MULTIGRID REDUCTION IN TIME WITH ADAPTIVE SPATIAL COARSENING FOR THE LINEAR ADVECTION EQUATION [*]

A. J. M. HOWSE [†]

In collaboration with: H. De Sterck, R. D. Falgout, S. P. MacLachlan, J. B. Schroder

Abstract. We apply a multigrid reduction-in-time algorithm to hyperbolic partial differential equations. This study is motivated by the observation that sequential time stepping is an obvious computational bottleneck when attempting to implement highly concurrent algorithms, thus parallel-in-time methods are particularly desirable. In the case of explicit time stepping, spatial coarsening may be necessary to ensure that stability conditions are satisfied on all levels, and it may be useful for implicit time stepping by producing cheaper multigrid cycles. Unfortunately, classical spatial coarsening results in extremely slow convergence when the wave speed is near zero, even if only locally. We present an adaptive spatial coarsening strategy that addresses this issue for 1D linear advection and implicit time stepping. Numerical results show that this offers significant improvements over classical coarsening. Future improvements and extensions to explicit time stepping are discussed.

Key words. Adaptive spatial coarsening, multigrid reduction in time, MGRIT, parallel-in-time, hyperbolic problems, XBraid

1. Introduction.

Due to stagnating processor speeds and increasing core counts, the current paradigm of high performance computing is to achieve shorter computing times by increasing the concurrency of computations. Time integration represents an obvious bottleneck for achieving greater speedup due to the sequential nature of many numerical integration schemes. While temporal parallelism may seem counterintuitive, the development of parallel-in-time methods is an active area of research, with a history spanning several decades [9]. Variants include direct methods and iterative methods based on deferred corrections [5], domain decomposition [11], multigrid [12], multiple shooting [2], and waveform relaxation [16] approaches. For instance, one of the most prominent parallel-in-time methods, parareal [15], is equivalent to a two-level multigrid scheme [10].

In this paper, we discuss the multigrid reduction-in-time (MGRIT) method [7] and use XBraid [1], an open-source implementation of MGRIT. A strength of the MGRIT framework is its non-intrusive nature, which allows existing time-stepping routines to be used within the MGRIT implementation. Thus far, MGRIT has been successfully implemented using time stepping routines for linear [7] and nonlinear [8] parabolic partial differential equations (PDEs) in multiple dimensions, computational fluid dynamics problems [6], and power system models [14]. We now consider applying MGRIT to hyperbolic PDEs.

As a multigrid method, MGRIT primarily involves temporal coarsening, but spatial coarsening may be necessary for explicit time integration to ensure that stability conditions are satisfied on all levels of the grid hierarchy. Spatial coarsening may also be used with implicit time integration to produce smaller coarse grid problems and, hence, cheaper multigrid cycles. However, small local Courant numbers induce a sort of anisotropy in the discrete equations, meaning that the nodal connections in space are small compared to those in time. These so-called weak connections prevent

[*] This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-PROC-716758).

[†] University of Waterloo, Canada (ahowse@uwaterloo.ca).

This manuscript is for review purposes only.

pointwise relaxation from smoothing the error in space, thus inhibiting the effectiveness of spatial coarsening and leading to slow convergence. In this paper, we present an adaptive spatial coarsening strategy for implicit time stepping that resolves this problem for the conservative variable coefficient linear advection equation,

$$\partial_t u + \partial_x(a(x,t)u) = 0, \tag{1}$$

by locally preventing coarsening in regions with near zero Courant numbers.

The remainder of this paper is as follows. In §2, we describe the MGRIT algorithm and discuss results for classical spatial coarsening. In §3, we present our adaptive coarsening approach: providing algorithms for restriction and interpolation, and transferring solutions between spatial grids at different time points. In §4, we provide numerical results illustrating the efficacy of the adaptive coarsening strategy. In §5, we summarize our results and briefly describe related current and future work.

2. Background Information.

Consider a system of ordinary differential equations (ODEs) of the form

$$\mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0, \quad t \in [0, T],$$

which can represent the system obtained from a method-of-lines discretization of (1). This system is discretized on a uniform temporal mesh $t_i = i\delta t$, $i = 0, 1 \ldots, N_t$, $\delta t = T/N_t$, with $\mathbf{u}_i \approx \mathbf{u}(t_i)$. A general one-step iteration for this discretization is

$$\mathbf{u}_i = \mathbf{\Phi}_{i,\delta t}(\mathbf{u_{i-1}}) + \mathbf{g}_i, \quad i = 1, 2, \ldots, N_t,$$

where $\mathbf{\Phi}_{i,\delta t}$ is a time stepping function depending on t_i and δt, and $\mathbf{g}_i$ contains solution-independent terms. We write this as the equivalent matrix equation (abusing notation in the nonlinear case)

$$\mathbf{A}\mathbf{u} \equiv \begin{bmatrix} \mathbf{I} & & & \\ -\mathbf{\Phi}_{1,\delta t} & \mathbf{I} & & \\ & \ddots & \ddots & \\ & & -\mathbf{\Phi}_{N_t,\delta t} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N_t} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{N_t} \end{bmatrix} \equiv \mathbf{g}, \tag{2}$$

where $\mathbf{g}_0 = \mathbf{u}_0$. Note that forward substitution corresponds to sequential time stepping.

