PARALLEL-IN-TIME MULTIGRID WITH ADAPTIVE SPATIAL COARSENING FOR THE LINEAR ADVECTION AND INVISCID BURGERS EQUATIONS*

HANS DE STERCK†, ROBERT D. FALGOUT‡, ALEXANDER J.M. HOWSE§, SCOTT P. MACLACHLAN¶, AND JACOB B. SCHRODER‡

Abstract. We apply a multigrid reduction-in-time (MGRIT) algorithm to hyperbolic partial differential equations in one spatial dimension. This study is motivated by the observation that sequential time-stepping is a computational bottleneck when attempting to implement highly concurrent algorithms, thus parallel-in-time methods are desirable. MGRIT adds parallelism by using a hierarchy of successively coarser temporal levels to accelerate the solution on the finest level. In the case of explicit time-stepping, spatial coarsening is a suitable approach 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, uniform 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 the variable coefficient linear advection equation and the inviscid Burgers equation using first-order explicit or implicit time-stepping methods. Serial numerical results show this method offers significant improvements over uniform coarsening and is convergent for the inviscid Burgers equation with and without shocks. Parallel scaling tests on up to 128K cores indicate that run-time improvements over serial time-stepping strategies are possible when spatial parallelism alone saturates, and that scalability is robust for oscillatory solutions which change on the scale of the grid spacing.

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

AMS subject classifications. 65F10, 65M22, 65M55, 35L03, 35L60

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 time integration schemes. While temporal parallelism may seem counter-intuitive, the development of parallel-in-time methods is an active area of research, with a history spanning several decades [15]. Variants include direct methods and iterative methods based on deferred corrections [10], domain decomposition [17], multigrid [18], multiple shooting [5], and waveform relaxation [27] approaches. These methods have had significant success in providing further speedup in the solution of parabolic equations, or equations with significant diffusivity, but have had markedly less success with hyperbolic or advection dominated problems [24].

For example, one of the most influential parallel-in-time methods is parareal [22],

*Submitted to the editors August 25, 2017.

Funding: This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-737050). The work of SPM was partially supported by an NSERC Discovery Grant. The work of HDS and AJMH was partially supported by an ARC discovery project.

†School of Mathematical Sciences, Monash University, Melbourne, VIC hans.desterck@monash.edu.

‡Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA (rfalgout@llnl.gov, schroder2@llnl.gov).

§Department of Applied Mathematics, University of Waterloo, Waterloo, ON (ahowse@uwaterloo.ca).

¶Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL (smaclachlan@mun.ca).

an iterative predictor-corrector method (that is equivalent to a two-level multigrid scheme [16]) which combines the use of a coarse time integrator in serial and a fine time integrator in parallel. Parareal has been shown to have stability issues for the constant coefficient linear advection equation [16]. A number of variants and modifications have been proposed, and analysis has identified that issues arise when solutions lack regularity [8] due to phase errors in the coarse propagator [24]. A number of variants have been proposed to stabilize and improve the convergence of parareal for such problems [6, 8, 14, 25], but with increased memory requirements or other restrictions. As a result, parallel-in-time methods which can be effectively applied to hyperbolic or advection dominated problems are still highly sought after.

In this paper, we discuss the multigrid reduction-in-time (MGRIT) method [11] and use XBraid [2], 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 [11] and nonlinear [13] parabolic partial differential equations (PDEs) in multiple dimensions, the Navier-Stokes equations [12], and power system models [20]. We now consider applying MGRIT to hyperbolic PDEs.

As a multigrid method, MGRIT primarily involves temporal coarsening, but spatial coarsening is a suitable approach 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 – resulting from small local wave speeds – 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 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 that resolves this problem for the conservative hyperbolic PDE

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

by locally preventing coarsening in regions with near zero Courant numbers. In particular, we consider the variable coefficient linear advection equation, $f(u,x,t) = a(x,t)u$, and the inviscid Burgers equation, $f(u,x,t) = \frac{1}{2}u^2$.

The remainder of this paper is as follows. In §2, we describe the MGRIT algorithm and the discretization of (1). In §3, we present our adaptive coarsening approach, providing algorithms for grid coarsening and transferring solutions between different spatial grids. In §4, we provide serial numerical results illustrating the efficacy of the adaptive coarsening strategy. In §5, we provide parallel scaling results comparing MGRIT with adaptive coarsening and different combinations of space-time parallelism to sequential time-stepping with spatial parallelism, illustrating the robustness of the approach for large problem sizes and its potential to achieve run-time speedups when spatial parallelism alone saturates. In §6, we summarize our results and briefly describe related current and future work.

2. MGRIT Formulation and Discretization. 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 a 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 computing the discrete solution 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 $\mathbf{g}_0 = \mathbf{u}_0$. Here 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. We 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 [11, original]. The T_{i_c} present on both fine and coarse grids are *C-points* and the remaining t_i are *F-points*. We define a coarse time stepper $\mathbf{\Phi}_{i_c,\Delta T}$ by rediscretizing on the coarse-in-time grid. In two-level MGRIT, this coarse-grid problem is solved exactly, whereas multilevel MGRIT applies this process recursively.

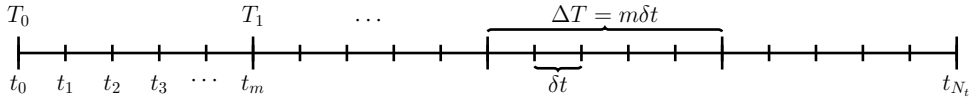

FIG. 1. *Fine and coarse temporal grids.*

Two fundamental types of temporal relaxation are used in MGRIT: F-relaxation and C-relaxation. F-relaxation updates the F-point values $\mathbf{u}_i$ in the interval (T_{i_c}, T_{i_c+1}) by starting with the C-point value $\mathbf{u}_{mi_c}$ and then applying 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 [11, original]. In particular, note that two-level MGRIT with F-relaxation is equivalent to parareal [11, 16]. 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 generate the Schur complement as the Petrov-Galerkin coarse-grid operator) 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 [11].

MGRIT uses the Full Approximation Storage (FAS) framework [3] for solving both linear and nonlinear problems, which involves computing the coarse-grid correction by solving a coarsened version of the residual equation $\mathcal{A}(\mathbf{u} + \mathbf{e}) - \mathcal{A}(\mathbf{u}) = \mathbf{r}$, where $\mathcal{A}$ is the (potentially nonlinear) operator to be inverted. The two-grid MGRIT FAS algorithm first appeared in [12], though we instead reproduce here the variant from [13] which accounts for the possibility of spatial coarsening, see Algorithm 1. We

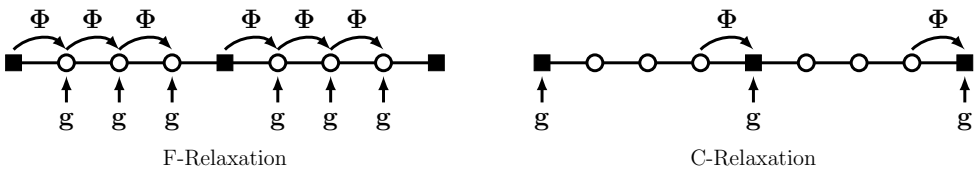

FIG. 2. *Illustration of F- and C-relaxation on a 9-point temporal grid with coarsening factor 4.*

denote injection-based temporal restriction by $\mathbf{R}_{\mathrm{I}}$, ideal temporal prolongation by $\mathbf{P}$, spatial restriction by $\mathbf{R}_{\mathrm{s}}$, and spatial prolongation by $\mathbf{P}_{\mathrm{s}}$. The multigrid variant is obtained by replacing line 5 with a recursive call. In the case of $\mathcal{A}$ being a matrix $\mathbf{A}$ this reduces to the standard multigrid algorithm.

Algorithm 1 FAS-MGRIT

1: **procedure** FAS-MGRIT($\mathcal{A}, \mathbf{u}, \mathbf{g}$)
2: Apply F- or FCF-relaxation to $\mathcal{A}(\mathbf{u}) = \mathbf{g}$
3: Inject the fine-grid approximation and residual to the coarse grid
$\mathbf{u}_\Delta = \mathbf{R}_{\mathrm{I}}(\mathbf{u}), \quad \mathbf{r}_\Delta = \mathbf{R}_{\mathrm{I}}(\mathbf{g} - \mathcal{A}(\mathbf{u}))$
4: If using spatial coarsening then:
$\mathbf{u}_\Delta = \mathbf{R}_{\mathrm{s}}(\mathbf{u}_\Delta), \quad \mathbf{r}_\Delta = \mathbf{R}_{\mathrm{s}}(\mathbf{r}_\Delta)$
5: Solve $\mathcal{A}_\Delta(\mathbf{v}_\Delta) = \mathcal{A}_\Delta(\mathbf{u}_\Delta) + \mathbf{r}_\Delta$
6: Compute the coarse-grid error approximation: $\mathbf{e}_\Delta = \mathbf{v}_\Delta - \mathbf{u}_\Delta$
7: If using spatial coarsening then: $\mathbf{e}_\Delta = \mathbf{P}_{\mathrm{s}}(\mathbf{e}_\Delta)$
8: Correct using ideal interpolation: $\mathbf{u} = \mathbf{u} + \mathbf{P}(\mathbf{e}_\Delta)$
9: **end procedure**

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 [19, § 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 $x_{j\pm1/2} = \frac{1}{2}(x_j + x_{j\pm1})$; 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

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

Defining $\delta x_j = \frac{1}{2}(x_{j+1} - x_{j-1})$, (1) is semi-discretized in space as [19]

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

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

$$f^*_{j+1/2}(t) = \frac{f(u_{j+1}(t), x_{j+1/2}, t) + f(u_j(t), x_{j+1/2}, t)}{2} - \frac{1}{2}\frac{|\partial_u f(u_{j+1}(t), x_{j+1/2}, t)| + |\partial_u f(u_j(t), x_{j+1/2}, t)|}{2}(u_{j+1}(t) - u_j(t)). \tag{4}$$

For variable coefficient linear advection, this reduces to

$$f^*_{j+1/2}(t) = \tfrac{1}{2}\left[a(x_{j+1/2}, t)\,(u_{j+1}(t) + u_j(t)) - |a(x_{j+1/2}, t)|\,(u_{j+1}(t) - u_j(t))\right], \tag{5}$$

and for Burgers' equation

$$f^*_{j+1/2}(t) = \tfrac{1}{4}\left[(u_{j+1}(t))^2 + (u_j(t))^2 - (|u_{j+1}(t)| + |u_j(t)|)\,(u_{j+1}(t) - u_j(t))\right]. \tag{6}$$

