SIAM J. SCI. COMPUT.
Vol. 41, No. 1, pp. A538–A565

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

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

DOI. 10.1137/17M1144982

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. The need for parallel-in-time is further exacerbated

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 28, 2017; accepted for publication (in revised form) November 21, 2018; published electronically February 12, 2019.

http://www.siam.org/journals/sisc/41-1/M114498.html

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-JRNL737050). The work of the fourth author was partially supported by an NSERC discovery grant. The work of the first and third authors was partially supported by an ARC discovery project.

†School of Arts and Sciences, Red Deer College, Red Deer, AB, T4N 5H5 Canada (ajmhowse@gmail.com).

‡School of Mathematical Sciences, Monash University, Melbourne, VIC 3800, Australia (hans.desterck@monash.edu).

§Center for Applied Scientific Computing, Lawrence Livermore National Lab, Livermore, CA 94551-0808 (falgout2@llnl.gov).

¶Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL, A1C 5S7 Canada (smaclachlan@mun.ca).

‖Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87106 (jbschroder@unm.edu).

by communication latency costs, which limit the degree to which codes can parallellize in space effectively; i.e., strong scaling in space often becomes inefficient when less than a few hundred or a few thousand spatial degrees-of-freedom remain per core. In such cases, parallelism in time can lead to further speedups. 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 [18]. Variants include direct methods and iterative methods based on deferred corrections [12], domain decomposition [20], multigrid [21, 3], multiple shooting [6], and waveform relaxation [33] 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 [28].

For example, one of the most influential parallel-in-time methods is parareal [25], an iterative predictor-corrector method (that is equivalent to a 2-level multigrid scheme [19]) which combines the use of a coarse time integrator in serial and a fine time integrator in parallel. However, highly advective problems (and hyperbolic problems in general) are known to be difficult for parareal [19]. In general, the more dissipation present in an advective problem, the faster parareal converges [31]. Thus, the highly advective case remains a topic of active research. The work [16] proposes the parareal implicit time-integrator method (PITA), which is further developed and interpreted as a Krylov subspace enhanced parareal method [9, 29, 17]. This method stabilizes parareal for hyperbolic problems, but in contrast to the present work this method requires the large cost of storing the space-time solution vector from each parareal iteration. Other notable parareal variants for hyperbolic problems [10, 7] have similarly stabilized parareal by projecting coarse grid error corrections onto subspaces incorporating previously computed information. More recently, the work [26] showed improved parareal performance by pipelining several parareal solves applied to successive intervals of the time-line. This approach takes advantage of the fact that parareal converges faster for smaller time windows (i.e., smaller amounts of time parallelism), even for advective problems [32]. In contrast, our work considers applying massive parallelism to the entire time domain. In general, future performance gains for parareal applied to hyperbolic problems appear to depend on finding coarse time propagators that match the fine-grid phase as closely as possible [28].

In this paper, we discuss the multigrid reduction-in-time (MGRIT) method [13] and use XBraid [2], an open-source implementation of MGRIT. This work takes a multilevel approach, as opposed to the previous 2-level parareal work, and builds on the insight from [32], which shows, by investigating iteration counts and asymptotic convergence rates theoretically and using serial numerical tests, that MGRIT with multiple levels, small per-level temporal coarsening factors, and FCF-relaxation can converge for advection-dominated advection-diffusion problems (including some hyperbolic problems with wave speed 1), when parareal does not. A strength of the MGRIT framework is its nonintrusive 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 [13] and nonlinear [15] parabolic partial differential equations (PDEs) in multiple dimensions, the Navier–Stokes equations [14], and power system models [23]. 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. As will be explained in more detail in section 3.1, 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 structured as follows. In section 2, we describe the MGRIT algorithm and the discretization of (1). In section 3, we present our adaptive coarsening approach, providing algorithms for grid coarsening and transferring solutions between different spatial grids. In section 4, we provide serial numerical results illustrating the efficacy of the adaptive coarsening strategy. In section 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 section 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 [13, original figure]. 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 2-level MGRIT, this coarse-grid problem is solved exactly, whereas multilevel MGRIT applies this process recursively.