2.1. MGRIT.

To solve (2) by MGRIT, we require a coarse-grid problem, a relaxation scheme, and restriction and prolongation operators. Set a temporal coarsening factor m and define a coarse time grid $T_{i_c} = i_c\Delta T$, $i_c = 0, 1, \ldots, N_T = N_t/m$, $\Delta T = m\delta t$, as pictured in Figure 1 [7, original]. The T_{i_c} present on both fine and coarse grids are *C-points* and the remaining t_i are *F-points*. On the coarse grid, define a coarse time stepper $\mathbf{\Phi}_{i_c,\Delta T}$. In two-level MGRIT, this coarse-grid problem is solved exactly, whereas multilevel MGRIT applies this process recursively.

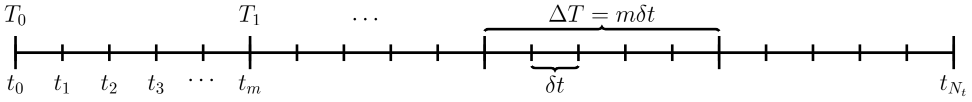

FIG. 1. *Fine and coarse temporal grids.*

Two fundamental types of temporal relaxation are used in MGRIT: F-relaxation and C-relaxation. F-relaxation updates in parallel the F-point values $\mathbf{u}_i$ in interval (T_{i_c}, T_{i_c+1}) by propagating the current C-point value $\mathbf{u}_{mi_c}$ across the interval using each $\mathbf{\Phi}_{i,\delta t}$ in sequence. Since each interval is updated independently, the intervals can be processed in parallel. Similarly, C-relaxation updates C-point values $\mathbf{u}_{mi_c}$ based on current F-point values $\mathbf{u}_{mi_c-1}$, which can also be done in parallel. These relaxation strategies are illustrated in Figure 2 [7, original]. In particular, note that two-level MGRIT with F-relaxation is equivalent to parareal [7, 10]. These sweeps can also be combined into FCF-relaxation: F-relaxation followed by C-relaxation followed by a second F-relaxation. Ideal restriction and prolongation ("ideal" as they correspond to a Schur complement coarse grid) are equivalent to particular combinations of injection and F-relaxation: ideal restriction is injection preceded by an F-relaxation, and ideal prolongation is injection followed by an F-relaxation [7].

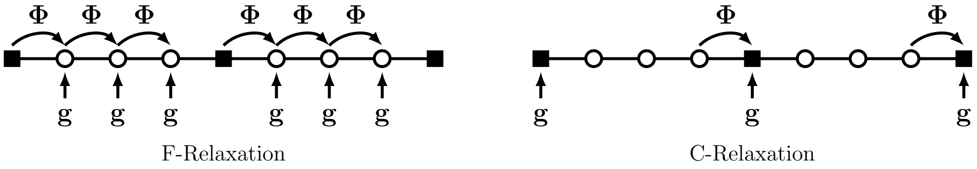

Fig. 2. *Illustration of F- and C-relaxation on a 9-point temporal grid with coarsening factor 4.*

2.2. Discretization. We consider the numerical solution of (1) on a finite spatial interval $[a, b]$ and assume periodic boundary conditions in all that follows. We use the vertex-centered approach to construct spatial grids [13, § III.4]: a grid is defined by points $\{x_j\}_{j=0}^{N-1}$ and has cells $\Omega_j = [x_{j-1/2}, x_{j+1/2}]$, where $x_{j\pm 1/2} = \frac{1}{2}(x_j + x_{j\pm1})$; i.e., the vertices (boundaries/cell interfaces) are *centered* between x_j and $x_{j\pm1}$. When performing spatial coarsening, the vertex-centered approach allows us to use a subset of $\{x_j\}_{j=0}^{N-1}$ to describe the grid on each level: no new reference points are required. Dividing $[a, b]$ into N_x cells of equal width, the fine-grid points $\{x_j\}$ are

$$x_j = a + \tfrac{1}{N_x}(b-a)\left(\tfrac{1}{2} + j\right), j = 0, 1, \dots, N_x - 1,$$

Defining $\delta x_j = \frac{1}{2}(x_{j+1} - x_{j-1})$, (1) is semi-discretized in space as [13]

$$\partial_t u_j + \tfrac{1}{\delta x_j}\left(f^*_{j+1/2}(t) - f^*_{j-1/2}(t)\right) = 0, \tag{3}$$

where $f^*_{j+1/2}(t)$ is chosen as the local Lax-Friedrichs flux approximation:

$$f^*_{j+1/2}(t) = \tfrac{1}{2}\left[a(x_{j+1/2}, t)\left(u_{j+1}(t) + u_j(t)\right) - |a(x_{j+1/2}, t)|\left(u_{j+1}(t) - u_j(t)\right)\right]. \tag{4}$$