This conservative discretization was chosen to make our approach applicable to nonlinear conservation laws $\partial_t u + \partial_x f(u) = 0$, where (4) guarantees correct shock speeds. In this paper we consider the forward and backward Euler time discretizations, which result in the fully discrete equations (space index j, time index i)

$$\begin{aligned}\left(a^i_{j-1/2} + \left|a^i_{j-1/2}\right|\right)\frac{\delta t}{2\delta x_j}u^i_{j-1} - \left(a^i_{j+1/2} - \left|a^i_{j+1/2}\right|\right)\frac{\delta t}{2\delta x_j}u^i_{j+1}\\ + \left[1 - \left(a^i_{j+1/2} - a^i_{j-1/2} + \left|a^i_{j+1/2}\right| + \left|a^i_{j-1/2}\right|\right)\frac{\delta t}{2\delta x_j}\right]u^i_j = u^{i+1}_j\end{aligned} \tag{7}$$

and

$$\begin{aligned}-\left(a^{i+1}_{j-1/2} + \left|a^{i+1}_{j-1/2}\right|\right)\frac{\delta t}{2\delta x_j}u^{i+1}_{j-1} + \left(a^{i+1}_{j+1/2} - \left|a^{i+1}_{j+1/2}\right|\right)\frac{\delta t}{2\delta x_j}u^{i+1}_{j+1}\\ + \left[1 + \left(a^{i+1}_{j+1/2} - a^{i+1}_{j-1/2} + \left|a^{i+1}_{j+1/2}\right| + \left|a^{i+1}_{j-1/2}\right|\right)\frac{\delta t}{2\delta x_j}\right]u^{i+1}_j = u^i_j\end{aligned} \tag{8}$$

for linear advection, and the fully discrete equations

$$\begin{aligned}\left(u^i_{j-1} + \left|u^i_j\right| + \left|u^i_{j-1}\right|\right)\frac{\delta t}{4\delta x_j}u^i_{j-1} - \left(u^i_{j+1} - \left|u^i_{j+1}\right| - \left|u^i_j\right|\right)\frac{\delta t}{4\delta x_j}u^i_{j+1}\\ + \left[1 - \left(\left|u^i_{j+1}\right| + 2\left|u^i_j\right| + \left|u^i_{j-1}\right|\right)\frac{\delta t}{4\delta x_j}\right]u^i_j = u^{i+1}_j\end{aligned} \tag{9}$$

and

$$\begin{aligned}-\left(u^{i+1}_{j-1} + \left|u^{i+1}_j\right| + \left|u^{i+1}_{j-1}\right|\right)\frac{\delta t}{4\delta x_j}u^{i+1}_{j-1} + \left(u^{i+1}_{j+1} - \left|u^{i+1}_{j+1}\right| - \left|u^{i+1}_j\right|\right)\frac{\delta t}{4\delta x_j}u^{i+1}_{j+1}\\ + \left[1 + \left(\left|u^{i+1}_{j+1}\right| + 2\left|u^{i+1}_j\right| + \left|u^{i+1}_{j-1}\right|\right)\frac{\delta t}{4\delta x_j}\right]u^{i+1}_j = u^i_j\end{aligned} \tag{10}$$

for Burgers' equation.

2.3. Coarse-Grid Time Steppers. For temporal coarsening, the coarse-grid time stepper $\mathbf{\Phi}_{i_c,\Delta T}$ is obtained by using ΔT in place of δt in (7–10). For spatial coarsening we handle the explicit and implicit cases in different ways. For explicit time-stepping we simply use (7/9) on the coarse spatial grid, but for implicit time-stepping we use a Galerkin definition involving $\mathbf{\Phi}_{i_c,\Delta T}$. Galerkin-type discretizations lead to optimal results in the A-norm for SPD problems [4], and they have also been used for nonsymmetric matrices, for example, in [26]. We use a Galerkin approach in this paper for implicit timestepping, because we find it leads to better results than rediscretization. To describe this method, we first note that the MGRIT matrix equation described in (2) typically corresponds to cases where $\mathbf{\Phi}$ is a sparse matrix, such as that defined by (7/9). 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 (8/10).

Working with the sparse $\mathbf{\Phi}^{-1}$ MGRIT matrix in the implicit case and assuming spatial restriction $\mathbf{R}_{s,i}$ and prolongation $\mathbf{P}_{s,i}$ correspond to time t_i, we write the coarse-grid block equation as

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

and thus we compute

$$\mathbf{u}_{c,i} = \left(\mathbf{R}_{s,i}\Phi_{i,\Delta T}^{-1}\mathbf{P}_{s,i}\right)^{-1}\left[\mathbf{R}_{s,i}\mathbf{P}_{s,i-1}\mathbf{u}_{c,i-1} + \mathbf{R}_{s,i}\mathbf{g}_i\right]. \tag{11}$$

For linear advection the matrix $\mathbf{R}_{s,i}\Phi_{i,\Delta T}^{-1}\mathbf{P}_{s,i}$ is computed as the product of the three sparse matrices $\mathbf{R}_{s,i}$, $\Phi_{i,\Delta T}^{-1}$, and $\mathbf{P}_{s,i}$, which is then factored and stored for future use. In the nonlinear case we first prolong the coarse-grid vector to the previous intermediate grid (coarse-in-time, fine-in-space), evaluate and compute the Jacobian for $\Phi_{i,\Delta T}^{-1}(\mathbf{P}_{s,i}\mathbf{u}_{c,i}) - \mathbf{P}_{s,i-1}\mathbf{u}_{c,i-1} - \mathbf{g}_i = \mathbf{0}$, then restrict both and solve the resulting coarse-grid linear system. Compared to rediscretization we find this definition results in cheaper overall algorithms in the linear case, both in terms of iterations required and overall time to solution, and comparable results in the nonlinear case.

We do not consider defining an explicit time-stepping coarse-grid operator in this way for two reasons. First, it would result in a stricter stability condition when compared to the rediscretized coarse-grid operator. Second, compared to the implicit case, where this definition adds a matrix-vector product to the computational cost of the iteration, in the explicit case the Galerkin definition adds a linear system solve (computing the product as above for the explicit formulation results in a matrix $\mathbf{R}_{s,i}\mathbf{P}_{s,i}$ multiplying $\mathbf{u}_i$ that will need to be inverted), which is not as parallelizable as the initial matrix-vector product required, becoming a significant bottleneck as spatial parallelism is added.

3. Adaptive Spatial Coarsening. 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. The wave speed for a hyperbolic PDE is the derivative of the flux function: $\lambda(u,x,t) := \partial_u f(u,x,t)$, the characteristic speed with which small-amplitude perturbations propagate. For linear advection we have $\lambda(u,x,t) = a(x,t)$, and for the inviscid Burgers equation $\lambda(u,x,t) = u$.

In § 3.1, we provide some motivating examples which illustrate both why spatial coarsening may be desirable, and why adaptive spatial coarsening is necessary in certain cases. In § 3.2, we propose a criterion for determining if spatial coarsening should occur, and provide some examples of the meshes generated by following it. In § 3.3, we describe the cell selection strategies used with explicit and implicit time-stepping, and in § 3.4, we outline a method for moving vectors representing solutions or residuals between grids, which is required for restriction, prolongation, and time-stepping on spatial grids which vary in time.

3.1. Motivating Examples. To illustrate the need for adaptive coarsening we solve the linear advection equation for $(x,t) \in [-2,2]\times[0,4]$ using explicit and implicit schemes with MGRIT, using FCF-relaxation, factor-two temporal coarsening, and either no spatial coarsening (No SC) or uniform factor-two spatial coarsening (SC-2), which employs full weighting restriction and linear interpolation. The stopping condition is based on the size of the ℓ_2 norm of the residual vector, which uses a halting tolerance of 10^{-10} scaled by the domain size: $\text{tol} = (2.5\times10^{-11})\sqrt{N_t N_x}$.

We impose the initial condition $u_0(x) = \sin(0.5\pi x)$ and consider the constant wave speeds

A1. $a(x,t) = 1.0$, and

A2. $a(x,t) = 0.1$,

for which (7) and (8) reduce to simple upwinding. The results for these tests are presented in Table 1, which records iteration count, time to solution, and time per

	$N_x \times N_t$				$2^7 \times 2^8$	$2^8 \times 2^9$	$2^9 \times 2^{10}$	$2^{10} \times 2^{11}$	$2^{11} \times 2^{12}$
Explicit	Case A1	No SC	2-level	It	50	92	100*	100*	100*
				Time (TPI)	0.20 (.004)	0.97 (.01)	3.69 (.03)	16.57 (.16)	59.62 (.59)
			F-cycle	It	100*	100*	100*	100*	100*
				Time (TPI)	1.19 (.012)	4.39 (.04)	14.23 (.14)	51.49 (.51)	327.21 (3.27)
		SC-2	2-level	It	30	30	31	31	31
				Time (TPI)	0.13 (.004)	0.42 (.01)	1.53 (.04)	6.30 (.20)	22.18 (.71)
			F-cycle	It	34	37	41	47	54
				Time (TPI)	**0.30** (.009)	**0.96** (.02)	**3.77** (.09)	**14.73** (.31)	**63.31** (1.17)
	Case A2	No SC	2-level	It	7	7	7	7	7
				Time (TPI)	0.08 (.011)	0.25 (.03)	0.86 (.12)	3.24 (.46)	11.19 (1.59)
			F-cycle	It	8	9	34	100*	100*
				Time (TPI)	0.15 (.019)	0.49 (.05)	5.48 (.16)	53.44 (.53)	199.39 (1.99)
		SC-2	2-level	It	100*	100*	100*	100*	100*
				Time (TPI)	0.34 (.003)	1.17 (.01)	4.72 (.04)	16.15 (.16)	54.83 (.54)
			F-cycle	It	100*	100*	100*	100*	100*
				Time (TPI)	0.77 (.008)	2.27 (.02)	7.38 (.07)	25.62 (.25)	95.90 (.95)
	$N_x \times N_t$				$2^7 \times 2^7$	$2^8 \times 2^8$	$2^9 \times 2^9$	$2^{10} \times 2^{10}$	$2^{11} \times 2^{11}$
Implicit	Case A1	No SC	2-level	It	14	14	15	15	15
				Time (TPI)	0.12 (.009)	0.35 (.02)	1.31 (.08)	5.16 (.34)	26.99 (1.79)
			F-cycle	It	14	15	17	20	22
				Time (TPI)	0.23 (.016)	0.91 (.06)	4.04 (.23)	19.90 (.99)	91.15 (4.14)
		SC-2	2-level	It	15	15	15	16	16
				Time (TPI)	0.09 (.006)	0.32 (.02)	1.18 (.07)	5.01 (.31)	23.72 (1.48)
			F-cycle	It	15	17	20	24	28
				Time (TPI)	**0.15** (.010)	**0.58** (.03)	**2.40** (.12)	**10.56** (.44)	**56.84** (2.03)
	Case A2	No SC	2-level	It	8	8	8	8	8
				Time (TPI)	0.06 (.008)	0.23 (.02)	0.90 (.11)	3.44 (.43)	13.54 (1.69)
			F-cycle	It	8	8	9	9	10
				Time (TPI)	**0.15** (.019)	**0.51** (.06)	**2.14** (.23)	**8.51** (.94)	**37.73** (3.77)
		SC-2	2-level	It	64	90	92	92	92
				Time (TPI)	0.21 (.003)	1.05 (.01)	4.09 (.04)	16.26 (.17)	61.18 (.66)
			F-cycle	It	64	92	94	95	95
				Time (TPI)	0.53 (.008)	2.12 (.02)	7.88 (.08)	30.70 (.32)	114.37 (1.20)

TABLE 1

Linear advection results for Cases A1 and A2. No SC: no spatial coarsening; SC-2: factor-two uniform spatial coarsening. For each problem, the fastest F-cycle results are shown in bold. Asterisks denote tests which failed to converge due to instability (Explicit - No SC) or exceeded 100 iterations.

iteration (TPI). For explicit time-stepping we see the importance of maintaining stability on all levels of the grid hierarchy. For Case A1 the Courant number $\lambda\delta t/\delta x$ for SC-2 is 0.5 on all levels, with temporal coarsening terminating when further spatial coarsening is impossible. Thus, time-stepping is stable on all levels and MGRIT terminates successfully. In contrast, the Courant number for No SC is $2^{\ell-1}$ on level ℓ, where $\ell = 0$ is the finest grid, indicating that time-stepping will be unstable on all coarse levels, hence the majority of 2-level and F-cycle tests failing to converge. However, blindly applying spatial coarsening is not the answer, as illustrated by Case A2 which features a small wave speed that causes weak spatial connections in (7) and (8). Here the Courant number for SC-2 remains fixed at 0.05, hence time-stepping is certainly stable on all levels, but the convergence is extremely poor due to the weak connections. The Courant number for No SC is $0.05(2^\ell)$, hence time-stepping is stable on the first four coarse grids, and thus while the F-cycles become worse as the problem size grows, the two-level method works well.

