coarse grid residual on $\Omega^{1,H}$:

$$R_{j,l}^{1,H} = (R^{1,H})_{j,l}^+ + (R^{1,H})_{j,l}^-, \quad j, l = 2, 4, \dots, N - 3.$$

For the other coarse grid, $\Omega^{2,H}$, the restriction of the residual is computed similarly as

$$R_{j,l}^{2,H} = (R^{2,H})_{j,l}^+ + (R^{2,H})_{j,l}^-, \quad j, l = 1, 3, \dots, N - 2,$$

where

$$(R^{2,H})_{j,l}^+ = 0.25[(R^h)_{j,l}^+ + (R^h)_{j-1,l}^+] + 0.25[(R^h)_{j,l}^+ + (R^h)_{j,l-1}^+],$$

$$(R^{2,H})_{j,l}^- = 0.25[(R^h)_{j,l}^- + (R^h)_{j+1,l}^-] + 0.25[(R^h)_{j,l}^- + (R^h)_{j,l+1}^-],$$

$$j, l = 1, 3, \dots, N - 2.$$

6. Numerical results. In this section, we demonstrate the effectiveness of our multigrid time stepping scheme numerically by several examples. The stopping criteria is that the residual norm is smaller than 10^{-6} . First we provide the results for a simple linear system and then for other nonlinear systems, including the Euler equations.

Example 6.1. We numerically verify that the multigrid time stepping schemes are nonoscillatory and have optimal speed of propagation in reaching the steady state solution. The linear system of hyperbolic conservation laws is given by

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} v \\ u \end{pmatrix}_x = 0, \quad 0 < x < 1, \quad t > 0.$$

The boundary condition is chosen as 0, and the initial condition is a square wave for both u and v . The steady state solution for this system is $(u, v) \equiv 0$, as given in Figure 7(d). We apply the three level multiplicative multigrid scheme to solve the system with a CFL number of 1. The plots for u at different multigrid steps are

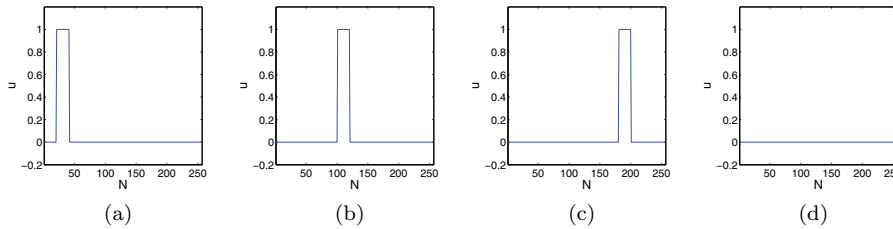


FIG. 7. The numerical solution given by the three-level multiplicative scheme for one-dimensional linear system, $\frac{\Delta t}{\Delta x} = 1$. (a) Multigrid step = 0, (b) multigrid step = 10, (c) multigrid step = 20, (d) multigrid step = 29.

shown in Figure 7 for the multiplicative scheme. It can be seen that the square wave remains a square wave as it propagates. We omit the results for v , as they are similar to the solution u . While it takes 236 iterations to propagate to the steady state on a fine grid with a grid size of $\Delta x^h = 1/256$, the three level algorithm converges in 29 iterations. Thus, the speed up of the three level method is 8.14. Our multigrid method achieves a speed up close to 2^L , where L is the number of multigrid levels, while the theoretical optimal speed up of a standard multigrid method was predicted to be $2^L - 1$ by [14, 5].

Example 6.2. We consider a system of nonlinear hyperbolic conservation laws described by

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} \frac{1}{2}u^2 + \frac{1}{2}v^2 \\ \frac{1}{2}v^2 + \frac{1}{2}uv \end{pmatrix}_x = 0, \quad 0 < x < 1, t > 0,$$

with the Neumann boundary condition and the initial condition

$$u(x, 0) = v(x, 0) = \begin{cases} 1 & \text{if } x = 0, \\ -1 & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The steady state solution is obtained as

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

Figure 8 shows the plots for u at different multigrid steps for a three level multiplicative scheme with a fine grid size of $\Delta x^h = 1/256$ and a CFL number of 0.1. The steady state solution is given by Figure 8(d). The solution v shows similar results which are not presented here. It is observed from Figure 8 that there are no oscillations in the solution, and the shock is resolved as a sharp discontinuity without any wiggles. This ensures an optimal speed of propagation of the wave, resulting in convergence in a small number of multigrid cycles.

The speed ups for different multigrid levels and different grid sizes for the multiplicative scheme are given in Table 1. The speed up consistently increases with the increase in the number of coarse grids used.

Example 6.3. The one-dimensional Euler equations of gas dynamics constitute a hyperbolic system of conservation laws given by

$$(6.1) \quad \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + P \\ u(E + P) \end{pmatrix}_x = 0,$$

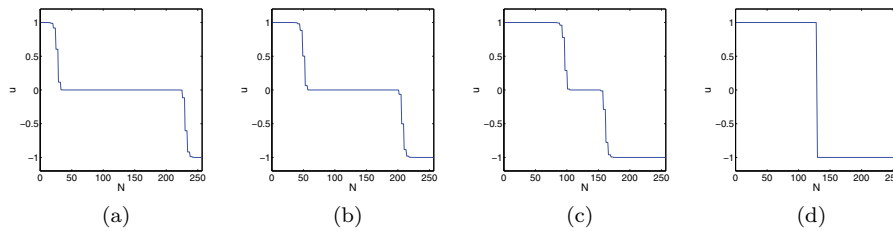


FIG. 8. The numerical solution given by the four level multiplicative scheme, $\frac{\Delta t}{\Delta x} = 0.1$. (a) Multigrid step = 20, (b) multigrid step = 40, (c) multigrid step = 80, (d) multigrid step = 120.