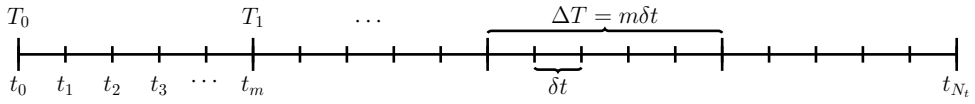

FIG. 1. *Fine and coarse temporal grids.*

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 [13, original figure]. In particular, note that 2-level MGRIT with F-relaxation is equivalent to parareal [13, 19]. 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 [13].

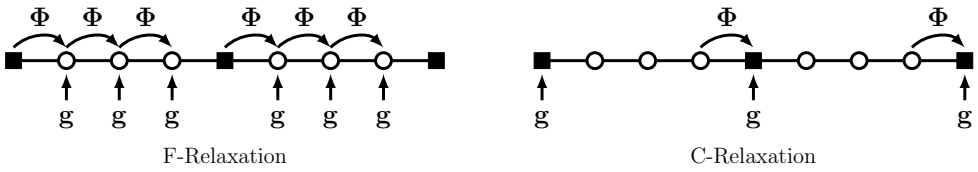

FIG. 2. *Illustration of F- and C-relaxation on a* 9*-point temporal grid with coarsening factor* 4*.*

MGRIT uses the full approximation storage (FAS) framework [4] 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 [14], though we instead reproduce here the variant from [15] which accounts for the possibility of spatial coarsening; see Algorithm 1. We 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.

Using MGRIT to solve the system in (2) requires an increase in overall memory, but it allows us to effectively make use of additional parallel resources, which includes more available memory. In particular, with spatial coarsening and enough parallelism in time, the memory use of MGRIT per processor is a small multiple of that required in the serial time-stepping case (for a spatial coarsening factor of m, the multiplier is

Algorithm 1 FAS-MGRIT.

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

$m/(m-1)$).

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 to ensure a nonconstant solution on the majority of $[a, b]$ for all times. The methods considered are essentially unchanged when Dirichlet boundary conditions are used, and numerical results for Dirichlet conditions are very similar to those reported later for periodic boundary conditions.

We use the vertex-centered approach to construct spatial grids [22, section 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

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

Defining $\delta x_j = \frac{1}{2}(x_{j+1} - x_{j-1})$, (1) is semidiscretized in space as [22]

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

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

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

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 to

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

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)

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

and

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

for linear advection, and the fully discrete equations

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

and

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

for Burgers' equation.

We note that these first-order schemes exhibit significant numerical diffusion, which may contribute to the efficient convergence of MGRIT, based on the strong performance of MGRIT for diffusive parabolic problems [13].

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 symmetric positive definite (SPD) problems [5], and they have also been used for nonsymmetric matrices, for example, in [30]. We use a Galerkin approach in this paper for implicit time-stepping, 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 section 3.1, we provide some motivating examples which illustrate why adaptive spatial coarsening is necessary in certain cases. In section 3.2, we propose a criterion for determining whether spatial coarsening should occur, and provide some examples of the meshes generated by following it. In section 3.3, we describe the cell selection strategies used with explicit and implicit time-stepping, and in section 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. Iteration counts for these tests are presented in Table 1. The table compares MGRIT without spatial coarsening (No SC)

TABLE 1

Linear advection results (number of iterations required for convergence) for Cases A1 *and* A2. *No SC: no spatial coarsening; SC-2: factor-two uniform spatial coarsening. Asterisks denote tests which failed to converge due to instability (Explicit - No SC) or exceeded* 100 *iterations due to poor MG convergence. The boxes in the "Explicit" column indicate runs where instability is expected (since explicit methods require spatial coarsening for stability). The box in the bottom block row indicates runs where uniform spatial coarsening leads to a severe growth in iteration numbers (compared to the* $a = 1$ *case), since adaptive spatial coarsening is required when the advection speed* a *is small.*

			Implicit			Explicit		
$N_x \times N_t$			$2^7 \times 2^7$	$2^9 \times 2^9$	$2^{11} \times 2^{11}$	$2^7 \times 2^8$	$2^9 \times 2^{10}$	$2^{11} \times 2^{12}$
$a = 1.0$	No SC	2-level	14	15	15	50	100*	100*
		F-cycle	14	17	22	100*	100*	100*
	SC-2	2-level	15	15	16	30	31	31
		F-cycle	15	20	28	34	41	54
$a = 0.1$	No SC	2-level	8	8	8	7	7	7
		F-cycle	8	9	10	8	34	100*
	SC-2	2-level	64	92	92	100*	100*	100*
		F-cycle	64	94	95	100*	100*	100*

with MGRIT using uniform spatial coarsening with a coarsening factor of 2 (SC-2), for implicit and explicit time integration.