For implicit time-stepping the No SC and SC-2 methods produce similar results for Case A1 in terms of iteration count, and there can be substantial savings of approximately 30% in terms of time to solution by using spatial coarsening. For Case A2, however, both the iteration count and time to solution for SC-2 are many times

larger than the corresponding values for No SC, making uniform spatial coarsening a non-starter due to the small wave speed.

We observe similar behaviour when solving the inviscid Burgers equation via MGRIT, though in this case the convergence of MGRIT with spatial coarsening depends on the choice of initial condition. If $u_0(x)$ is bounded sufficiently far away from zero we observe results for SC-2 similar to those for Case A1, and if $u_0(x)$ is sufficiently close or equal to zero on part of the domain, we observe convergence issues for MGRIT with spatial coarsening similar to those in Case A2.

Combined, these results indicate why spatial coarsening may be desirable for implicit MGRIT and necessary for explicit MGRIT. For explicit MGRIT, No SC will break down once the coarse-grid time step becomes sufficiently large, though it can work for grid hierarchies with few levels where the maximum wave speed is small enough to ensure stability throughout. In contrast, SC-2 ensures stability on all levels. For both explicit and implicit MGRIT, uniform spatial coarsening (SC-2) can work well when the wave speed is bounded away from zero, but can exhibit extremely poor convergence when the wave speed is small due to the weak spatial connections. This is analogous to the case of multigrid using Gauss-Seidel or weighted Jacobi applied to strongly anisotropic elliptic problems [4]. For implicit time stepping, SC-2 beats No SC in total time-to-solution when the wave speed is bounded away from zero due to the lower work per cycle.

3.2. Adaptive Coarsening Criteria. 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 on 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.

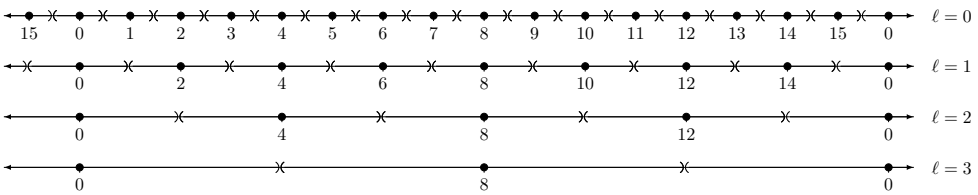

FIG. 3. *Factor-two coarsening in 1D with periodic BCs. The)(symbols represent cell boundaries.*

Considering the discretizations (7-10) and the results of the previous section, we see that a wave speed $\lambda(u, x, t)$ near zero can result in weak couplings in the spatial direction, meaning high frequency errors are not reduced effectively by relaxation. Thus, the error after relaxation cannot be represented properly on coarse spatial grids, drastically reducing the efficiency of a multigrid iteration. Thus, if the wave speed 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:

$$\text{(12)} \qquad \text{Advection: If } \min_{x\in\Omega_j} |\lambda(u,x,t)| \frac{\delta t}{\delta x_j} < \text{tol}_*\text{: keep } \Omega_j\text{; else: coarsen normally.}$$

Since $\lambda(u, x, t)\delta t/\delta x_j$ appears in the coefficients of the equations (7-10), this is an appropriate measure to identify small matrix elements that indicate weak coupling

and may lead to degraded multigrid performance if spatial coarsening is used. This approach has similarities to algebraic multigrid [23], where coarsening is operator dependent, based on the strength of different nodal connections. 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.

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 $\{0,3,4,8,9,12\}$. An array of `grid_info` structures serves as a grid hierarchy for a given time point t_i. Descriptions of the cell selection strategies employed for implicit and explicit time-stepping are described in the following subsections.

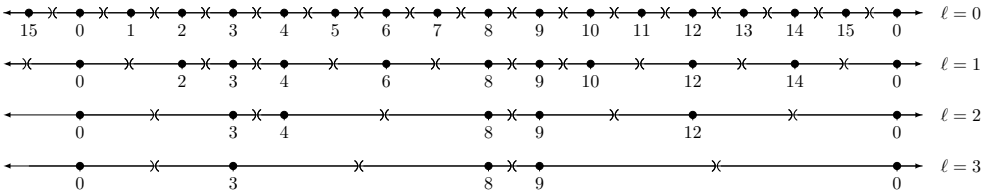

FIG. 4. *Adaptive coarsening in 1D with periodic BCs.*

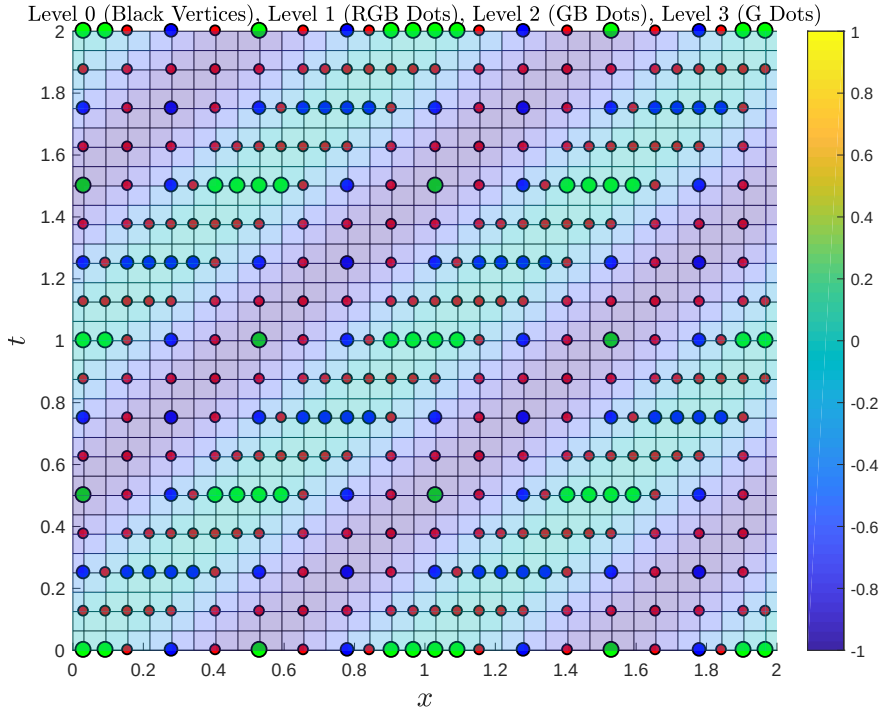

FIG. 5. *Linear advection, $a(x,t) = -\sin^2(\pi(x-t))$ (Case A4): space-time meshes obtained from adaptive spatial coarsening over 4 levels, starting with $N_x = N_t = 64$. The color map indicates the value of $a(x,t)$. Temporal coarsening in MGRIT proceeds in a uniform way, but spatial coarsening is inhibited where $|a|$ is small.*

An example of this coarsening process is shown in Figure 4 for the same fine grid as in Figure 3 at a fixed time point, where (12) happened to be satisfied on all levels in

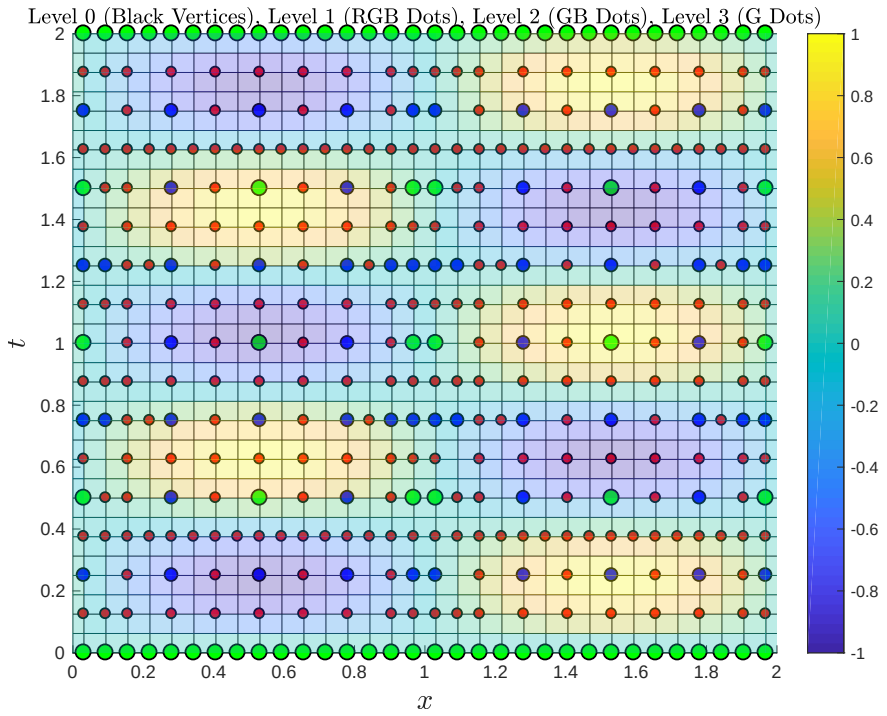

FIG. 6. *Linear advection, $a(x,t) = -\sin(2.5\pi t)\sin(\pi x)$ (Case A5): space-time meshes obtained from adaptive spatial coarsening over 4 levels, starting with $N_x = N_t = 64$. The color map indicates the value of $a(x,t)$. Temporal coarsening in MGRIT proceeds in a uniform way, but spatial coarsening is inhibited where $|a|$ is small.*

cells 3 and 9. The labeled reference points are used to compute cell boundaries as per the definition of vertex-centered grids. It is worth noting that this strategy is easily adapted to non-periodic spatial domains by ensuring that the final cell is retained on all levels. An easy way of doing so is to take $N_x = 2^k + 1$ for some $k \in \mathbb{N}$, which ensures that the final cell is always part of the uniformly coarsened grid, and hence will also always be part of the adaptively coarsened grid.