This conservative discretization was chosen to make our approach applicable to general nonlinear conservation laws $\partial_t u + \partial_x f(u) = 0$, where (4) guarantees correct shock speeds. In fact, numerical results for the 1D Burgers' equation similar to the ones reported in this paper have already been obtained, but are omitted due to space constraints. For simplicity and to avoid the need to ensure that the CFL condition is satisfied on all grid levels, we focus in this paper on the backward Euler time discretization, resulting in the fully discrete equation (space index j, time index i)

$$\begin{aligned}(5)\qquad &\left(a^{i+1}_{j-1/2} + \left|a^{i+1}_{j-1/2}\right|\right)\frac{\delta t}{2\delta x_j}u^{i+1}_{j-1} - \left(a^{i+1}_{j+1/2} - \left|a^{i+1}_{j+1/2}\right|\right)\frac{\delta t}{2\delta x_j}u^{i+1}_{j+1} \\ &+ \left[1 + \left(a^{i+1}_{j+1/2} - a^{i+1}_{j-1/2} + \left|a^{i+1}_{j+1/2}\right| + \left|a^{i+1}_{j-1/2}\right|\right)\frac{\delta t}{2\delta x_j}\right]u^{i+1}_j = u^i_j.\end{aligned}$$

The MGRIT matrix equation described in (2) typically corresponds to cases where $\mathbf{\Phi}$ is a sparse matrix. If $\mathbf{\Phi}$ is the *inverse* of a sparse matrix, we may instead write $-\mathbf{I}$ on the first block subdiagonal and $\mathbf{\Phi}^{-1}_{i,\delta t}$ on the block main diagonal. In this case, applying $\mathbf{\Phi}_{i,\delta t}$ is a linear solve and $\mathbf{\Phi}^{-1}_{i,\delta t}$ is the matrix defined by (5).

For temporal coarsening, the coarse-grid time stepper $\mathbf{\Phi}_{i_c,\Delta T}$ is obtained by using ΔT in place of δt in (5). For spatial coarsening, we use a Galerkin definition involving $\mathbf{\Phi}_{i_c,\Delta T}$ on the fine spatial grid, which we find results in cheaper overall algorithms compared to using (5) on the coarse spatial grid, both in terms of iterations required and overall time to solution. Working with the sparse $\mathbf{\Phi}^{-1}$ MGRIT matrix and assuming spatial restriction $\mathbf{R}_i$ and prolongation $\mathbf{P}_i$ correspond to time t_i, we write the coarse-grid block equation as

$$-\mathbf{R}_i\mathbf{P}_{i-1}\mathbf{u}_{i-1} + \mathbf{R}_i\Phi^{-1}_{i,\Delta T}\mathbf{P}_i\mathbf{u}_i = \mathbf{g}_i,$$

and so

$$\mathbf{u}_i = \left(\mathbf{R}_i\Phi^{-1}_{i,\Delta T}\mathbf{P}_i\right)^{-1}\left[\mathbf{R}_i\mathbf{P}_{i-1}\mathbf{u}_{i-1} + \mathbf{g}_i\right].$$

$N_x \times N_t$				$2^7 \times 2^7$	$2^8 \times 2^8$	$2^9 \times 2^9$	$2^{10} \times 2^{10}$	$2^{11} \times 2^{11}$
$a(x,t) = 1.0$	No SC	2-level	It	14	14	15	15	15
			Time (TPI)	0.11 (0)	0.39 (.02)	1.52 (.10)	5.88 (.39)	28.98 (1.93)
		F-cycle	It	14	15	17	20	22
			Time (TPI)	0.24 (.01)	0.95 (.06)	4.20 (.24)	21.00 (1.05)	90.07 (4.09)
	SC-2	2-level	It	15	15	15	16	16
			Time (TPI)	0.11 (0)	0.40 (.02)	1.49 (.09)	5.81 (.36)	27.84 (1.74)
		F-cycle	It	15	17	20	24	28
			Time (TPI)	**0.18** (.01)	**0.69** (.04)	**2.79** (.13)	**12.31** (.51)	**64.41** (2.30)
$a(x,t) = 0.1$	No SC	2-level	It	8	8	8	8	8
			Time (TPI)	0.09 (.01)	0.32 (.04)	1.17 (.14)	4.40 (.55)	20.06 (2.50)
		F-cycle	It	8	8	9	9	10
			Time (TPI)	**0.16** (.02)	**0.61** (.07)	**2.48** (.27)	**10.90** (1.21)	**45.61** (4.56)
	SC-2	2-level	It	64	90	92	92	92
			Time (TPI)	0.25 (0)	1.15 (.01)	4.50 (.04)	16.91 (.18)	68.14 (.74)
		F-cycle	It	64	92	94	95	95
			Time (TPI)	0.52 (0)	2.36 (.02)	9.44 (.10)	35.74 (.37)	135.25 (1.42)

TABLE 1

No SC: no spatial coarsening; SC-2: factor-two spatial coarsening. For each problem, the fastest F-cycle results are shown in bold.

To illustrate the need for adaptive coarsening ,we solve (1) for $(x,t) \in [-2,2] \times [0,4]$, using either $a(x,t) = 1.0$ or $a(x,t) = 0.1$, and $u_0(x) = \sin(0.5\pi x)$, in which case (5) reduces to simple upwinding. We use factor-two temporal and spatial coarsening and a halting tolerance for the residual vector 2-norm of 10^{-10} scaled by the domain size: $\text{tol} = (2.5 \times 10^{-11})\sqrt{N_t N_x}$. Results for two -level MGRIT with FCF-relaxation and either no spatial coarsening (No SC) or both temporal coarsening and factor-two spatial coarsening (SC-2) (full weighting restriction, linear interpolation, and a Galerkin coarse-grid operator) are presented in Table 1, which records iteration count, time to solution, and time per iteration (TPI). When $a(x,t) = 1.0$, the results are quite similar in terms of iteration count and there can be substantial savings of approximately 30% in terms of time to solution for spatial coarsening, indicating why it is desirable for implicit time stepping, in addition to it being necessary in the case of explicit time stepping. For $a(x,t) = 0.1$, however, including spatial coarsening increases both iteration count and time to solution many times over due to the induced anisotropy, making it a non-starter in such cases.