We first observe that, with No SC, the 2-level and F-cycle variants of MGRIT converge in a modest number of iterations, except in the explicit case (boxes in the right column of the table), where propagation on coarse levels is unstable unless spatial coarsening is employed (or unless the fine-level Courant number $\lambda \delta t / \delta x$ is very small). With the grids employed in the table, for case $a = 1.0$ the Courant number for No SC is $2^{\ell-1}$ on level ℓ, where $\ell = 0$ is the finest grid, indicating that explicit time-stepping will be unstable on all coarse levels. For case $a = 0.1$, the Courant number for No SC is $0.05(2^{\ell})$; hence time-stepping is stable on the first four coarse grids. Thus, while the F-cycles with No SC become worse as the problem size grows, the 2-level method still works well.

Next, the table shows that the instabilities that plague explicit time integration without spatial coarsening are remedied by considering spatial coarsening.

Uniform spatial coarsening by a factor of 2 (SC-2) works well for $a = 1.0$. For the explicit case, it removes the instability, and for the implicit case, it results in cheaper cycles without increasing the iteration counts too much. Further examples in section 4 will show that this may lead to speedup for the implicit case.

However, the situation is different for the case $a = 0.1$. Here, uniform spatial coarsening by a factor of 2 (SC-2) removes the instability for the explicit case, since the Courant number for SC-2 remains fixed at 0.05; hence time-stepping is certainly stable on all levels. However, for both implicit and explicit time integration, the number of multigrid iterations required for convergence is very high due to the weak spatial connections in (7) and (8) caused by the small wave speed. The fundamental reason for this convergence degradation can be illustrated easily, for example, for the case of linear advection with constant wave speed $a > 0$ and forward Euler time discretization, in which case method (7) reduces to

$$u_j^{i+1} + \left(\frac{a\delta t}{\delta x} - 1 \right) u_j^i - \frac{a\delta t}{\delta x} u_{j-1}^i = 0.$$

When $a\delta t / \delta x = \epsilon \ll 1$, the connection in the temporal direction is of size $O(1)$, whereas the connection in the spatial direction is $O(\epsilon)$. This means that relaxation can be effective in removing high-frequency error in the temporal direction, but not in the

spatial direction. Also, when spatial coarsening is employed, the coarse-grid correction cannot correct the high-frequency error that remains in the spatial direction after relaxation, since the coarsened spatial grid cannot represent high-frequency spatial error. As a consequence, the high-frequency spatial error cannot be removed either by relaxation or by coarse-grid correction, and multigrid convergence stalls. This is analogous to the case of multigrid using Gauss–Seidel or weighted Jacobi applied to strongly anisotropic elliptic problems [5].

We observe similar behavior 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.

Our goal is to develop MGRIT methods that employ spatial coarsening to make explicit time integration stable for explicit methods, and to make implicit cycles computationally less expensive. The motivating results in Table 1 indicate that uniform spatial coarsening does not lead to an efficient MGRIT method when wave speeds are small. For this reason, we develop an adaptive spatial coarsening strategy in the next subsections.

3.2. Adaptive coarsening criteria. The one-dimensional (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.

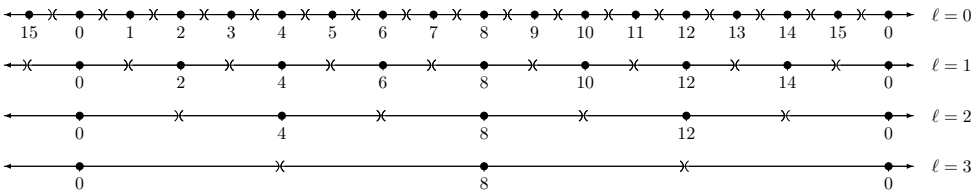

FIG. 3. *Factor-two coarsening in one dimension with periodic boundary conditions (BCs). The)(symbols represent cell boundaries.*

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:

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

Since $\lambda(u,x,t)\delta t/\delta x_j$ appears in the coefficients of (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 [27], 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 six cells, which have local indices $\{0,\dots,5\}$ 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.