In Figures 5–7 we show adaptive grid hierarchies generated by three rounds of coarsening, starting from a fine 64×64 space-time grid. In all three cases the 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. It will be shown in § 4 that these grids lead to good MGRIT convergence, and thus adaptive coarsening solves the problem of small local wave speeds.

The first two grids are based on solving the linear advection equation with implicit time-stepping over $[-2,2] \times [0,4]$ for $a(x,t) = -\sin^2(\pi(x-t))$ and $a(x,t) = \frac{1}{2}(1 - \sin(2\pi t))\sin(\pi x)$, respectively (these are Cases A4 and A5 defined in § 4.1). Due to the periodicity of $a(x,t)$ the grid in each quadrant is identical, so we may restrict our discussion to the bottom-right quadrant of each grid, corresponding to $(x,t) \in [0,2] \times [0,2]$. In Figure 5 we see that adaptation results in additional cells being kept along the lines $t = x + b$ for $b \in \mathbb{Z}$, corresponding to the solution of $a(x,t) = 0$. Similarly, in Figure 6 we see adaptivity keeping cells along vertical lines defined by integer values of x and horizontal lines defined by multiples of 0.4 for t. Further, we

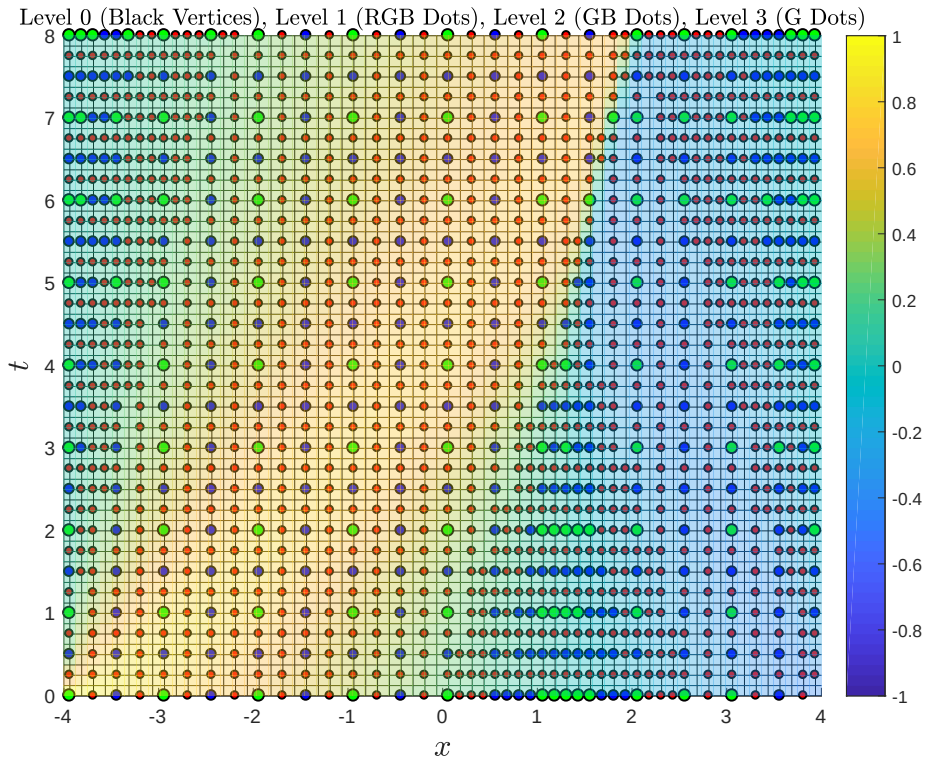

FIG. 7. *Burgers' equation, $u_0(x) = 0.25 - 0.75\sin(\pi x/16)$ (Case B1): space-time mesh obtained from adaptive spatial coarsening over 4 levels, starting with $N_x = N_t = 64$. The color map indicates the value of $u(x,t)$. Temporal coarsening in MGRIT proceeds in a uniform way, but spatial coarsening is inhibited where $|a|$ is small.*

see that by coarsening in time we can eliminate the lines near $t = 0.4$ and $t = 1.6$ where no coarsening has taken place, resulting in cheaper coarse-grid problems with no significant deterioration in the convergence of MGRIT.

The grid in Figure 7 is based on the solution of Burgers' equation over the domain $(x,t) \in [-4,4] \times [0,8]$ with the initial condition

B1. $u_0(x) = 0.25 - \sin(\pi x/16)$.

Due to the initial lack of periodicity in the local wave speed (which is the solution $u(x,t)$, pictured in Figure 8) we show the grid for the entire domain. Once more we see that adaptivity results in more grid cells being retained in regions where the wave speed is near zero, and the location and size of these regions change in response to the evolution of the solution.

3.3. Cell Selection Strategies. The following algorithms are intended as proof of concept for first-order time-stepping routines applied to the linear advection equation and Burgers' equation: further modifications may be required to handle other equations or time-stepping routines. For linear PDEs such as variable coefficient linear advection, the adaptive grid hierarchies generated will not change between MGRIT iterations, so the grids and associated transfer operators need only be computed once and then stored for reuse. In contrast, for nonlinear PDEs such as Burgers' equation the grids can change as the solution approximation is refined, and hence the adaptive grid hierarchy and the transfer operators will need to be recomputed until a certain

MGRIT residual tolerance is reached.

3.3.1. Implicit Time-stepping. In our adaptive coarsening strategy we begin with the grid hierarchy generated by uniform factor-two coarsening, meaning that on level ℓ all cells with global indices that are multiples of 2^ℓ are retained. For implicit time-stepping we then use condition (12) to identify other cells which should be retained due to small local Courant numbers. Note that, for implicit time-stepping, we do not need to worry about violating a stability constraint when retaining spatial cells while increasing δt. Thus when restricting from level ℓ to $\ell+1$, we keep Ω_j^ℓ if

(i) fidx[j] mod $2^\ell = 0$, or
(ii) (12) holds.

For implicit time-stepping we specify the tolerance in the second condition to be $\text{tol}_* = 0.25$. This cell selection strategy is local in scope, so it can be used in both serial and parallel implementations.

3.3.2. Explicit Time-stepping – Linear Advection. To use explicit time-stepping when solving the linear advection equation we must ensure $|a(x,t)|\delta t/\delta x_j < 1$ for numerical stability, which necessitates computing the local Courant number for all cells not part of the uniform coarsening grid hierarchy on each level. We need to find the right balance between removing cells as required for stability, and keeping cells to maintain good multigrid convergence corresponding to (12). If we consider each cell independently, we may inadvertently end up deleting more cells than necessary for stability, leading to poorer MGRIT convergence. Instead, we collectively consider all cells between each subsequent pair of cells that belong to the uniform grid on the current level and decide which of these non-uniform grid cells must be removed for stability and which should be kept for better convergence.

If there is only one cell between two uniform grid cells, we compute

$$\text{test}_- = \frac{|a(x_{j-1/2},t)|\delta t}{\delta x_{j-1/2}} \text{ and } \text{test}_+ = \frac{|a(x_{j+1/2},t)|\delta t}{\delta x_{j+1/2}}$$

and keep the cell if doing so is beneficial for convergence and is not detrimental for stability:

$$\min(\text{test}_-, \text{test}_+) < \text{tol}_* \text{ and } \max(\text{test}_-, \text{test}_+) < \max_*,$$

where we use $\max_* = 0.95$ and set tol_* to be 0.25 if $\ell = 0$, 0.4 if $\ell = 1$, and 0.49 for $\ell \geq 2$. The values for tol_* were tuned by repeated experimentation and are based on the observation that we can afford, from a computational cost perspective, to keep more spatial cells on coarser grids. Otherwise, for each of the cell interfaces we compute

$$\text{test[j]} = \frac{|a(x_{j+1/2},t)|\delta t}{\delta x_{j+1/2}}$$

and based on the value of test[j] the interface is labeled as K (keep), N (neutral), or D (delete). Specifically, if test[j] $< \text{tol}_*$ we label this as K, if test[j] $< \max_*$ we label this as N, and otherwise we label it as D. If the sequence of labels is:

(i) X–D–D–$\cdots$–D–X: delete every second cell between D-interfaces (X = K or N)
(ii) N–D–N: delete both cells.
(iii) K–D–N: delete the right cell.
(iv) N–D–K: delete the left cell.
(v) K–D–K: further consideration is required.

In the last case we compute

$$\text{test}_- = \frac{|a(0.5(x_{j+1} + x_{j-1}), t)|\delta t}{\delta x_j} \text{ and } \text{test}_+ = \frac{|a(0.5(x_j + x_{j+2}), t)|\delta t}{\delta x_{j+1}}$$

which are the coarse-grid local Courant numbers which would result from deleting the left or right cells, respectively. We then perform a sequence of comparisons which is designed to remove both cells if the predicted coarse-grid values are both greater than $\max_*$, delete the opposite cell if only one of the test values is greater than $\max_*$, and otherwise keep the cell with the largest Courant value to maintain good MGRIT convergence.

(i) if $\min(\text{test}_-, \text{test}_+) > \max_*$
Delete both cells
(ii) else if $\text{test}_- > \max_*$
Delete right cell
(iii) else if $\text{test}_+ > \max_*$
Delete left cell
(iv) else if $\min(\text{test}_-, \text{test}_+) > \text{tol}_*$ and $\text{test}_- > \text{test}_+$
Delete right cell
(v) else if $\min(\text{test}_-, \text{test}_+) > \text{tol}_*$ and $\text{test}_- \leq \text{test}_+$
Delete left cell
(vi) else if $\text{test}_- > \text{tol}_*$ and $\text{test}_+ < \text{tol}_*$
Delete right cell
(vii) else if $\text{test}_- < \text{tol}_*$ and $\text{test}_+ > \text{tol}_*$
Delete left cell
(viii) else if $\max(\text{test}_-, \text{test}_+) < \text{tol}_*$ and $\text{test}_- > \text{test}_+$
Delete right cell
(ix) else if $\max(\text{test}_-, \text{test}_+) < \text{tol}_*$ and $\text{test}_- \leq \text{test}_+$
Delete left cell

This process is repeated until no D-labeled interfaces remain. If there are multiple adjacent N-interfaces, we next delete every second cell defined by these interfaces. At the end of this process we are left with the cells that are to be kept to ensure effective MGRIT coarse-grid corrections while maintaining stability.

To adapt this process to allow spatial parallelism we only have to make adjustments to account for how the grid is partitioned over the set of processors. If the first (respectively, last) cell on a given processor is not part of the uniform coarsening grid, then we assume that the final cell on the previous processor (respectively, first cell on the next processor) belongs to the uniform coarsening grid, and perform the previously described sequence of tests.

3.3.3. Explicit Time-stepping – Burgers' Equation. For Burgers' equation we use a more stringent version of the strategy for linear advection because of the greater likelihood of stability related issues arising in the nonlinear case. As before we keep all cells which are part of the uniform grid, and make use of (12) to determine which of the remaining cells will be retained to improve convergence.