3. Adaptive Spatial Coarsening Strategy.

The main contribution of this paper is a set of algorithms used to implement adaptive spatial coarsening such that local wave speeds near zero do not cause extremely slow MGRIT convergence. These algorithms are intended as proof of concept for implicit time stepping routines: modifications that are not described here are required to handle other equations and explicit time stepping. As a note on implementing these algorithms: in the case of variable coefficient linear advection, the grid hierarchies determined will not change from one MGRIT iteration to the next, so the grids and associated transfer operators need only be computed once and then stored for reuse on later iterations.

3.1. Restriction.

The 1D factor-two restriction strategy for a periodic domain is illustrated for four levels and sixteen cells in Figure 3. The numerical labels on each level serve as global cell indices, recording which fine-grid reference points are used in defining the coarser levels. Rather than aggregating pairs of adjacent cells when moving from level ℓ to $\ell+1$, we instead remove every second cell, with remaining cells expanding to cover the removed cells' portion of the domain.

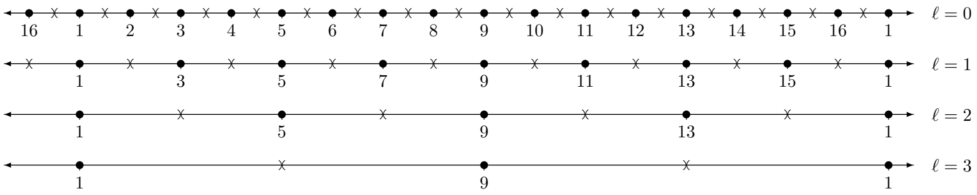

Fig. 3. *Factor-two coarsening in 1D with periodic BCs. The)(symbols represent cell boundaries.*

Considering the discretization (5) and the results of the previous section, we see that $a(x,t)$ near zero can result in weak couplings in the spatial direction, meaning high frequency errors are not reduced effectively by relaxation, and thus the error cannot be represented properly on coarse spatial grids, drastically reducing the efficiency of a multigrid iteration. Thus, if $a(x,t)$ within cell Ω_j is relatively small, we wish to retain Ω_j for the next level, as coarsening in this region will not benefit the solution process. Experiments (not included here) suggest that it is unnecessary to fix the width of Ω_j; it is sufficient to ensure Ω_j is not removed. To determine if Ω_j is to be kept, we propose the following condition:

$$\text{If } \min_{x\in\Omega_j} |a(x,t)| \frac{\delta t}{\delta x_j} < \text{tol}_*\text{: keep } \Omega_j\text{; else: coarsen normally.} \tag{6}$$

Since $a(x,t)\delta t/\delta x_j$ appears in the coefficients of (5), this is the appropriate measure to identify small matrix elements that indicate weak coupling and may lead to degraded multigrid performance if spatial coarsening is used. To minimize $|a(x,t)|$ in the discrete setting, we take the smaller of the two values $|a(x_{j\pm 1/2}, t)|$.

The result of this coarsening process is shown in Figure 4 for the same fine grid as in Figure 3 at a fixed time point, now with the arbitrary assumption that (6) is satisfied on all levels in cells 4 and 10. The labeled global indices are used to compute

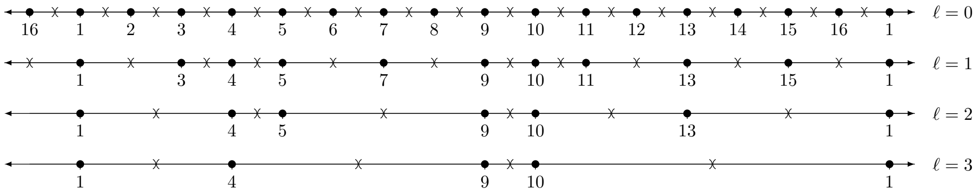

Fig. 4. *Adaptive coarsening in 1D with periodic BCs.*

cell boundaries as per the definition of vertex-centered grids. In Figure 5, we show resulting adaptive grid hierarchies for $[-2,2] \times [0,4]$ discretized as a 64×64 space-time grid, with $a(x,t) = -\sin^2(\pi(x-t))$ (top) or $a(x,t) = \frac{1}{2}(1-\sin(2\pi t))\sin(\pi x)$ (bottom). Black vertices indicate reference points for cells only present on level 0, red dots indicate reference points for cells present on levels 0 through 1, blue dots indicate cells present on levels 0 through 2, and green dots indicate cells present on levels 0 through 3.