TABLE 1

Speed up for different multigrid levels and different grid spacings for a one-dimensional non-linear hyperbolic system.

	1/256	1/512	1/1024	1/2048	1/4096
L = 2	3.91	3.96	3.98	3.99	3.99
L = 3	10.98	11.25	11.44	11.51	11.56
L = 4	26.14	27.82	28.75	29.17	29.46
L = 5	56.83	63.10	66.84	68.91	69.99

where ρ , ρu , and E are conserved quantities of density, momentum, and energy. u is the velocity and P is the pressure. The equation of state is given by

$$(6.2) \quad P = (\gamma - 1)(E - 0.5\rho u^2), \quad \gamma = 1.4.$$

We compute the steady state solution for (6.1) using two types of initial value problems (Sod tube and Lax tube) and the Neumann boundary condition. The initial data for the Sod tube problem is given by

$$(\rho, \rho u, E)(x, 0) = \begin{cases} (1, 0, 2.5) & \text{if } x < \frac{1}{2}, \\ (0.125, 0, 0.25) & \text{if } x > \frac{1}{2}. \end{cases}$$

Figures 9, 10, and 11 show the density, velocity, and pressure plots, respectively, at different intermediate multigrid steps computed by a four level multiplicative scheme with a CFL number of 0.1 for a grid size of 1/1024. Note that the multiplicative scheme introduces minor oscillations. This behavior is expected, as the multiplicative scheme in general does not preserve monotonicity and is not TVD when the number of grids is greater than two [33]. However, these oscillations are very small and do not affect the speed of convergence of the multigrid algorithm, as shown in Table 2.

Example 6.4. We apply the multigrid schemes to (6.1) for the Lax tube problem, in which the initial conditions are given by

$$(\rho, \rho u, E)(x, 0) = \begin{cases} (0.445, 0.311, 8.928) & \text{if } x < \frac{1}{2}, \\ (0.5, 0, 1.4275) & \text{if } x > \frac{1}{2}. \end{cases}$$

The convergence results for the Lax tube problem for different levels of the multigrid method are given in Table 3. The speed up results again show that taking larger time steps on coarser grids results in faster multigrid convergence.

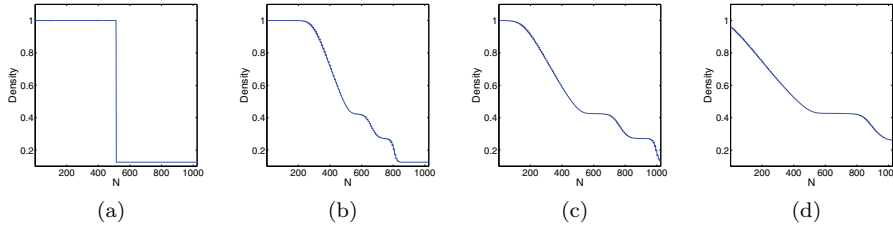


FIG. 9. Density plots for the Sod tube problem given by the four level multiplicative scheme, $\frac{\Delta t}{\Delta x} = 0.1$. (a) Multigrid step = 0, (b) multigrid step = 60, (c) multigrid step = 100, (d) multigrid step = 150.

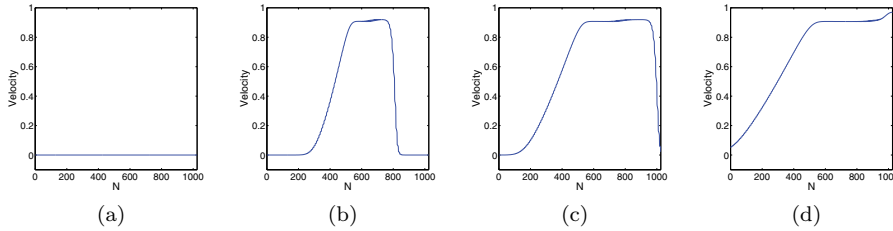


FIG. 10. Velocity plots for the Sod tube problem given by the four level multiplicative scheme, $\frac{\Delta t}{\Delta x} = 0.1$. (a) Multigrid step = 0, (b) multigrid step = 60, (c) multigrid step = 100, (d) multigrid step = 150.

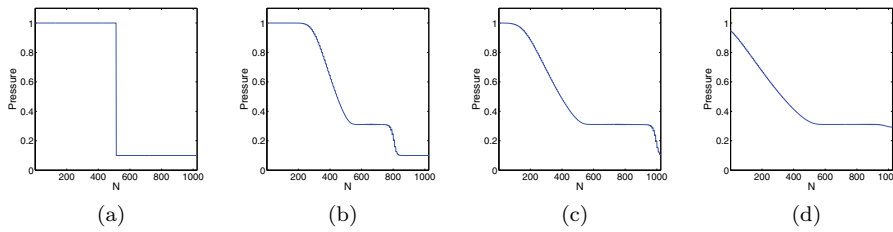


FIG. 11. Pressure plots for the Sod tube problem given by the four level multiplicative scheme, $\frac{\Delta t}{\Delta x} = 0.1$. (a) Multigrid step = 0, (b) multigrid step = 60, (c) multigrid step = 100, (d) multigrid step = 150.

TABLE 2
Speed up for the Sod tube problem.