If there is only one cell between two uniform grid cells, we compute

$$\text{test}_- = \frac{\max(|u_{j-1}|, |u_j|)\delta t}{\delta x_{j-1/2}} \text{ and } \text{test}_+ = \frac{\max(|u_j|, |u_{j+1}|)\delta t}{\delta x_{j+1/2}}$$

and keep the cell if

$$\max(\text{test}_-, \text{test}_+) < \text{tol}_*,$$

where we set tol_* to be 0.25 if $\ell = 0$, 0.35 if $\ell = 1$, and 0.45 for $\ell \geq 2$. Otherwise, for each of the cell interfaces we compute

$$\text{test[j]} = \frac{|u_j|\Delta t}{\Delta x_{j+1/2}}$$

and if $\text{test[j]} < \text{tol}_*$ we label this as K, otherwise labeling it as D. If there are multiple adjacent D-interfaces we delete every second cell that they define, and for isolated D-interfaces we compute

$$\text{test}_+ = \frac{|u_{j+1}|\delta t}{\delta x_{j+1}} \text{ and } \text{test}_- = \frac{|u_j|\delta t}{\delta x_j}$$

and perform the following sequence of tests.

(i) if $\min(\text{test}_-, \text{test}_+) > \text{tol}_*$
Delete both cells
(ii) else if $\text{test}_- > \text{tol}_*$ and $\text{test}_+ < \text{tol}_*$
Delete right cell
(iii) else if $\text{test}_- < \text{tol}_*$ and $\text{test}_+ > \text{tol}_*$
Delete left cell
(iv) else if $\max(\text{test}_-, \text{test}_+) < \text{tol}_*$ and $\text{test}_- > \text{test}_+$
Delete right cell
(v) else if $\max(\text{test}_-, \text{test}_+) < \text{tol}_*$ and $\text{test}_- \leq \text{test}_+$
Delete left cell

This process is repeated until no D-labeled interfaces remain, at which point the remaining cells are those to be kept to ensure effective MGRIT coarse-grid corrections.

3.4. Movement Between Grids. In addition to 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 grid on a given level may vary with time. This means that a representation of $\mathbf{u}_i$ must be computed on the spatial grid for time t_{i+1} before $\mathbf{u}_{i+1}$ can be computed by time marching.

To map an arbitrary vector $\mathbf{v}$ from grid A to grid B we use the following strategy. For each cell Ω_j^B on grid B, we first identify the cells on grid A that contain its left boundary (Ω_α^A) and right boundary (Ω_ω^A). We compute the cell average v_j^B on Ω_j^B as a weighted average of the cell values from α to ω, scaled by the width of Ω_j^B:

$$v_j^B = \frac{1}{|\Omega_j^B|} \sum_{k=\alpha}^{\omega} |\Omega_j^B \cap \Omega_k^A| v_k^A$$

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.

For factor-two coarsening, this reduces to full weighting restriction and linear interpolation prolongation, which were our initial choices; and if no spatial coarsening is carried out this reduces to $v_j^{\ell+1} = v_j^\ell$. In all cases this approach is conservative.

4. Serial Numerical Results. Numerical results within this section were generated using the XBraid parallel-in-time software package [2], and the CHOLMOD [7] and UMFPACK [9] packages from SuiteSparse for sparse matrix multiplication and factorization, respectively.

$N_x \times N_t$				$2^7 \times 2^7$	$2^8 \times 2^8$	$2^9 \times 2^9$	$2^{10} \times 2^{10}$	$2^{11} \times 2^{11}$
Case A3	No SC	2-level	It	12	14	14	14	14
			Time (TPI)	0.03 (.003)	0.13 (.009)	0.49 (.03)	2.26 (.16)	8.18 (.58)
		F-cycle	It	12	14	16	18	20
			Time (TPI)	**0.11** (.009)	**0.49** (.035)	2.48 (.15)	12.46 (.69)	56.12 (2.80)
	SC-2	2-level	It	64	78	83	84	85
			Time (TPI)	0.15 (.002)	0.70 (.002)	2.82 (.03)	11.54 (.13)	46.31 (.54)
		F-cycle	It	64	80	85	86	87
			Time (TPI)	0.38 (.006)	1.53 (.019)	6.42 (.07)	25.94 (.30)	95.85 (1.10)
	SC-A	2-level	It	26	27	28	29	29
			Time (TPI)	0.06 (.002)	0.25 (.009)	0.99 (.03)	4.17 (.14)	16.51 (.56)
		F-cycle	It	27	27	28	29	30
			Time (TPI)	0.19 (.007)	0.60 (.022)	**2.35** (.08)	**9.45** (.32)	**37.52** (1.25)
Case A4	No SC	2-level	It	12	12	13	13	13
			Time (TPI)	0.06 (.005)	0.23 (.019)	0.94 (.07)	3.82 (.29)	15.07 (1.15)
		F-cycle	It	12	13	15	16	18
			Time (TPI)	**0.14** (.012)	**0.62** (.048)	2.94 (.19)	13.05 (.81)	60.88 (3.38)
	SC-A	2-level	It	16	15	17	19	22
			Time (TPI)	0.08 (.005)	0.26 (.017)	1.13 (.06)	4.70 (.24)	20.16 (.91)
		F-cycle	It	16	18	20	23	28
			Time (TPI)	0.16 (.010)	0.64 (.036)	**2.54** (.12)	**11.01** (.47)	**51.61** (1.84)
Case A5	No SC	2-level	It	13	12	12	12	13
			Time (TPI)	0.06 (.005)	0.23 (.019)	0.92 (.07)	3.54 (.29)	15.03 (1.15)
		F-cycle	It	13	14	14	16	17
			Time (TPI)	**0.15** (.012)	**0.64** (.046)	**2.62** (.18)	11.86 (.74)	51.02 (3.00)
	SC-A	2-level	It	19	19	20	24	26
			Time (TPI)	0.09 (.005)	0.31 (.016)	1.23 (.06)	5.31 (.22)	22.98 (.88)
		F-cycle	It	20	20	22	24	28
			Time (TPI)	0.20 (.010)	0.71 (.036)	2.77 (.12)	**11.52** (.48)	**49.69** (1.77)

TABLE 2

Linear advection: implicit time-stepping results for Cases A3, A4, and A5. No SC: no spatial coarsening; SC-2: factor-two uniform spatial coarsening, SC-A: adaptive spatial coarsening. For each test problem, the fastest F-cycle results are shown in bold.

4.1. Linear Advection. We first revisit the linear advection equation with initial condition $u_0(x) = \sin(0.5\pi x)$ solved over $[-2, 2]\times[0, 4]$ and consider three different variable wave speeds:

A3. $a(x) = -(0.1 + 0.9\cos^2(0.25\pi(x + 2)))$ (a varies in space only),

A4. $a(x, t) = -\sin^2(\pi(x - t))$, and

A5. $a(x, t) = -\sin(2.5\pi t)\sin(\pi x)$.

We refer to these as Cases A3, A4, and A5, respectively, and note cases A4 and A5 were previously used to produce the example grids in Figures 5 and 6. We solve these problems using MGRIT with factor-two temporal coarsening and one of (i) no spatial coarsening, (ii) factor-two spatial coarsening (for Case A3 only), or (iii) adaptive spatial coarsening. All tests again use a halting tolerance of $\text{tol} = (2.5\times10^{-11})\sqrt{N_t N_x}$. Tables 2 and 3 summarize the results for MGRIT using implicit and explicit time integration, respectively.

For implicit Case A3, 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 33% improvement in total time to solution on the No SC F-cycle results in spite of the increased iterations required. For explicit time integration applied to Case A3 we see that SC-A is the only method with convergent F-cycles, with SC-2 once again failing to converge even for two-level methods. Note that, when comparing the entries of Tables 2 and 3, we are not concerned with the

$N_x \times N_t$				$2^7 \times 2^8$	$2^8 \times 2^9$	$2^9 \times 2^{10}$	$2^{10} \times 2^{11}$	$2^{11} \times 2^{12}$
Case A3	No SC	2-level	It	30	47	100*	100*	100*
			Time (TPI)	0.08 (.003)	0.40 (.009)	3.12 (.03)	12.36 (.12)	49.58 (.49)
		F-cycle	It	100*	100*	100*	100*	100*
			Time (TPI)	1.03 (.010)	3.40 (.034)	12.35 (.12)	48.25 (.48)	192.37 (1.92)
	SC-2	2-level	It	100*	100*	100*	100*	100*
			Time (TPI)	0.28 (.003)	0.85 (.009)	3.06 (.03)	11.79 (.11)	46.19 (.46)
		F-cycle	It	100*	100*	100*	100*	100*
			Time (TPI)	0.71 (.007)	1.98 (.020)	6.23 (.06)	22.65 (.22)	82.90 (.82)
	SC-A	2-level	It	30	30	31	31	32
			Time (TPI)	0.09 (.003)	0.30 (.010)	1.12 (.03)	4.20 (.13)	17.27 (.53)
		F-cycle	It	32	33	35	36	37
			Time (TPI)	**0.30** (.009)	**0.95** (.029)	**3.35** (.09)	**12.61** (.35)	**50.70** (1.37)
Case A4	No SC	2-level	It	20	25	31	38	48
			Time (TPI)	0.06 (.003)	0.26 (.010)	1.21 (.03)	5.78 (.15)	25.59 (.53)
		F-cycle	It	100*	100*	100*	100*	100*
			Time (TPI)	0.96 (.010)	3.28 (.033)	12.16 (.12)	46.98 (.46)	190.90 (1.90)
	SC-A	2-level	It	21	23	27	30	30
			Time (TPI)	0.09 (.004)	0.33 (.014)	1.39 (.05)	5.79 (.19)	21.01 (.70)
		F-cycle	It	21	23	28	31	33
			Time (TPI)	**0.31** (.015)	**1.09** (.047)	**4.47** (.15)	**16.76** (.54)	**67.13** (2.03)
Case A5	No SC	2-level	It	29	42	70	100*	100*
			Time (TPI)	0.08 (.003)	0.38 (.009)	2.33 (.03)	13.19 (.13)	49.91 (.49)
		F-cycle	It	100*	100*	100*	100*	100*
			Time (TPI)	1.02 (.010)	3.50 (.035)	12.76 (.12)	48.36 (.48)	186.78 (1.86)
	SC-A	2-level	It	26	26	27	29	30
			Time (TPI)	0.09 (.004)	0.32 (.012)	1.18 (.04)	4.68 (.16)	18.95 (.63)
		F-cycle	It	27	27	28	30	31
			Time (TPI)	**0.36** (.013)	**1.21** (.045)	**4.02** (.14)	**15.61** (.52)	**59.36** (1.91)

TABLE 3

Linear advection: explicit time-stepping results for Cases A3, A4, and A5. No SC: no spatial coarsening; SC-2: factor-two uniform spatial coarsening, SC-A: adaptive spatial coarsening. For each test problem, the fastest F-cycle results are shown in bold. Asterisks denote tests which failed to converge due to instability.

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, which we see for both implicit and explicit discretizations of Case A3.

For Cases A4 and A5, for both types of time integration 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 A3, SC-2 leads to convergence degradation for implicit integration and outright failure for explicit integration (not shown). We see a benefit to using SC-A over No SC for implicit time-stepping in all test problems once the problem size is large enough. The iterations show a moderate increase as a function of problem size. The near scalability for both implicit and explicit results are promising for very large parallel machines, where gains can be expected over sequential time-stepping due to the vastly increased parallelism in MGRIT. Future work will explore eliminating the growth in iteration count for SC-A compared to No SC, while maintaining a similar time per iteration, thus bringing the iteration counts closer to those for No SC implicit timestepping. Such a result would yield significant savings for both implicit and explicit schemes.