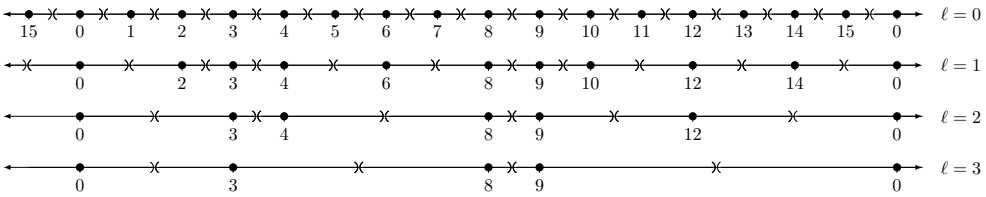

FIG. 4. *Adaptive coarsening in one dimension with periodic BCs.*

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 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 nonperiodic 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 section 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 section 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 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.

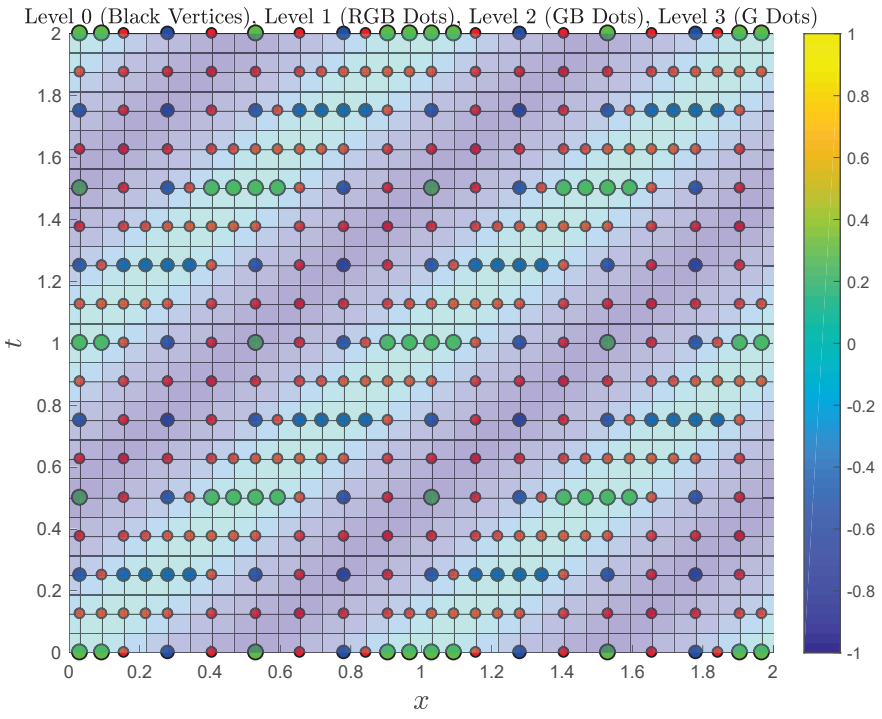

FIG. 5. *Linear advection,* $a(x,t) = -\sin^2(\pi(x-t))$ *(Case* A4*): space-time meshes obtained from adaptive spatial coarsening over four levels, starting with* $N_x = N_t = 64$*. The color map indicates the value of* $a(x,t)$*. Temporal coarsening in MGRIT proceeds in a uniform way, but spatial coarsening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