The value u_j^i approximates the cell average of $u(x,t_i)$ over Ω_j. We want $u(x,t)$ to be conserved by restriction, so we compute coarse-grid cell averages that ensure the total area below the curve remains constant. If cell Ω_j is retained from level ℓ to $\ell+1$, it may stay the same, or one or both of its boundaries may expand outwards, increasing its size. If Ω_j remains the same size, u_j^i is unchanged from level ℓ to $\ell+1$. If Ω_j expands, we recompute u_j^i as a volume-weighted average based on information from the neighboring cells that were removed. Suppose that $\Theta(\cdot)$ maps the local indices of cells being retained on level ℓ to the corresponding local indices on level $\ell+1$. When cell j is kept, then Ω_j^ℓ is the fine-grid cell associated with the coarse-grid cell $\Omega_{\Theta(j)}^{\ell+1}$. To compute $u_{\Theta(j)}^{\ell+1}$ (omitting the time superscript i for clarity), we require u_{j-1}^ℓ, u_j^ℓ, u_{j+1}^ℓ, $x_{j\pm1}^\ell$, and $x_{\Theta(j)\pm1}^{\ell+1}$. The restriction formula for $u_{\Theta(j)}^{\ell+1}$ is

$$\begin{aligned}
u_{\Theta(j)}^{\ell+1} &= \frac{(x_{j+1/2}^\ell - x_{j-1/2}^\ell)u_j^\ell + (x_{\Theta(j)+1/2}^{\ell+1} - x_{j+1/2}^\ell)u_{j+1}^\ell + (x_{j-1/2}^\ell - x_{\Theta(j)-1/2}^{\ell+1})u_{j-1}^\ell}{(x_{j+1/2}^\ell - x_{j-1/2}^\ell) + (x_{\Theta(j)+1/2}^{\ell+1} - x_{j+1/2}^\ell) + (x_{j-1/2}^\ell - x_{\Theta(j)-1/2}^{\ell+1})} \\
&= \frac{\frac{1}{2}(x_{j+1}^\ell - x_{j-1}^\ell)u_j^\ell + \frac{1}{2}(x_{\Theta(j)+1}^{\ell+1} - x_{j+1}^\ell)u_{j+1}^\ell + \frac{1}{2}(x_{j-1}^\ell - x_{\Theta(j)-1}^{\ell+1})u_{j-1}^\ell}{\frac{1}{2}(x_{\Theta(j)+1}^{\ell+1} - x_{\Theta(j)-1}^{\ell+1})} \\
&= \frac{(x_{j+1}^\ell - x_{j-1}^\ell)u_j^\ell + (x_{\Theta(j)+1}^{\ell+1} - x_{j+1}^\ell)u_{j+1}^\ell + (x_{j-1}^\ell - x_{\Theta(j)-1}^{\ell+1})u_{j-1}^\ell}{x_{\Theta(j)+1}^{\ell+1} - x_{\Theta(j)-1}^{\ell+1}}. \qquad (7)
\end{aligned}$$

For factor-two coarsening, this reduces to the full weighting formula, and if no spatial coarsening is carried out (i.e., $\Theta(\cdot)$ is the identity), this reduces to $u_j^{\ell+1} = u_j^\ell$.

To implement this in XBraid, we create a grid_info structure that contains

1. int *fidx: array of global cell indices.
2. double *xref: array of cell reference points x_j.
3. int *indicator: array of indicators used by the restriction algorithm.

The values in fidx are global cell indices: for example, level 2 in Figure 4 contains 6 cells, which have local indices $\{0,\ldots,5\}$ and global indices $\{1,4,5,9,10,13\}$. The values in xref are required by (7). The values in indicator record which cells are to be kept for the next level, and can store information from prior levels. An array of grid_info structures serves as a grid hierarchy for a given time point t_i.