4.2. Burgers' Equation. We solve Burgers' equation for Case B1 on the spatial domain $[-4, 4]$. As $u_0'(x) < 0$ at some point in the domain, the wave will break and a

shock will occur. The time at which characteristics cross and a shock forms is called the *breaking time*, T_b, and for the inviscid Burgers equation this time is given exactly as [21]

$$T_b = -\frac{1}{\min(u_0'(x))}.$$

For this particular example we see that the breaking time is $T_b = 16/\pi \approx 5.09$, which matches the solution for the problem illustrated in Figure 8. Based on this observation we solve this problem on both $[-4,4]\times[0,4]$ and $[-4,4]\times[0,8]$ to consider solutions with and without shock. Test results for the half- and full-domain problems are recorded in Tables 4 and 5, respectively.

$N_x \times N_t$				$2^7\times 2^7$	$2^8\times 2^8$	$2^9\times 2^9$	$2^{10}\times 2^{10}$	$2^{11}\times 2^{11}$
Implicit Case B1 $[-4,4]\times[0,4]$ Max fine grid CFL: $(1+2\sqrt{2})/4$	No SC	2-level	It	10	11	11	11	11
			Time (TPI)	1.30 (.13)	5.28 (.48)	19.79 (1.79)	76.80 (6.98)	296.21 (26.92)
		F-cycle	It	11	12	12	14	16
			Time (TPI)	**4.58** (.41)	**21.18** (1.76)	**78.39** (6.53)	**332.34** (23.73)	**1496.17** (93.51)
	SC-G	2-level	It	25	27	28	28	29
			Time (TPI)	3.12 (.12)	12.62 (.46)	46.62 (1.66)	180.30 (6.43)	733.46 (25.29)
		F-cycle	It	26	27	28	29	31
			Time (TPI)	7.88 (.30)	30.73 (1.13)	113.13 (4.04)	426.53 (14.70)	1714.89 (55.31)
$N_x \times N_t$				$2^7\times 2^7$	$2^8\times 2^8$	$2^9\times 2^9$	$2^{10}\times 2^{10}$	$2^{11}\times 2^{11}$
Explicit Case B1 $[-4,4]\times[0,4]$ Max fine grid CFL: $(1+2\sqrt{2})/8$	No SC	2-level	It	25	2*	2*	2*	2*
			Time (TPI)	0.04 (.0001)	0.01 (.005)	0.06 (.03)	0.23 (.11)	0.90 (.45)
		F-cycle	It	2*	2*	2*	2*	2*
			Time (TPI)	0.01 (.005)	0.05 (.025)	0.20 (.10)	0.80 (.40)	3.17 (1.58)
	SC-D	2-level	It	29	31	32	32	32
			Time (TPI)	0.09 (.003)	0.33 (.011)	1.25 (.03)	4.94 (.15)	19.44 (.60)
		F-cycle	It	30	34	36	38	42
			Time (TPI)	**0.24** (.008)	**0.95** (.028)	**3.71** (.10)	**13.96** (.36)	**60.56** (1.44)
$N_x \times N_t$				$2^7\times 2^8$	$2^8\times 2^9$	$2^9\times 2^{10}$	$2^{10}\times 2^{11}$	$2^{11}\times 2^{12}$
Explicit Case B1 $[-4,4]\times[0,4]$ Max fine grid CFL: $(1+2\sqrt{2})/16$	No SC	2-level	It	14	15	15	16	16
			Time (TPI)	0.05 (.004)	0.20 (.013)	0.80 (.05)	3.35 (.20)	13.56 (.84)
		F-cycle	It	2*	2*	2*	2*	2*
			Time (TPI)	0.02 (.010)	0.09 (.045)	0.39 (.19)	1.62 (.81)	6.50 (3.25)
	SC-D	2-level	It	19	21	21	22	22
			Time (TPI)	0.12 (.006)	0.45 (.021)	1.82 (.08)	7.29 (.33)	28.40 (1.29)
		F-cycle	It	19	20	21	24	27
			Time (TPI)	**0.38** (.020)	**1.33** (.067)	**5.32** (.25)	**22.62** (.94)	**97.29** (3.60)

TABLE 4

Burgers' equation results for Case B1: no shock formation. No SC: no spatial coarsening; SC-D: adaptive spatial coarsening with rediscretized coarse-grid operator; SC-G: adaptive spatial coarsening with Galerkin coarse-grid operator. The fastest F-cycle results are shown in bold. Asterisks denote tests which failed due to instability.

For MGRIT using implicit time-stepping the adaptive coarsening method fails to outperform no spatial coarsening in the short domain results due to approximately doubling the iterations required for convergence. Better performance for large grid sizes is observed in the long domain results, due to a relative increase in the no spatial coarsening iteration count and a better time per iteration for the adaptive results (only 46% of the no spatial coarsening time per iteration for the largest test, compared to 59% in the short domain case). Furthermore, the current implementation of the Galerkin definition requires a return to the previous fine grid for each iteration, resulting in an increased time per iteration for adaptive spatial coarsening. This is generally an issue in FAS-style algorithms, which we intend to be a focus of future research.

For explicit time-stepping we first note that the results for both the half and full domain tests are very similar, with the main difference being that those in Table 5 correspond to using twice as many time steps as those in 4 (to maintain the same

$N_x \times N_t$				$2^7 \times 2^7$	$2^8 \times 2^8$	$2^9 \times 2^9$	$2^{10} \times 2^{10}$	$2^{11} \times 2^{11}$
Implicit Case B1 $[-4,4] \times [0,8]$ Max fine grid CFL: $(1+2\sqrt{2})/4$	No SC	2-level	It	12	13	13	14	14
			Time (TPI)	1.85 (.15)	7.57 (.58)	28.72 (2.20)	115.34 (8.23)	444.03 (31.71)
		F-cycle	It	12	14	15	17	20
			Time (TPI)	**6.40** (.53)	**30.87** (2.20)	135.21 (9.01)	606.01 (35.64)	2846.20 (142.31)
	SC-G	2-level	It	26	27	28	28	29
			Time (TPI)	3.86 (.14)	14.98 (.55)	57.18 (2.04)	219.74 (7.84)	905.99 (31.24)
		F-cycle	It	26	27	28	29	30
			Time (TPI)	8.54 (.32)	32.62 (1.20)	**133.85** (4.78)	**514.33** (17.73)	**1961.20** (65.37)

$N_x \times N_t$				$2^7 \times 2^8$	$2^8 \times 2^9$	$2^9 \times 2^{10}$	$2^{10} \times 2^{11}$	$2^{11} \times 2^{12}$
Explicit Case B1 $[-4,4] \times [0,8]$ Max fine grid CFL: $(1+2\sqrt{2})/8$	No SC	2-level	It	35	2*	2*	2*	2*
			Time (TPI)	0.11 (.003)	0.02 (.010)	0.11 (.05)	0.44 (.22)	1.74 (.87)
		F-cycle	It	2*	2*	2*	2*	2*
			Time (TPI)	0.02 (.010)	0.10 (.050)	0.40 (.20)	1.56 (.78)	5.99 (2.99)
	SC-D	2-level	It	31	32	33	33	33
			Time (TPI)	0.18 (.006)	0.64 (.020)	2.42 (.07)	9.13 (.27)	36.93 (1.11)
		F-cycle	It	32	35	37	42	49
			Time (TPI)	**0.50** (.016)	**2.11** (.060)	**7.21** (.19)	**31.75** (.75)	**142.96** (2.91)

$N_x \times N_t$				$2^7 \times 2^9$	$2^8 \times 2^{10}$	$2^9 \times 2^{11}$	$2^{10} \times 2^{12}$	$2^{11} \times 2^{13}$
Explicit Case B1 $[-4,4] \times [0,8]$ Max fine grid CFL: $(1+2\sqrt{2})/16$	No SC	2-level	It	14	15	16	16	17
			Time (TPI)	0.10 (.007)	0.39 (.026)	1.66 (.10)	6.41 (.40)	27.61 (1.62)
		F-cycle	It	2*	2*	2*	2*	2*
			Time (TPI)	0.05 (.025)	0.19 (.095)	0.85 (.42)	2.93 (1.46)	11.54 (5.77)
	SC-D	2-level	It	20	21	22	22	22
			Time (TPI)	0.22 (.011)	0.83 (.040)	3.57 (.16)	14.13 (.64)	52.82 (2.40)
		F-cycle	It	20	20	21	26	31
			Time (TPI)	**0.78** (.039)	**2.78** (.139)	**9.90** (.47)	**45.71** (1.75)	**217.47** (7.01)

TABLE 5

Burgers' equation results for Case B1: with shock formation. No SC: no spatial coarsening; SC-D: adaptive spatial coarsening with rediscretized coarse-grid operator; SC-G: adaptive spatial coarsening with Galerkin coarse-grid operator. The fastest F-cycle results are shown in bold. Asterisks denote tests which failed due to instability.

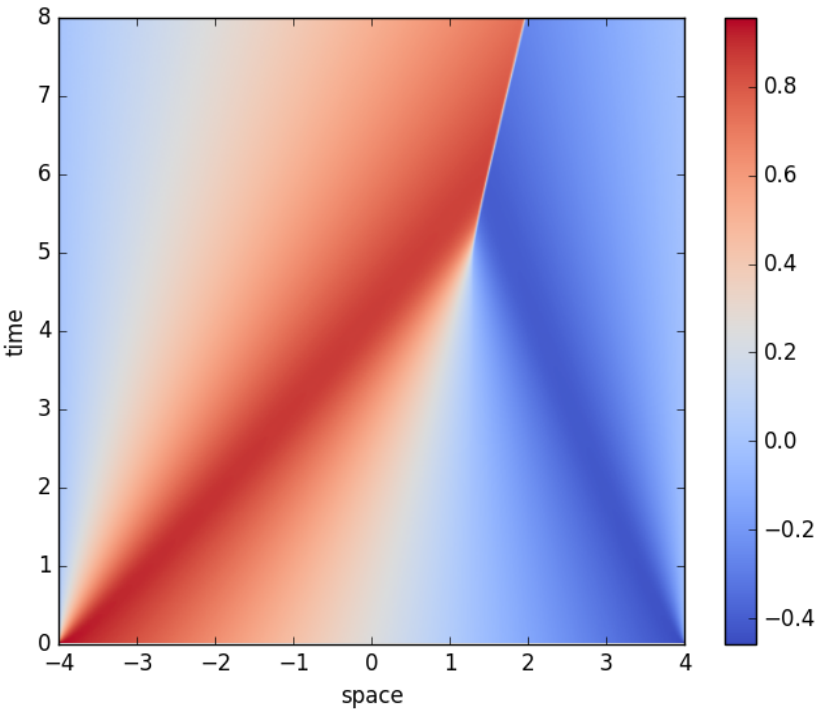

FIG. 8. *Burgers' equation, Case B1: numerical solution on* $[-4,4] \times [0,8]$.

fine-grid Δt in both cases), which results in times that are approximately doubled. Much like in the case of linear advection, spatial coarsening is necessary for convergence. Adaptive coarsening also greatly improves convergence, but, like in the case of linear advection, we observe modest growth in iteration count with problem size and number of levels in the multigrid cycle. Yet, these results are significant, as we have a convergent method for the inviscid Burgers equation with a shock wave, a difficult problem for parallel-in-time methods, and furthermore the presence of the shock does not lead to convergence degradation compared to the smooth solution.