	1/64	1/128	1/256	1/512	1/1024
L = 2	3.26	3.35	3.41	3.44	3.49
L = 3	8.39	8.17	8.63	8.80	8.98
L = 4	19.40	18.00	18.83	19.82	20.33
L = 5	42.73	37.24	38.11	40.83	42.87

TABLE 3
Speed up table for the Lax tube problem.

	1/64	1/128	1/256	1/512	1/1024
L = 2	3.86	3.89	3.89	3.91	3.95
L = 3	9.77	9.99	10.00	9.93	10.23
L = 4	21.34	22.23	22.57	22.31	22.37
L = 5	41.59	45.75	47.79	47.57	47.37

Downloaded 11/19/14 to 129.97.140.11. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Example 6.5. The two-dimensional Euler equations constitute a system of hyperbolic conservation laws described by

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ u(E + P) \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ v(E + P) \end{pmatrix}_y = 0,$$

and the two-dimensional equation of state is given by

$$P = (\gamma - 1)(E - 0.5\rho(u^2 + v^2)), \quad \gamma = 1.4.$$

We conduct our experiments on Riemann problems for two-dimensional gas dynamics in which the initial data are constant in each quadrant [23]. The initial data are restricted so that the only types of behavior allowed at interfaces are a one-dimensional elementary wave, a one-dimensional shock, a one-dimensional rarefaction, and a contact discontinuity. There are a total of 19 genuinely different configurations for a polytropic gas [23]. We test our algorithm on six different initial configurations.

The rectangular domain for the computations is $\{(x, y) | 0 < x, y < 1\}$. The initial conditions are listed as (P_i, ρ_i, u_i, v_i) , where i refers to the quadrant number. The initial data for the six configurations are given in Table 4. Here C denotes configuration, and the numbers listed for each configuration refer to those used in [23].

TABLE 4
Initial data for the six configurations for two-dimensional Euler equations.

Quadrant 2: (P_2, ρ_2, u_2, v_2) C1: (0.4, 0.5197, -0.7259, 0) C2: (0.4, 0.5197, -0.7259, 0) C4: (0.35, 0.5065, 0.8939, 0) C5: (1, 2, -0.75, 0.5) C9: (1, 2, 0, -0.3) C11: (0.4, 0.5313, 0.8276, 0)	Quadrant 1: (P_1, ρ_1, u_1, v_1) C1: (1, 1, 0, 0) C2: (1, 1, 0, 0) C4: (1.1, 1.1, 0, 0) C5: (1, 1, -0.75, -0.5) C9: (1, 1, 0, 0.3) C11: (1, 1, 0.1, 0)
Quadrant 3: (P_3, ρ_3, u_3, v_3) C1: (0.0439, 0.1072, -0.7259, -1.4045) C2: (1, 1, -0.7259, -0.7259) C4: (1.1, 1.1, 0.8939, 0.8939) C5: (1, 1, 0.75, 0.5) C9: (0.4, 1.039, 0, -0.8133) C11: (0.4, 0.8, 0.1, 0)	Quadrant 4: (P_4, ρ_4, u_4, v_4) C1: (0.15, 0.2579, 0, -1.4045) C2: (0.4, 0.5197, 0, -0.7259) C4: (0.35, 0.5065, 0, 0.8939) C5: (1, 3, 0.75, -0.5) C9: (0.4, 0.5197, 0, -0.4259) C11: (0.4, 0.5313, 0.1, 0.7276)

We apply the multiplicative multigrid scheme to compute the steady state solution of the Euler equations. Figures 12 and 13 demonstrate the propagation of the density and velocity, respectively, using a three level multiplicative scheme on a fine grid size of 65×65 for the initial configuration 4. It can be seen that the solution converges smoothly to the steady state without any major oscillations. The steady state solutions for density and velocity are shown in Figures 12(d) and 13(d), respectively. On a single grid, the FVS scheme (2.2) takes 4340 iterations to converge to the steady state. The three level multiplicative scheme takes 510 multigrid cycles

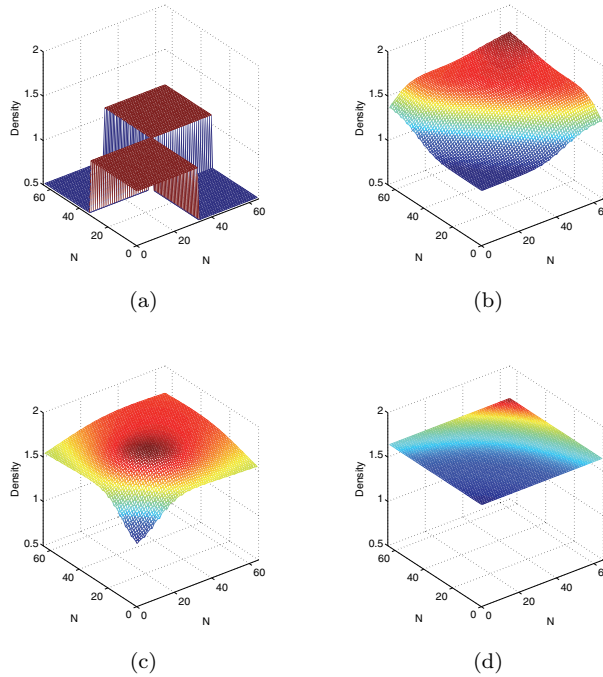


FIG. 12. Density plots for two-dimensional Euler equations (configuration 4) given by the three level multiplicative scheme, $\lambda = 0.1$. (a) Multigrid step = 0, (b) multigrid step = 50, (c) multigrid step = 100, (d) multigrid step = 510.

giving a speed up of 8.51. In Table 5, we present the speed up for different levels of the multiplicative method for the six initial configurations for a fine grid size of 65×65 . While the theoretical optimal speed up of an L grid method was computed asymptotically as $2^L - 1$ [14, 5], in practice our multigrid scheme achieves speed up much higher than the theoretical optimal for certain initial configurations, as shown in Table 5.