The restriction algorithm requires two loops over the elements of the fine input

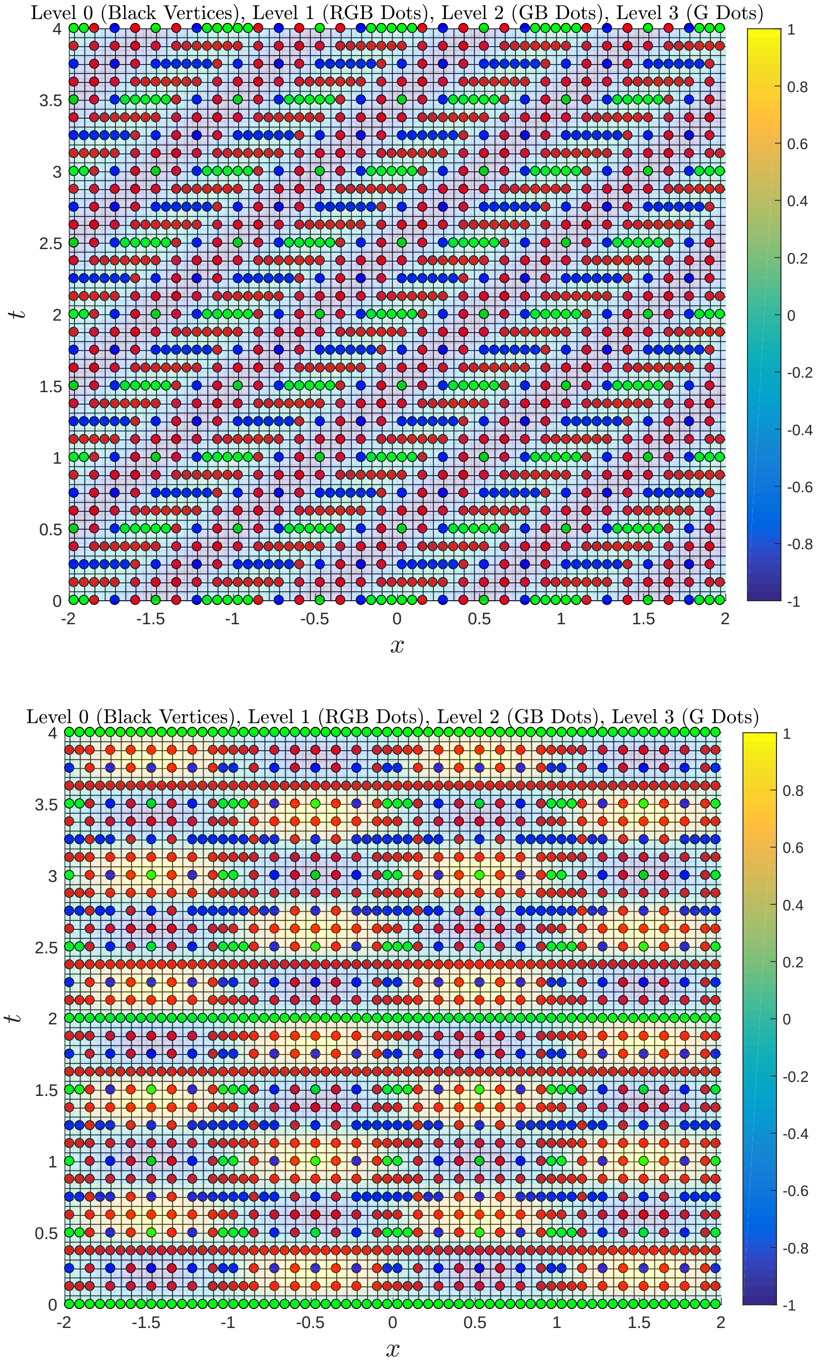

Fig. 5. *Space-time meshes obtained from adaptive spatial coarsening over 4 levels, starting with $N_x = N_t = 64$. Top: $a(x,t) = -\sin^2(\pi(x-t))$, bottom: $a(x,t) = -\sin(2.5\pi t)\sin(\pi x)$. The background color indicates the size of $a(x,t)$. Spatial coarsening is inhibited in regions where $|a|$ is small.*

vector. The first is used to determine the total number of coarse cells and their global indices. When restricting from level ℓ to $\ell+1$, we keep Ω_j^ℓ if any of the following conditions are satisfied:

(a) `indicator`$[j]$ shows cell with `fidx`$[j]$ satisfied condition (c) on prior level.
(b) `fidx`$[j] \in \{0, 2^{\ell+1}, 2\cdot 2^{\ell+1}, 3\cdot 2^{\ell+1}, 4\cdot 2^{\ell+1}, \ldots\} \Leftrightarrow$ `fidx`$[j] \mod 2^{\ell+1} = 0$.
(c) Ω_j^ℓ satisfies condition (6).

Condition (a) ensures that if the cell with global index `fidx`$[j]$ is kept on level ℓ, it will be kept for level $\ell+1$, which allows us to use (7) to compute coarse-grid cell averages. If all cells are flagged to be kept (i.e., further coarsening is impossible), then we return the input vector as the coarsened vector.

The second loop copies information to the `grid_info` structure for the next level and computes new cell averages. To compute $u_{\Theta(j)}^{\ell+1}$, we must identify the closest coarse cell on either side of Ω_j^ℓ. Given the stated assumptions, coarse cells are separated by at most one fine cell (factor-two coarsening), so either $\Theta(j)-1 = \Theta(j-1)$ or $\Theta(j)-1 = \Theta(j-2)$; likewise, either $\Theta(j)+1 = \Theta(j+1)$ or $\Theta(j)+1 = \Theta(j+2)$.

3.2. Prolongation. Because the grid hierarchy was recorded during the restriction process, prolongation is simply a matter of computing fine-grid cell averages. If Ω_j was part of the coarse grid, the cell average remains unchanged. If Ω_j was not part of the coarse grid, its average is computed as a conservative weighted combination of the cell averages of its nearest neighbors. To simplify expressions, we change notation, now using subscripts to indicate *global* instead of *local* indices. Suppose cell $\Omega_j^{\ell-1}$ is introduced between preexisting cells $\Omega_{j-\alpha}^{\ell-1}$ and $\Omega_{j+\beta}^{\ell-1}$, where $\alpha, \beta \in \mathbb{N}$. Then

$$\begin{aligned}
u_j^{\ell-1} &= \frac{\left(x_{j+\beta/2}^{\ell-1} - x_{j+(\beta-\alpha)/2}^{\ell}\right) u_{j+\beta}^{\ell} + \left(x_{j+(\beta-\alpha)/2}^{\ell} - x_{j-\alpha/2}^{\ell-1}\right) u_{j-\alpha}^{\ell}}{x_{j+\beta/2}^{\ell-1} - x_{j-\alpha/2}^{\ell-1}} \\
&= \frac{\left(x_j^{\ell-1} + x_{j+\beta}^{\ell-1} - x_{j-\alpha}^{\ell} - x_{j+\beta}^{\ell}\right) u_{j+\beta}^{\ell} + \left(x_{j-\alpha}^{\ell} + x_{j+\beta}^{\ell} - x_{j-\alpha}^{\ell-1} - x_j^{\ell-1}\right) u_{j-\alpha}^{\ell}}{x_j^{\ell-1} + x_{j+\beta}^{\ell-1} - x_{j-\alpha}^{\ell-1} - x_j^{\ell-1}} \\
&= \frac{\left(x_j^{\ell-1} - x_{j-\alpha}^{\ell-1}\right) u_{j+\beta}^{\ell} + \left(x_{j+\beta}^{\ell-1} - x_j^{\ell-1}\right) u_{j-\alpha}^{\ell}}{x_{j+\beta}^{\ell-1} - x_{j-\alpha}^{\ell-1}}. \qquad (8)
\end{aligned}$$