5. Parallel Scaling Results. In this section we present strong and weak parallel scaling results for MGRIT applied to the variable coefficient linear advection equation for $(x,t) \in [-2,2]\times[0,4]$ and $u_0(x) = \sin(0.5\pi x)$ using $a(x,t) = -\sin^2(\pi(x-t))$ (Case A4). The results for $a(x,t) = -\sin(2.5\pi t)\sin(\pi x)$ (Case A5) are also similar, hence are relegated to Supplementary Material Sections SM3 and SM4. Results for explicit time integration are presented in 5.1, followed by results for implicit time integration in 5.2. We consider different combinations of spatial and temporal parallelism, with spatial parallelism implemented using the software package hypre [1] and temporal parallelism implemented using XBraid [2]. These tests were implemented on Vulcan, an IBM Blue Gene/Q machine at Lawrence Livermore National Laboratory consisting of 24,576 nodes, with sixteen 1.6GHz PowerPC A2 cores per node and a 5D Torus interconnect, utilizing up to $2^{17} = 131072$ cores across 8192 nodes.

5.1. Explicit Time-stepping.

5.1.1. Strong Scaling. For strong scaling tests we use a fine space-time mesh specified by $(N_x, N_t) = (2^n, 2^{n+1})$ for $n = 14$, 15, or 16. The results for these cases are presented using figures in the main text, with further details being provided using tables in the supplementary materials. We compare MGRIT F-cycles with factor-two temporal coarsening, adaptive spatial coarsening (coarsening $n-1$ times) and space-time parallelism to serial time-stepping with spatial parallelism. Forward Euler time-stepping requires a matrix-vector multiplication, which is easily parallelized using hypre. For each problem size we set the minimum number of processors in each dimension to be $(p_x, p_t) = (2^a, 2^b)$ for fixed a and b. Processors are allocated to spatial and temporal dimensions in two ways:

(i) $(p_x, p_t) = (2^{a+k}, 2^{b+k})$ for $k = 0, 1, 2, \ldots$,
(ii) $(p_x, p_t) = (2^a, 2^{b+k})$ for $k = 0, 1, 2, \ldots$.

When tabulating results in the supplementary materials we also consider:

(iii) $(p_x, p_t) = (2^{a+k}, 2^b)$ for $k = 0, 1, 2, \ldots$,
(iv) $(p_x, p_t) = (2^k, 2^{P-k})$ for $k = a, a+1, \ldots, P-b$,

where in case (iv) the total number of processors is fixed at 2^P.

		(a,b,n)		
		$(2,3,14)$	$(3,4,15)$	$(4,5,16)$
(p_x, p_t)	$(2^{a+k}, 2^{b+k})$	2.21	2.31	2.06
	$(2^a, 2^{b+k})$	1.97	2.85	4.15

TABLE 6
Best speedup achieved for explicit time-stepping strong scaling tests, $(N_x, N_t) = (2^n, 2^{n+1})$.

While algorithms for serial time-stepping with only spatial parallelism could be optimized differently from algorithms for MGRIT, we choose to use the same framework in both cases with the intent to provide fair, representative comparisons that

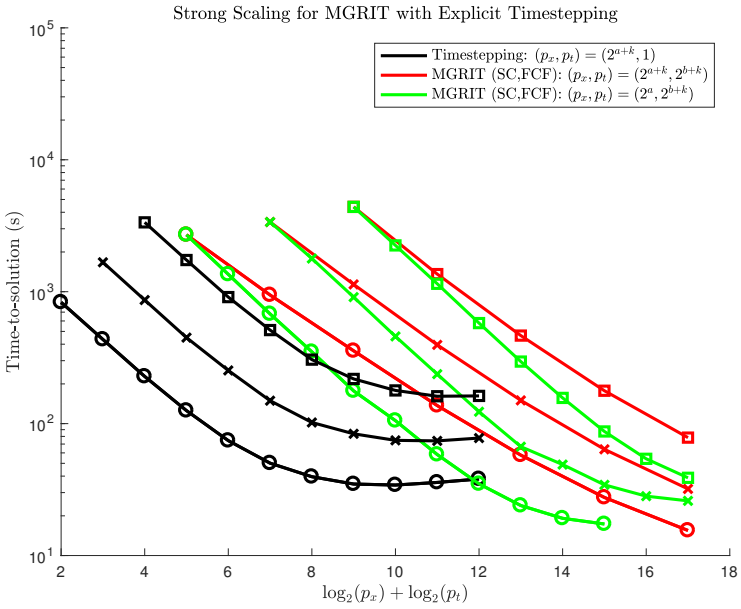

FIG. 9. *Comparison of serial time-stepping with spatial parallelism to MGRIT with FCF relaxation and different combinations of space-time parallelism for three different problem sizes on up to 131072 cores,* $(N_x, N_t) = (2^n, 2^{n+1})$. *These results correspond to Tables SM1–SM4.* ○*:* $n = 14$, $a = 2$, $b = 3$. ×*:* $n = 15$, $a = 3$, $b = 4$. □*:* $n = 16$, $a = 4$, $b = 5$.

would remain consistent for more spatial dimensions and increased problem complexity. Specifically, we use hypre to form and store the sparse matrix used in the matrix-vector product representing a time step.

In Figure 9 we compare serial time-stepping and MGRIT using FCF relaxations for three different problem sizes: $(N_x, N_t) = (2^n, 2^{n+1})$ for $n = 14, 15, 16$. As the basis of comparison we use strong scaling results for serial time-stepping with spatial parallelism. The results for the three different fine grids considered are recorded in Table SM1 and are shown as the black curves in Figure 9. For smaller amounts of parallelism, doubling the problem size in both dimensions roughly quadruples the time to solution, and at the limit of effective parallelism the time to solution approximately doubles as the problem size is increased. The results in Figure 9 are similar for each problem size, where we see that, given enough resources, we are able to improve upon the time-stepping run-times using MGRIT. For a fixed number of processors, the best use of resources is to use the majority for temporal parallelism (green curve) rather than have proportional amounts of temporal and spatial parallelism (red curve). However, when the green curves begin to flatten out there is still potential for more scalability, as indicated by the red curves, suggesting spatial parallelism should be increased when temporal parallelism approaches the saturation point. The best speed-up observed for the cases of $(p_x, p_t) = (2^{a+k}, 2^{b+k})$ (red curve) and $(p_x, p_t) = (2^a, 2^{b+k})$ (green curve) compared to time-stepping (black curve) are recorded in Table 6. Numerical values corresponding to these plots are recorded in Tables SM2–SM4. These Tables illustrate that the iteration count increases modestly with problem size from 37 to 39 to 44, but we do obtain the largest overall parallel speedup for the largest problem size.

5.1.2. Weak Scaling. For weak scaling we increase problem size and processor count while keeping the ratios $N_t : p_t$ and $N_x : p_x$ fixed at $2^{10} : 1$ and the space-time domain fixed at $[-2, 2] \times [0, 4]$. In addition to the original initial condition $u_0(x) = \sin(\pi x/2)$ we also consider the high frequency initial condition $u_0(x) = \sin(2\pi\xi x)$ where ξ is chosen so that there are 16 spatial cells per wavelength. Strong scaling tests were also considered for this initial condition, but the run-times observed were within a few percent of those for the low frequency initial condition, hence are omitted.

We start with a grid of size $(N_x, N_t) = (2^{10}, 2^{11})$ and either double both N_x and N_t at each step (Table 7) or double N_t while leaving N_x fixed (Table 8); we cannot increase N_x while leaving N_t fixed due to the CFL condition. If N_x and N_t are increased simultaneously, while increasing core counts from 2 to 512, and problem size from 2M to 512M degrees of freedom, we see only a factor 2 increase in solution time, indicating excellent weak parallel scaling of the MGRIT algorithm. If we increase N_t while N_x remains fixed, we observe decreases in the iteration count and time to solution due to the increasingly weak couplings in space bringing MGRIT closer to an exact solver (when $a(x, t) = 0$ MGRIT with no spatial coarsening converges in one iteration, and in this case the adaptive coarsening forces all spatial cells to be kept on all levels). It is interesting to observe that the results for the different initial conditions are extremely similar, suggesting that the scalability is robust for oscillatory solutions, where the solution is changing at the scale of the grid spacing.

					Original			Oscillatory		
Trial	$\log_2(N_x)$	$\log_2(N_t)$	$\log_2(p_x)$	$\log_2(p_t)$	ξ	Iter	Time	ξ	Iter	Time
1	10	11	0	1	1/4	31	184.83	16	31	183.20
2	11	12	1	2	1/4	33	234.81	32	33	234.19
3	12	13	2	3	1/4	34	247.17	64	34	246.70
4	13	14	3	4	1/4	36	310.55	128	36	309.92
5	14	15	4	5	1/4	39	359.84	256	39	376.04

TABLE 7
Weak scaling for explicit MGRIT F-cycles with $u_0(x) = \sin(2\pi\xi x)$ and increasing N_x and N_t.

					Original			Oscillatory		
Trial	$\log_2(N_x)$	$\log_2(N_t)$	$\log_2(p_x)$	$\log_2(p_t)$	ξ	Iter	Time	ξ	Iter	Time
1	10	11	0	1	1/4	31	184.84	16	31	183.20
2	10	12	0	2	1/4	14	110.47	16	14	109.81
3	10	13	0	3	1/4	11	98.35	16	12	104.09
4	10	14	0	4	1/4	9	88.08	16	10	94.11
5	10	15	0	5	1/4	7	76.70	16	8	82.96
6	10	16	0	6	1/4	6	71.93	16	6	71.68
7	10	17	0	7	1/4	5	68.24	16	5	68.06

TABLE 8
Weak scaling for explicit MGRIT F-cycles with $u_0(x) = \sin(2\pi\xi x)$ and fixed N_x.

5.2. Implicit Results.

5.2.1. Strong Scaling. For implicit time-stepping we use a fine space-time mesh with equal resolution in both dimensions specified by $(N_x, N_t) = (2^{14}, 2^{14})$ and set the tolerance in our coarsening condition (12) to be $\text{tol}_* = 0.25$. Serial time-stepping with spatial parallelism is compared to MGRIT F-cycles with factor two temporal

coarsening, either no spatial coarsening or adaptive spatial coarsening (coarsening $n-1$ times), and space-time parallelism. Backward Euler time-stepping requires tridiagonal solves which are parallelized by using the hypre 1D cyclic reduction solver. Processors are allocated as in § 5.1 for the explicit case, except that we start with $a = b = 2$.