As a comparison, Kanarachos and Vournas [18] developed a multigrid method for the Euler equations using full weighting and bilinear interpolation techniques. With three grid levels, they obtained a reduction in the number of work units by a factor of 5–9 (compared to a single grid), while our upwind biased multigrid technique gives a factor of 4–13 for the different initial configurations.

7. Conclusion. As with the scalar conservation laws, we have shown that the upwind biased restriction and interpolation techniques result in efficient multigrid methods for systems of hyperbolic conservation laws as well. Although the systems are a mixture of characteristics, we can still exploit the direction of propagation of information for the restriction and interpolation operations. The interpolation is formulated as solving a local Riemann problem to steady state, and we perform upwind restriction by splitting the residual into positive and negative components based on the idea of flux vector splitting. We theoretically proved that both the additive and multiplicative multigrid schemes satisfy the TVD property for one-dimensional linear systems. We also proved that these schemes are consistent and convergent for one-dimensional linear systems. We showed that the upwind techniques can be extended

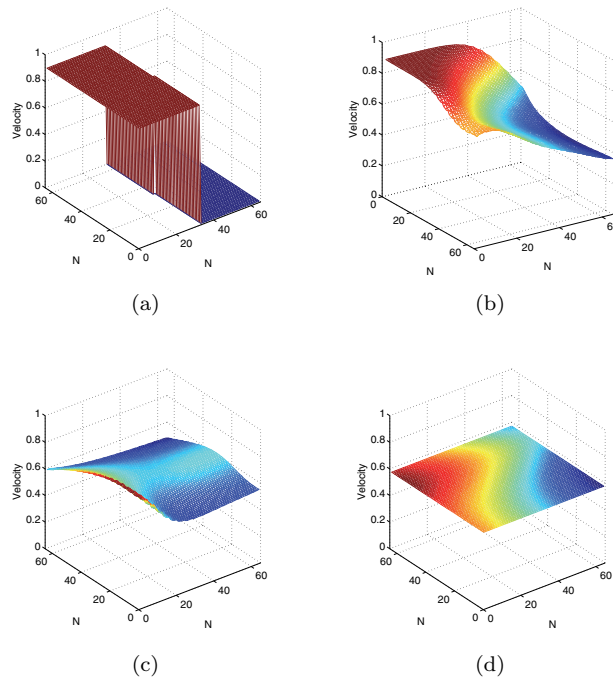


FIG. 13. Velocity plots for two-dimensional Euler equations (configuration 4) given by the three level multiplicative scheme, $\lambda = 0.1$. (a) Multigrid step = 0, (b) multigrid step = 50, (c) multigrid step = 100, (d) multigrid step = 510.

TABLE 5

Speed ups of multiplicative scheme for the six initial configurations for two-dimensional Euler equations for a fine grid size of 65×65 .

	C1	C2	C4	C5	C9	C11
L = 2	3.22	4.19	3.71	4.35	4.76	3.87
L = 3	9.11	12.04	8.51	6.13	22.43	7.83
L = 4	18.69	25.78	12.88	22.44	40.37	25.58

to higher dimensional systems and also introduced a novel coarsening technique. The numerical experiments for both one- and two-dimensional hyperbolic systems have consistently shown that our multigrid methods result in smooth and fast convergence to the steady state solution. The upwind multigrid methods reduce the number of work units by a factor of 4–13 (compared to the single grid), while factors of 3–9 are reported in the literature.

Appendix A. Linear analysis.

A.1. Monotonicity analysis for the linear wave equation. We present the monotonicity preserving properties of the multigrid time stepping schemes for solving the linear wave equation

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

DEFINITION A.1. Suppose u^n is a nonincreasing (nondecreasing) function. Let u^{n+1} be computed by one multigrid cycle. Then the multigrid method is said to preserve monotonicity if u^{n+1} is also nonincreasing (nondecreasing).

THEOREM A.2. Both the two level multiplicative and additive multigrid time stepping schemes preserve monotonicity for the linear wave equation.

Proof. Given u^n , let u^{n+1} be the solution after one iteration of the two level multiplicative scheme. We assume a Dirichlet boundary condition as given in (A.1). For $j = 2, 4, \dots, N - 1$, u_j^{n+1} and u_{j-1}^{n+1} are computed using (3.5) and (3.9),

$$(A.2) \quad u_j^{n+1} = \tilde{u}_j^H = \bar{u}_j^h - \lambda(\bar{u}_j^h - \bar{u}_{j-2}^h),$$

$$(A.3) \quad u_{j-1}^{n+1} = \tilde{u}_{j-2}^H = \bar{u}_{j-2}^h - \lambda(\bar{u}_{j-2}^h - \bar{u}_{j-4}^h),$$

where \bar{u}_j^h is the solution after fine grid upwind smoothing:

$$(A.4) \quad \bar{u}_j^h = u_j^n - \lambda(u_j^n - u_{j-1}^n).$$

Subtracting (A.3) from (A.2),

$$(A.5) \quad \begin{aligned} u_j^{n+1} - u_{j-1}^{n+1} &= \bar{u}_j^h - \lambda(\bar{u}_j^h - \bar{u}_{j-2}^h) - \bar{u}_{j-2}^h + \lambda(\bar{u}_{j-2}^h - \bar{u}_{j-4}^h) \\ &= (1 - \lambda)\bar{u}_j^h + (2\lambda - 1)\bar{u}_{j-2}^h - \lambda\bar{u}_{j-4}^h. \end{aligned}$$