The final simplification follows from the fact that reference points do not change position between levels: $x_{j+\beta}^{\ell-1} = x_{j+\beta}^{\ell}$ and $x_{j-\alpha}^{\ell-1} = x_{j-\alpha}^{\ell}$. This is simply a convex combination of $u_{j-\alpha}^{\ell}$ and $u_{j+\beta}^{\ell}$ with weights depending on the location of $x_j^{\ell-1} \in [x_{j-\alpha}^{\ell-1}, x_{j+\beta}^{\ell-1}]$, and is standard linear interpolation when factor-two coarsening is used.

3.3. Movement Between Grids. In addition to the restriction and prolongation of solutions between levels, we also need to transfer solution approximations between time points on a fixed level. For adaptive grid refinement, the hierarchy of grids created may vary with time. This means that a representation of $\mathbf{u}_i$ must be computed on the spatial grid at time t_{i+1} before $\mathbf{u}_{i+1}$ can be computed in the time marching.

To map $\mathbf{u}$ from grid A to grid B, we use the `grid_info` structures corresponding to these grids. For each cell Ω_j^B on grid B, identify the cells on grid A that contain its left boundary (Ω_α^A) and right boundary (Ω_ω^A). Computing a weighted average of the cell values from α to ω, scaled by the width of Ω_j^B, gives the cell average u_j^B on Ω_j^B. For periodic boundary conditions, the first cell on both source and target grids may appear as a pair of disconnected intervals: one at the start and one at the end of the domain. To simplify this case, we treat the disconnected portions as separate cells before merging their results.

4. Numerical Results.

Numerical results were generated using the XBraid parallel-in-time software package [1], and the CHOLMOD [3] and UMFPACK [4] packages from SuiteSparse for sparse matrix storage, multiplication, and factorization.

We again solve (1) with $u_0(x) = \sin(0.5\pi x)$ on $[-2, 2] \times [0, 4]$, starting with a $N_x \times N_t$ fine space-time grid, now with three different variable wave speeds:

1. $a(x) = -(0.1 + \cos^2(0.25\pi(x + 2)))$ (a varies in space only),
2. $a(x, t) = -\sin^2(\pi(x - t))$, and
3. $a(x, t) = -\sin(2.5\pi t) \sin(\pi x)$.

The two $a(x, t)$ examples were previously used to produce the sample grids in Figure 5. We use MGRIT with factor-two temporal coarsening and a halting tolerance of $\text{tol} = (2.5 \times 10^{-11})\sqrt{N_t N_x}$.

$N_x \times N_t$				$2^7 \times 2^7$	$2^8 \times 2^8$	$2^9 \times 2^9$	$2^{10} \times 2^{10}$	$2^{11} \times 2^{11}$
$a(x) = -(0.1 + \cos^2(0.25\pi(x + 2)))$	No SC	2-level	It	12	14	14	14	15
			Time (TPI)	0.06 (0)	0.19 (.01)	0.72 (.05)	2.92 (.20)	12.22 (.81)
		F-cycle	It	12	15	16	18	20
			Time (TPI)	**0.12** (.01)	**0.59** (.03)	**2.54** (.15)	14.77 (.82)	59.01 (2.95)
	SC-2	2-level	It	64	78	83	84	85
			Time (TPI)	0.18 (0)	0.73 (0)	2.91 (.03)	12.25 (.14)	48.49 (.57)
		F-cycle	It	64	79	85	86	87
			Time (TPI)	0.39 (0)	1.67 (.02)	6.09 (.07)	25.42 (.29)	98.49 (1.13)
	SC-A	2-level	It	25	27	28	29	29
			Time (TPI)	0.09 (0)	0.30 (.01)	1.17 (.04)	4.83 (.16)	19.68 (.67)
		F-cycle	It	26	27	28	29	30
			Time (TPI)	0.21 (0)	0.72 (.02)	2.76 (.09)	**11.09** (.38)	**45.37** (1.51)
$a(x, t) = -\sin^2(\pi(x - t))$	No SC	2-level	It	12	12	13	13	13
			Time (TPI)	0.08 (0)	0.29 (.02)	1.15 (.08)	4.28 (.32)	18.48 (1.42)
		F-cycle	It	12	13	15	16	18
			Time (TPI)	**0.17** (.01)	**0.69** (.05)	**3.04** (.20)	13.46 (.84)	61.51 (3.41)
	SC-A	2-level	It	17	16	17	19	22
			Time (TPI)	0.10 (0)	0.35 (.02)	1.36 (.08)	5.59 (.29)	23.74 (1.07)
		F-cycle	It	18	17	19	21	25
			Time (TPI)	0.21 (.01)	0.74 (.04)	3.11 (.16)	**12.74** (.60)	**57.65** (2.30)
$a(x, t) = -\sin(2.5\pi t)\sin(\pi x)$	No SC	2-level	It	13	12	12	12	13
			Time (TPI)	0.09 (0)	0.27 (.02)	1.06 (.08)	4.12 (.34)	17.88 (1.37)
		F-cycle	It	13	14	14	16	17
			Time (TPI)	**0.17** (.01)	**0.70** (.05)	**2.74** (.19)	**12.27** (.76)	**54.07** (3.18)
	SC-A	2-level	It	19	19	20	24	26
			Time (TPI)	0.10 (0)	0.37 (.01)	1.48 (.07)	6.26 (.26)	25.20 (.96)
		F-cycle	It	19	20	20	24	26
			Time (TPI)	0.24 (.01)	0.86 (.04)	3.23 (.16)	14.11 (.58)	58.43 (2.24)