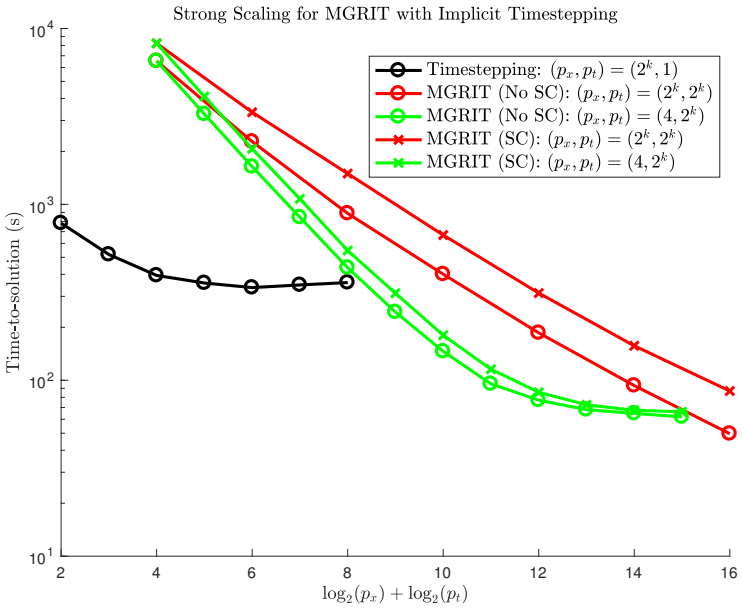

FIG. 10. *Comparison of serial time-stepping with spatial parallelism to MGRIT using FCF relaxation with or without spatial coarsening for different combinations of space-time parallelism on up to 65536 cores. These results correspond to Tables SM5–SM7.*

(p_x, p_t)	No SC	SC
$(2^k, 2^k)$	6.77	3.87
$(2^4, 2^k)$	5.43	5.08

TABLE 9
Best speedup achieved for implicit time-stepping strong scaling tests, $(N_x, N_t) = (2^{14}, 2^{14})$.

In Figure 10 we compare the results of serial implicit time-stepping to MGRIT with FCF temporal relaxation and either with or without spatial coarsening. As the basis of comparison we use strong scaling results for serial time-stepping with spatial parallelism (black curves), as recorded in Table SM5. Significant improvements on the serial time-stepping results are possible once enough temporal parallelism has been introduced. Similar to the explicit case, we see that for up to 2^{14} processors the best results are obtained by investing the majority into temporal parallelism, though further scalability is possible if spatial parallelism is increased as temporal parallelism approaches the saturation point, which would offer improved results for 2^{12} or more processors. The difference between spatial coarsening and no spatial coarsening is most pronounced in the cases where $p_x = p_t$, with the difference between the SC and no SC curves remaining nearly constant as the processor count increases. The best speedup for the No SC and SC cases are recorded in Table 9.

In Tables SM6 and SM7 we tabulate the results from the previous figure. Comparing the SC and no SC results, we see that the SC iteration counts are approximately 1.5 times as large as the iteration counts for no SC (increasing from 26 to 40), indicating that if this increase can be ameliorated we could see significant improvements in the SC time to solution.

5.2.2. Weak Scaling. For weak scaling tests we again consider the original initial condition $u_0(x) = \sin(\pi x/2)$ and the high frequency initial condition $u_0(x) = \sin(2\pi\xi x)$, keeping the ratios $N_t : p_t$ and $N_x : p_x$ fixed at $2^{10} : 1$ while solving over the fixed domain $[-2, 2] \times [0, 4]$. We start with a grid of size $(N_x, N_t) = (2^{10}, 2^{10})$ and (i) double both dimensions at each step, (ii) double N_t, leaving N_x fixed, or (iii) double N_x, leaving N_t fixed; results for these cases are recorded in Tables 10, 11, and 12, respectively. The results for the first two cases are similar to those for explicit time-stepping, though the results of Table 10 show a nearly sixfold increase in the time-to-solution from the smallest to largest test cases, compared to times approximately doubling in the explicit case. This is likely due to the fact that the exact cyclic reduction linear solve used in implicit MGRIT has less potential for spatial parallelism compared to the matrix-vector product required for explicit MGRIT. The third case, unique to the implicit timestepping context, shows that increasing N_x while N_t remains fixed results in a nearly constant iteration count and an increasing time to solution. Considering the results for all three cases, it appears that the growth in iteration count due to increasing problem size is primarily a result of increasing N_t while maintaining a fixed ratio for $\Delta t : \Delta x$.

					Original			Oscillatory		
Trial	$\log_2(N_x)$	$\log_2(N_t)$	$\log_2(p_x)$	$\log_2(p_t)$	ξ	Iter	Time	ξ	Iter	Time
1	10	10	0	0	1/4	21	239.23	16	23	276.81
2	11	11	1	1	1/4	25	554.72	32	25	596.79
3	12	12	2	2	1/4	29	808.77	64	29	859.19
4	13	13	3	3	1/4	37	1167.96	128	37	1245.90
5	14	14	4	4	1/4	45	1401.77	256	46	1489.30

TABLE 10
Weak scaling for implicit MGRIT F-cycles with $u_0(x) = \sin(2\pi\xi x)$ and increasing N_x and N_t.

					Original			Oscillatory		
Trial	$\log_2(N_x)$	$\log_2(N_t)$	$\log_2(p_x)$	$\log_2(p_t)$	ξ	Iter	Time	ξ	Iter	Time
1	10	10	0	0	1/4	21	240.45	16	23	276.54
2	10	11	0	1	1/4	25	306.88	16	25	328.86
3	10	12	0	2	1/4	13	208.58	16	15	250.45
4	10	13	0	3	1/4	10	177.35	16	12	217.68
5	10	14	0	4	1/4	9	166.21	16	10	191.35
6	10	15	0	5	1/4	7	139.22	16	8	162.35
7	10	16	0	6	1/4	5	112.63	16	7	148.56
8	10	17	0	7	1/4	5	114.46	16	5	120.30
9	10	18	0	8	1/4	4	103.10	16	4	107.93

TABLE 11
Weak scaling for implicit MGRIT F-cycles with $u_0(x) = \sin(2\pi\xi x)$ and fixed N_x.

					Original			Oscillatory		
Trial	$\log_2(N_x)$	$\log_2(N_t)$	$\log_2(p_x)$	$\log_2(p_t)$	ξ	Iter	Time	ξ	Iter	Time
1	10	10	0	0	1/4	21	258.54	16	23	276.77
2	11	10	1	0	1/4	22	450.75	32	22	451.55
3	12	10	2	0	1/4	24	592.95	64	24	591.35
4	13	10	3	0	1/4	24	640.04	128	24	636.20
5	14	10	4	0	1/4	25	713.46	256	25	709.57
6	15	10	5	0	1/4	25	769.92	512	25	766.19
7	16	10	6	0	1/4	25	827.91	1024	25	823.27
8	17	10	7	0	1/4	24	872.87	2048	24	867.65
9	18	10	8	0	1/4	24	959.04	4096	24	953.32

TABLE 12
Weak scaling for implicit MGRIT F-cycles with $u_0(x) = \sin(2\pi\xi x)$ and fixed N_t.

6. Conclusions. In this paper we discuss an adaptive spatial coarsening strategy for MGRIT applied to hyperbolic PDEs in one spatial dimension. We observe 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, while the results are nearly scalable as a function of problem size, there is an increase in iterations required for MGRIT to converge when spatial coarsening is introduced, compared to no spatial coarsening, which is the subject of ongoing research.

To our best knowledge, we obtain the first convergent parallel-in-time method for the inviscid Burgers equation, and solutions with shocks do not exhibit convergence deterioration. Parallel results on up to 131072 cores illustrate robustness and scalability of the approach for very large problem sizes, and its potential to achieve run-time speedups when spatial parallelism alone saturates. Weak scaling results show that the scalability is robust for solutions with oscillations on the scale of the grid spacing.

One area of future improvement is load balancing, as to ensure that processors have approximately equal spatial cell counts on coarse grids as the adaptation proceeds. Ongoing research includes mode analysis to understand convergence deterioration and aims to improve iteration counts by considering adding waveform relaxation on intermediate grids. As the adaptive coarsening strategy is extensible, in principle, to 2D and 3D, future plans for solving hyperbolic problems with MGRIT involve implementing adaptive spatial coarsening for problems in two or more spatial dimensions and/or systems of hyperbolic equations.

REFERENCES

[1] *HYPRE: Scalable linear solvers and multigrid methods*. http://llnl.gov/casc/hypre.
[2] *XBraid: Parallel multigrid in time*. http://llnl.gov/casc/xbraid.
[3] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31 (1977), pp. 333–390.
[4] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, SIAM, 2000.
[5] P. CHARTIER AND B. PHILIPPE, *A parallel shooting technique for solving dissipative ODE's*, Computing, 51 (1993), pp. 209–236.
[6] F. CHEN, J. S. HESTHAVEN, AND X. ZHU, *On the use of reduced basis methods to accelerate and stabilize the parareal method*, in Reduced Order Methods for Modeling and Computational Reduction, Springer, 2014, pp. 187–214.
[7] 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.

[8] X. Dai and Y. Maday, *Stable parareal in time method for first-and second-order hyperbolic systems*, SIAM Journal on Scientific Computing, 35 (2013), pp. A52–A78.
[9] T. A. Davis, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Softw., 30 (2004), pp. 196–199.
[10] 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.
[11] 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.
[12] R. D. Falgout, A. Katz, T. V. Kolev, J. B. Schroder, A. Wissink, and U. M. Yang, *Parallel time integration with multigrid reduction for a compressible fluid dynamics application*, (2015).
[13] R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, *Multigrid reduction in time for nonlinear parabolic problems: A case study*, SIAM J. Sci. Comput. (to appear), (2016). LLNL-JRNL-692258.
[14] M. Gander and M. Petcu, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, in ESAIM: Proceedings, vol. 25, EDP Sciences, 2008, pp. 114–129.
[15] M. J. Gander, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 69–113.
[16] 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.
[17] 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.
[18] G. Horton, *The time-parallel multigrid method*, Communications in Applied Numerical Methods, 8 (1992), pp. 585–595.
[19] W. Hundsdorfer and J. G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33, Springer Science & Business Media, 2013.
[20] M. Lecouvez, R. D. Falgout, C. S. Woodward, and P. Top, *A parallel multigrid reduction in time method for power systems*, in Power and Energy Society General Meeting (PESGM), 2016, IEEE, 2016, pp. 1–5.
[21] R. J. LeVeque, *Numerical methods for conservation laws*, vol. 132, Springer, 1992.
[22] 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.
[23] J. W. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid methods, 3 (1987), pp. 73–130.
[24] D. Ruprecht, *Wave propagation characteristics of parareal*, arXiv preprint arXiv:1701.01359, (2017).
[25] D. Ruprecht and R. Krause, *Explicit parallel-in-time integration of a linear acoustic-advection system*, Computers & Fluids, 59 (2012), pp. 72–83.
[26] B. Southworth, T. Manteuffel, S. McCormick, S. Munzenmaier, and J. Ruge, *Reduction-based algebraic multigrid for upwind discretizations*, arXiv preprint arXiv:1704.05001, (2017).
[27] S. Vandewalle and E. Van de Velde, *Space-time concurrent multigrid waveform relaxation*, Annals of Numer. Math, 1 (1994), pp. 347–363.