After substituting (A.4) into (A.5) and simplifying, we have

$$(A.6) \quad \begin{aligned} u_j^{n+1} - u_{j-1}^{n+1} &= (1 - \lambda)^2(u_j^n - u_{j-1}^n) + (1 - \lambda)(u_{j-1}^n - u_{j-2}^n) \\ &\quad + 2\lambda(1 - \lambda)(u_{j-2}^n - u_{j-3}^n) + \lambda(u_{j-3}^n - u_{j-4}^n) \\ &\quad + \lambda^2(u_{j-4}^n - u_{j-5}^n). \end{aligned}$$

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

$$(A.7) \quad u_{j-1}^{n+1} - u_{j-2}^{n+1} = \bar{u}_{j-2}^H - \bar{u}_{j-2}^H = 0.$$

Thus, given that $0 \leq \lambda \leq 1$ (CFL condition), u^{n+1} is nonincreasing (nondecreasing) if u^n is nonincreasing (nondecreasing).

Following similar calculations for the additive scheme, the solution when $j = 2, 4, \dots, N - 1$ is given by

$$(A.8) \quad u_j^{n+1} = \tilde{u}_j^H = u_j^n - \lambda(u_j^n - u_{j-2}^n),$$

$$(A.9) \quad u_{j-1}^{n+1} = \tilde{u}_{j-2}^H = u_{j-2}^n - \lambda(u_{j-2}^n - u_{j-4}^n).$$

After simplification,

$$(A.10) \quad \begin{aligned} u_j^{n+1} - u_{j-1}^{n+1} &= (1 - \lambda)(u_j^n - u_{j-1}^n) + (1 - \lambda)(u_{j-1}^n - u_{j-2}^n) \\ &\quad + \lambda(u_{j-2}^n - u_{j-3}^n) + \lambda(u_{j-3}^n - u_{j-4}^n), \end{aligned}$$

$$(A.11) \quad u_{j-1}^{n+1} - u_{j-2}^{n+1} = 0.$$

Thus, the additive scheme also preserves monotonicity for $0 \leq \lambda \leq 1$. \square

A.2. Linear systems. We derive expressions required for the analysis of the TVD, consistency, and convergence properties in section 4.

Let \tilde{K}_1 be an $m \times m_1$ block matrix containing the eigenvectors corresponding to α^+ , and let \tilde{K}_2 be an $m \times m_2$ block matrix containing the eigenvectors corresponding to α^- . Then we write K as

$$(A.12) \quad K = \begin{bmatrix} \tilde{K}_1 & \tilde{K}_2 \end{bmatrix}.$$

Similarly, we write the matrix K^{-1} in terms of another set of block matrices as

$$(A.13) \quad K^{-1} = \begin{bmatrix} \hat{K}_1 \\ \hat{K}_2 \end{bmatrix},$$

where \hat{K}_1 is an $m_1 \times m$ block matrix and \hat{K}_2 is an $m_2 \times m$ block matrix.

LEMMA A.3. For any $m \times m$ invertible matrix K , the following identities hold:

$$\begin{bmatrix} \hat{K}_1 \tilde{K}_1 & \hat{K}_1 \tilde{K}_2 \\ \hat{K}_2 \tilde{K}_1 & \hat{K}_2 \tilde{K}_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

LEMMA A.4. Given U_j^n and a zero boundary condition, the analytical expressions for p^{n+1} and q^{n+1} given by the two level multiplicative scheme are

$$(A.14) \quad \begin{aligned} p_j^{n+1} &= (1 - \lambda\alpha^+)^2 p_j^n + \lambda\alpha^+(1 - \lambda\alpha^+) p_{j-1}^n + \lambda\alpha^+(1 - \lambda\alpha^+) p_{j-2}^n \\ &\quad + (\lambda\alpha^+)^2 p_{j-3}^n, \quad j = 2, 4, \dots, N-3, \end{aligned}$$

$$(A.15) \quad \begin{aligned} p_{j-1}^{n+1} &= (1 - \lambda\alpha^+)^2 p_{j-2}^n + \lambda\alpha^+(1 - \lambda\alpha^+) p_{j-3}^n + \lambda\alpha^+(1 - \lambda\alpha^+) p_{j-4}^n \\ &\quad + (\lambda\alpha^+)^2 p_{j-5}^n, \quad j = 2, 4, \dots, N-1, \end{aligned}$$

$$(A.16) \quad \begin{aligned} q_j^{n+1} &= (1 + \lambda\alpha^-)^2 q_j^n - \lambda\alpha^-(1 + \lambda\alpha^-) q_{j+1}^n - \lambda\alpha^-(1 + \lambda\alpha^-) q_{j+2}^n \\ &\quad + (\lambda\alpha^-)^2 q_{j+3}^n, \quad j = 2, 4, \dots, N-3, \end{aligned}$$

$$(A.17) \quad \begin{aligned} q_{j-1}^{n+1} &= (1 + \lambda\alpha^-)^2 q_{j-2}^n - \lambda\alpha^-(1 + \lambda\alpha^-) q_{j-1}^n - \lambda\alpha^-(1 + \lambda\alpha^-) q_{j+2}^n \\ &\quad + (\lambda\alpha^-)^2 q_{j+3}^n, \quad j = 2, 4, \dots, N-1. \end{aligned}$$