TABLE 2

No SC: no spatial coarsening; SC-2: factor-two coarsening, SC-A: adaptive coarsening. For each test problem, the fastest F-cycle results are shown in bold.

For case 1, we see that small $a(x)$ in part of the domain causes significant deterioration for SC-2, and that the adaptive coarsening scheme SC-A recovers good convergence, offering a moderate improvement in total time to solution on the No SC F-cycle results in spite of the increased iterations required. Note that, when comparing the entries of Table 2, we are not concerned with the increased serial time to solution for F-cycles over 2-level cycles, because F-cycles parallelize better. We are instead looking for algorithmic scalability of the F-cycles in terms of iteration count.

For cases 2 and 3, the additional complexity of having grid hierarchies that vary in time results in a more costly set-up phase and a greater per iteration cost when compared to spatial variation only. As in case 1, SC-2 leads to convergence degradation (not shown). While we do not yet see a benefit to using SC-A over No SC for implicit time stepping for all test problems, these results are promising for the explicit case on very large parallel machines, where spatial coarsening is required for MGRIT, and gains can be expected over sequential time stepping due to the vastly increased parallelism in MGRIT. If we can, in future work, further reduce or eliminate the growth in iteration count for spatial coarsening without increasing the time per iteration, spatial coarsening should result in significant savings in the implicit case.

5. Conclusions.

In this paper, we discussed an MGRIT adaptive spatial coarsening strategy for the conservative linear advection equation and implicit time stepping. We observed that this adaptive coarsening strategy solves one of the two main problems involved in implementing spatial coarsening for hyperbolic problems: weak spatial couplings due to small wave speeds are no longer an issue. However, there remains the problem of the increase in iterations required for MGRIT to converge when spatial coarsening is introduced, which is the subject of active research.

As mentioned previously, the algorithm described is only intended to serve as proof of concept, with several improvements and extensions currently being developed. Ongoing research explores mode analysis to understand convergence deterioration and aims to improve MGRIT iteration counts by considering exact reduction schemes that may inspire improved restriction, interpolation, and coarse-grid operators. Future plans for solving hyperbolic problems with MGRIT involve implementing adaptive spatial coarsening for the 1D Burgers' equation, and then extending these ideas to two or more spatial dimensions and/or systems of hyperbolic equations.

REFERENCES

[1] *XBraid: Parallel multigrid in time.* http://llnl.gov/casc/xbraid.

[2] P. Chartier and B. Philippe, *A parallel shooting technique for solving dissipative ODE's*, Computing, 51 (1993), pp. 209–236.

[3] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, *Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate*, ACM Trans. Math. Softw., 35 (2008), pp. 22:1–22:14.

[4] T. A. Davis, *Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Softw., 30 (2004), pp. 196–199.

[5] M. Emmett and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Communications in Applied Mathematics and Computational Science, 7 (2012), pp. 105–132.

[6] R. Falgout, A. Katz, T. V. Kolev, J. Schroder, A. Wissink, and U. Yang, *Parallel time integration with multigrid reduction for a compressible fluid dynamics application*, Journal of Computational Physics, (2014).

[7] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp. C635–C661.

[8] R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, *Multigrid Reduction in Time for Nonlinear Parabolic Problems*, Jan 2016, https://doi.org/10.2172/1236132.

[9] M. J. Gander, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 69–113.

[10] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29 (2007), pp. 556–578.

[11] S. Güttel, *A parallel overlapping time-domain decomposition method for ODEs*, in Domain decomposition methods in science and engineering XX, Springer, 2013, pp. 459–466.

[12] G. Horton, *The time-parallel multigrid method*, Communications in applied numerical methods, 8 (1992), pp. 585–595.

[13] W. Hundsdorfer and J. G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33, Springer Science & Business Media, 2013.

[14] M. Lecouvez, R. D. Falgout, C. S. Woodward, and P. Top, *A parallel multigrid re-*

duction in time method for power systems, in Power and Energy Society General Meeting (PESGM), 2016, IEEE, 2016, pp. 1–5.

[15] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps «pararéel»*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 332 (2001), pp. 661–668.

[16] S. Vandewalle and E. Van de Velde, *Space-time concurrent multigrid waveform relaxation*, Annals of Numer. Math, 1 (1994), pp. 347–363.