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

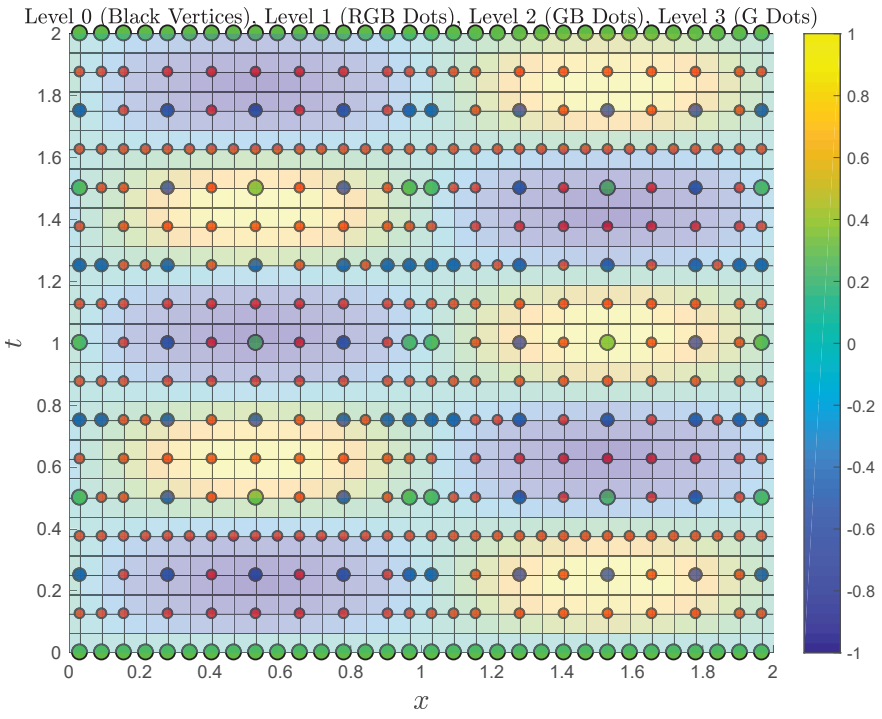

FIG. 6. *Linear advection,* $a(x,t) = -\sin(2.5\pi t)\sin(\pi x)$ *(Case* A5*): space-time meshes obtained from adaptive spatial coarsening over four levels, starting with* $N_x = N_t = 64$*. The color map indicates the value of* $a(x,t)$*. Temporal coarsening in MGRIT proceeds in a uniform way, but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

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) $\text{fidx}[j] \bmod 2^\ell = 0$ 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 nonuniform grid cells must be removed for stability and which should be kept for better convergence.