Proof. Given U_j^n and a zero boundary condition, we note that $U_j^n = 0$ for any $j \leq 0$ and $j \geq N-1$. Let U_j^{n+1} , $j = 1, 2, \dots, N-2$, be the solution obtained after one iteration of a two level multiplicative scheme. For the grid points which coincide with the coarse grid points, i.e., $j = 2, 4, \dots, N-3$, U_j^{n+1} is obtained from Algorithm 1 as

$$(A.18) \quad U_j^{n+1} = \tilde{U}_j^H,$$

where \tilde{U}_j^H is the solution after coarse grid evolution. For x_{j-1}^{n+1} , $j = 2, 4, \dots, N-1$, the interpolated solution is the steady state solution of the local Riemann problem using \tilde{U}_{j-2}^H and \tilde{U}_j^H as the boundary values, as described in section 3. Analytically, the steady state solution is not apparent from the U -space, as the system consists of a mixture of characteristics. However, for constant coefficient systems, the steady state solution in the U -space is equivalent to the steady state solution in the W -space. For the characteristic equations associated with p , the sign of the characteristic speed is positive; i.e., the wave is moving to the right, and hence the interpolated value at

x_{j-1}^{n+1} is \tilde{p}_{j-2}^H . In contrast, the sign of the characteristic speed for equations associated with q is negative, and hence $q_{j-1}^{n+1} = \tilde{q}_j^H$. Using the relations $W = K^{-1}U$ and (A.13), we have

$$W_{j-1}^{n+1} = \begin{bmatrix} \tilde{p}_{j-2}^H \\ \tilde{q}_j^H \end{bmatrix} = \begin{bmatrix} \widehat{K}_1 \tilde{U}_{j-2}^H \\ \widehat{K}_2 \tilde{U}_j^H \end{bmatrix}.$$

U_{j-1}^{n+1} is then computed as KW_{j-1}^{n+1} ,

$$\begin{aligned} U_{j-1}^{n+1} &= \begin{bmatrix} \tilde{K}_1 & \tilde{K}_2 \end{bmatrix} \begin{bmatrix} \widehat{K}_1 \tilde{U}_{j-2}^H \\ \widehat{K}_2 \tilde{U}_j^H \end{bmatrix} \\ (A.19) \quad &= \tilde{K}_1 \widehat{K}_1 \tilde{U}_{j-2}^H + \tilde{K}_2 \widehat{K}_2 \tilde{U}_j^H. \end{aligned}$$

For the multiplicative scheme, (A.18) is rewritten using (2.3) as

$$\begin{aligned} U_j^{n+1} &= (I - \lambda A^+ + \lambda A^-)U_j^n + \lambda A^+(I - \lambda A^+)U_{j-1}^n + \lambda A^+(I - \lambda A^+)U_{j-2}^n \\ &\quad + (\lambda A^+)^2 U_{j-3}^n - \lambda A^-(I + \lambda A^-)U_{j+1}^n - \lambda A^-(I + \lambda A^-)U_{j+2}^n \\ (A.20) \quad &\quad + (\lambda A^-)^2 U_{j+3}^n. \end{aligned}$$

Multiplying both sides of (A.20) by $K^{-1}A$, we obtain a decoupled equation in characteristic variables,

$$\begin{aligned} \Lambda W_j^{n+1} &= (\Lambda - 2\lambda(\Lambda^+)^2 + \lambda^2(\Lambda^+)^3 + 2\lambda(\Lambda^-)^2 + \lambda^2(\Lambda^-)^3)W_j^n \\ &\quad + \lambda(\Lambda^+)^2(I - \lambda\Lambda^+)W_{j-1}^n - \lambda(\Lambda^-)^2(I + \lambda\Lambda^-)W_{j+1}^n \\ &\quad + \lambda(\Lambda^+)^2(I - \lambda\Lambda^+)W_{j-2}^n + \lambda^2(\Lambda^+)^3W_{j-3}^n - \lambda(\Lambda^-)^2(I + \lambda\Lambda^-)W_{j+2}^n \\ (A.21) \quad &\quad + \lambda^2(\Lambda^-)^3W_{j+3}^n. \end{aligned}$$

We rewrite (A.21) as two separate equations, one for the vector $p = [p_1, p_2, \dots, p_{m_1}]$ associated with positive eigenvalues and the other for vector $q = [q_{m_1+1}, q_{m_1+2}, \dots, q_m]$ associated with negative eigenvalues, as given in (A.14) and (A.16), respectively.

Similarly, for the odd grid points, we multiply both sides of (A.19) by $K^{-1}A$ and simplify using Lemma A.3 and (2.3) as

$$\begin{aligned} \Lambda W_{j-1}^{n+1} &= \Lambda^+(I - \lambda\Lambda^+)^2 W_{j-2}^n + \lambda(\Lambda^+)^2(I - \lambda\Lambda^+)W_{j-3}^n + \lambda(\Lambda^+)^2(I - \lambda\Lambda^+)W_{j-4}^n \\ &\quad + \lambda^2(\Lambda^+)^3 W_{j-5}^n + \Lambda^-(I + \lambda\Lambda^-)^2 W_j^n - \lambda(\Lambda^-)^2(I + \lambda\Lambda^-)W_{j+1}^n \\ (A.22) \quad &\quad - \lambda(\Lambda^-)^2(I + \lambda\Lambda^-)W_{j+2}^n + \lambda^2(\Lambda^-)^3 W_{j+3}^n. \end{aligned}$$

Rewriting the decoupled set of m scalar equations (A.22) as two separate equations, we have (A.15) and (A.17), respectively. \square

LEMMA A.5. *Given U_j^n and a zero boundary condition, the analytical expressions*

for p^{n+1} and q^{n+1} given by a k -level additive scheme are

$$(A.23) \quad p_j^{n+1} = (1 - \lambda\alpha^+)p_j^n + \lambda\alpha^+p_{j-2^{k-1}}^n,$$

$$(A.24) \quad \left. \begin{array}{l} p_{j-1}^{n+1} \\ p_{j-2}^{n+1} \\ \vdots \\ p_{j-2^{k-1}+1} \end{array} \right\} = (1 - \lambda\alpha^+)p_{j-2^{k-1}}^n + \lambda\alpha^+p_{j-2 \cdot 2^{k-1}}^n,$$

$$(A.25) \quad q_j^{n+1} = (1 + \lambda\alpha^-)q_j^n - \lambda\alpha^-q_{j+2^{k-1}}^n,$$

$$(A.26) \quad \left. \begin{array}{l} q_{j-1}^{n+1} \\ q_{j-2}^{n+1} \\ \vdots \\ q_{j-2^{k-1}+1} \end{array} \right\} = (1 + \lambda\alpha^-)q_j^n - \lambda\alpha^-q_{j+2 \cdot 2^{k-1}}^n,$$

$$j = 2^{k-1}, 2 \cdot 2^{k-1}, \dots$$

Proof. For a k -level additive scheme and $j = 2^{k-1}, 2 \cdot 2^{k-1}, \dots$, the solution after the coarse grid evolution is

$$\tilde{U}_j^{(k)} = (I - \lambda A^+ + \lambda A^-)U_j^n + \lambda A^+U_{j-2^{k-1}}^n - \lambda A^+U_{j+2^{k-1}}^n.$$

The interpolated solution on level $(k-1)$ is given by

$$\begin{aligned} \tilde{U}_j^{(k-1)} &= \tilde{U}_j^{(k)}, \\ \tilde{U}_{j-2^{k-2}}^{(k-1)} &= \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-1}}^{(k)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_j^{(k)}. \end{aligned}$$

Similarly, for level $(k-2)$, we have

$$\begin{aligned} \tilde{U}_j^{(k-2)} &= \tilde{U}_j^{(k-1)} &&= \tilde{U}_j^{(k)}, \\ \tilde{U}_{j-2^{k-3}}^{(k-2)} &= \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-2}}^{(k-1)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_j^{(k-1)} &&= \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-1}}^{(k)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_j^{(k)}, \\ \tilde{U}_{j-2^{k-2}}^{(k-2)} &= \tilde{U}_{j-2^{k-2}}^{(k-1)} &&= \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-1}}^{(k)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_j^{(k)}, \\ \tilde{U}_{j-3 \cdot 2^{k-3}}^{(k-2)} &= \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-1}}^{(k-1)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_{j-2^{k-2}}^{(k-1)} &&= \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-1}}^{(k)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_j^{(k)}. \end{aligned}$$

By induction, the solution on the fine grid, U_j^{n+1} is given by

$$(A.27) \quad U_j^{n+1} = \tilde{U}_j^{(k)},$$

$$(A.28) \quad \left. \begin{array}{l} U_{j-1}^{n+1} \\ U_{j-2}^{n+1} \\ \vdots \\ U_{j-2^{k-1}+1} \end{array} \right\} = \tilde{K}_1 \hat{K}_1 \tilde{U}_{j-2^{k-1}}^{(k)} + \tilde{K}_2 \hat{K}_2 \tilde{U}_j^{(k)}.$$

Multiplying both sides of (A.27) and (A.28) by $K^{-1}A$ and simplifying reduces the

system to a set of m -decoupled equations,

$$\begin{aligned} \Lambda W_j^{n+1} &= (I - \lambda \Lambda^+ + \lambda \Lambda^-) W_j^n + \lambda \Lambda^+ W_{j-2^{k-1}}^n - \lambda \Lambda^- W_{j+2^{k-1}}^n, \\ \left. \begin{array}{l} \Lambda W_{j-1}^{n+1} \\ \Lambda W_{j-2}^{n+1} \\ \vdots \\ \Lambda W_{j-2^{k-1}+1} \end{array} \right\} &= \Lambda^+ (I - \Lambda^+) W_{j-2^{k-1}}^n + \lambda (\Lambda^+)^2 W_{j-2 \cdot 2^{k-1}}^n + \Lambda^- (I + \lambda \Lambda^-) W_j^n \\ &\quad - \lambda (\Lambda^-)^2 W_{j+2 \cdot 2^{k-1}}^n. \end{aligned}$$

We rewrite these decoupled equations as two separate sets of equations: one for all the characteristic equations with positive sign for the characteristic speeds ((A.23) and (A.24)), and the other for the characteristic equations with negative sign for the characteristic speeds ((A.25) and (A.26)). \square

REFERENCES

- [1] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
- [2] R. V. CHIMA AND G. M. JOHNSON, *Efficient solution of the Euler and Navier-Stokes equations with a vectorized multiple-grid algorithm*, AIAA J., 23 (1985), pp. 23–32.
- [3] E. DICK, *Multigrid acceleration of a flux-difference splitting method for steady Euler equations*, J. Comput. Appl. Math., 28 (1989), pp. 173–180.
- [4] F. GRASSO AND M. MARINI, *Solutions of hypersonic viscous flows with total variation diminishing multigrid techniques*, Comput. & Fluids, 24 (1995), pp. 571–592.
- [5] B. GUSTAFSSON AND P. LÖTSTEDT, *Analysis of the multigrid method applied to first order systems*, in Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods, J. Mandel et al., eds., SIAM, Philadelphia, 1989, pp. 181–233.
- [6] W. HACKBUSH, *Multigrid Methods and Applications*, Springer, Berlin, Heidelberg, New York, 1985.
- [7] A. HARTEN, *High resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 135 (1997), pp. 260–278.
- [8] P. W. HEMKER AND S. P. SPEKREIJSE, *Multigrid solution of the steady Euler equations*, in Advances in Multigrid Methods, Notes Numer. Fluid Mech. 11, D. Braess, W. Hackbusch, and U. Trottenberg, eds., Vieweg, Braunschweig, 1985, pp. 33–44.
- [9] P. W. HEMKER AND S. P. SPEKREIJSE, *Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations*, Appl. Numer. Math., 2 (1986), pp. 475–493.
- [10] A. JAMESON, *Monotonicity Preserving and Total Variation Diminishing Multigrid Time Stepping Methods*, manuscript, 2001.
- [11] A. JAMESON, *Solution of the Euler equations for two-dimensional transonic flow by a multigrid method*, Appl. Math. Comput., 13 (1983), pp. 327–355.
- [12] A. JAMESON AND S. YOON, *Multigrid solution of the Euler equations using implicit schemes*, in Proceedings of the AIAA 23rd Aerospace Sciences Meeting (Reno, NV), American Institute of Aeronautics and Astronautics, New York, 1985, AIAA-85-0293.
- [13] D. C. JESPERSEN, *Design and implementation of a multigrid code for the Euler equations*, Appl. Math. Comput., 13 (1983), pp. 357–374.
- [14] D. C. JESPERSEN, *Recent Developments in Multigrid Methods for the Steady Euler Equations*, Lecture Notes, von Karman Institute, Rhode-St.Genese, Belgium, 1984.
- [15] D. C. JESPERSEN, *A time-accurate multiple-grid algorithm*, in Proceedings of the 7th Computational Fluid Dynamics Conference (Cincinnati, OH), American Institute of Aeronautics and Astronautics, New York, 1985, pp. 58–66.
- [16] G. M. JOHNSON, *Multiple-Grid Acceleration of Lax-Wendroff Algorithms*, Technical Report, NASA, Cleveland, OH, 1982.
- [17] G. M. JOHNSON, *Accelerated solution of the steady Euler equations*, in Advances in Computational Transonics, Recent Adv. Num. Methods Fluids 4, W. G. Habashi, ed., Pineridge, Swansea, 1985, pp. 473–501.
- [18] A. KANARACHOS AND I. VOURNAS, *Multigrid technique applied to the compressible Euler equations*, Acta Mech., 97 (1993), pp. 23–40.

- [19] B. KOREN AND P. W. HEMKER, *Damped, direction-dependent multigrid for hypersonic flow computations*, Appl. Numer. Math., 7 (1991), pp. 309–328.
- [20] A. KURGANOV AND E. TADMOR, *New high resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys., 160 (2000), pp. 241–282.
- [21] M. P. LECLERCQ AND B. STOUFFLET, *Characteristic multigrid method application to solve the Euler equations with unstructured and unnested grids*, J. Comput. Phys., 104 (1993), pp. 329–346.
- [22] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK, 2004.
- [23] X.-D. LIU AND P. D. LAX, *Positive schemes for solving multi-dimensional hyperbolic systems of conservation laws*, J. Comput. Phys., 187 (2003), pp. 428–440.
- [24] P. LÖTSTEDT AND B. GUSTAFSSON, *Fourier analysis of multigrid methods for general systems of PDEs*, Math. Comp., 60 (1993), pp. 473–493.
- [25] W. A. MULDER, *Multigrid relaxation for the Euler equations*, J. Comput. Phys., 60 (1985), pp. 235–252.
- [26] W. A. MULDER, *A new multigrid approach to convection problems*, J. Comput. Phys., 83 (1989), pp. 303–323.
- [27] R. H. NI, *A multiple-grid scheme for solving the Euler equations*, in Proceedings of the 5th Computational Fluid Dynamics Conference (Palo Alto, CA), American Institute of Aeronautics and Astronautics, New York, 1981, pp. 257–264.
- [28] S. OSHER AND F. SOLOMON, *Upwind difference schemes for hyperbolic systems of conservation laws*, Math. Comp., 38 (1982), pp. 339–374.
- [29] S. SPEKREIJSE, *Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws*, Math. Comp., 49 (1987), pp. 135–155.
- [30] J. L. STEGER, *A Preliminary Study of Relaxation Methods for the Inviscid Conservative Gas Dynamics Equations Using Flux Splitting*, Contractor Report 3415, NASA, Cleveland, OH, 1981.
- [31] P. K. SWEBY, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal., 21 (1984), pp. 995–1011.
- [32] E. F. TORO, *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction*, 3rd ed., Springer-Verlag, Berlin, 2009.
- [33] J. W. L. WAN AND A. JAMESON, *Monotonicity preserving multigrid time stepping schemes for conservation laws*, Comput. Vis. Sci., 11 (2007), pp. 41–58.
- [34] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley and Sons, New York, 1992.