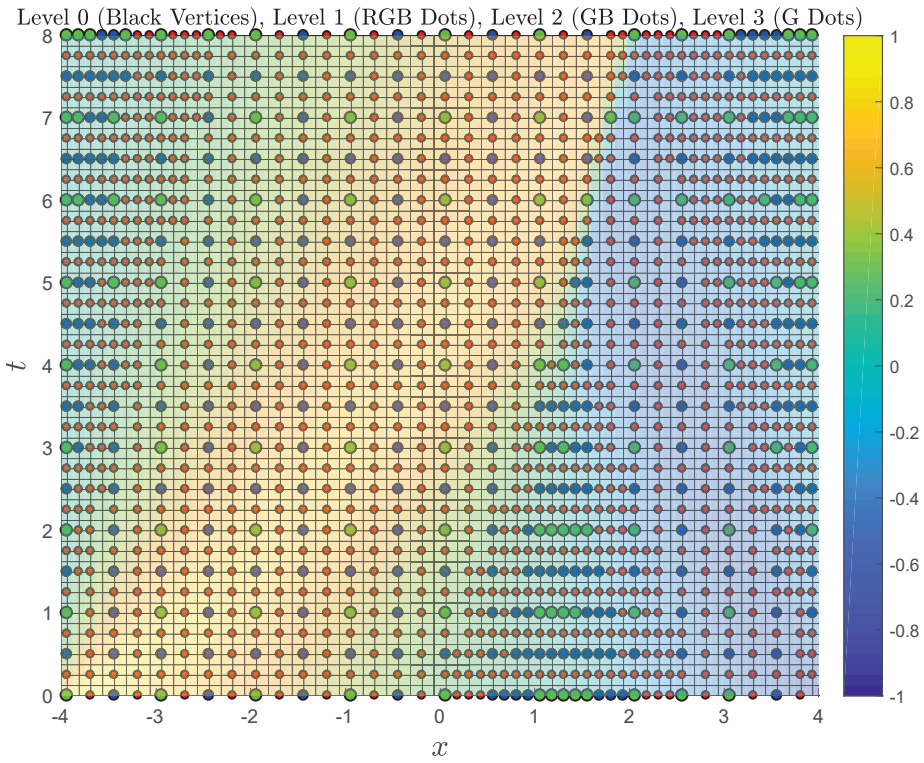

FIG. 7. *Burgers' equation,* $u_0(x) = 0.25 - 0.75\sin(\pi x/16)$ *(Case* B1*): space-time mesh obtained from adaptive spatial coarsening over four levels, starting with* $N_x = N_t = 64$*. The color map indicates the value of* $u(x,t)$*. Temporal coarsening in MGRIT proceeds in a uniform way, but spatial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

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

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

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

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

where we use $\max_* = 0.95$ to be strictly less than the CFL limit of 1 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; hence we can raise the value below which we require a cell to be kept. 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 matches a case below, we take the corresponding action.

(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

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

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

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

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

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

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 [8] and UMFPACK [11] packages from SuiteSparse for sparse matrix multiplication and factorization, respectively.

TABLE 2

Linear advection: implicit and 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. Asterisks denote tests which exceeded* 100 *iterations due to poor MG convergence. Explicit results for No SC are not shown since these simulations are unstable on coarse levels. For case* A3*, adaptive spatial coarsening (SC-A) cures the MG convergence problems that uniform spatial coarsening (SC-2) experiences due to small wavespeeds; see also Table* 1*. (This occurs similarly for cases* A4 *and* A5*, where the SC-2 results are not shown to save space.) Run-times in seconds are indicated in brackets for the largest problem sizes where relevant. For the implicit F-cycle runs, the bold run-times show that adaptive spatial coarsening (SC-A) has the potential to speed up implicit multilevel MGRIT runs without spatial coarsening (No SC). For explicit MGRIT runs, spatial coarsening is required for stability, and speedup with respect to sequential time-stepping will be demonstrated in section* 5*. Section* 5 *will also demonstrate speedup with respect to sequential time-stepping for implicit MGRIT runs.*

			Implicit			Explicit		
Problem	$N_x \times N_t$		$2^7 \times 2^7$	$2^9 \times 2^9$	$2^{11} \times 2^{11}$	$2^7 \times 2^8$	$2^9 \times 2^{10}$	$2^{11} \times 2^{12}$
A3	No SC	2-level	12	14	14 (8.2)	—	—	—
		F-cycle	12	16	20 (**56.1**)	—	—	—
	SC-2	2-level	64	83	85 (46.3)	100*	100*	100*
		F-cycle	64	85	87 (95.9)	100*	100*	100*
	SC-A	2-level	26	28	29 (16.5)	30	31	32 (17.3)
		F-cycle	27	28	30 (**37.5**)	32	35	37 (50.7)
A4	No SC	2-level	12	13	13 (15.1)	—	—	—
		F-cycle	12	15	18 (**60.9**)	—	—	—
	SC-A	2-level	16	17	22 (20.2)	21	27	30 (21.0)
		F-cycle	16	20	28 (**51.6**)	21	28	33 (67.1)
A5	No SC	2-level	13	12	13 (15.0)	—	—	—
		F-cycle	13	14	17 (**51.0**)	—	—	—
	SC-A	2-level	19	20	26 (23.0)	26	27	30 (19.0)
		F-cycle	20	22	28 (**49.7**)	27	28	31 (59.4)

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. For each of these cases, the wave speed vanishes at certain locations in the simulation domain, so we expect that uniform spatial coarsening will not be effective. 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 tol $= (2.5\times10^{-11})\sqrt{N_t N_x}$. Table 2 summarizes the results for MGRIT using implicit and explicit time integration.

For each of the cases A3, A4, and A5, explicit results with No SC are not shown since these simulations are unstable on coarse levels. For case A3, uniform spatial coarsening (SC-2) results in high iteration counts due to the presence of weak spatial connections. Adaptive spatial coarsening (SC-A) dramatically improves the iteration counts over SC-2, enabling efficient MGRIT runs with stable coarse propagation in the explicit case, and enabling cheaper MGRIT cycles in the implicit case, without increasing the iteration counts too much. SC-A is superior to SC-2 in the same way also for cases A4 and A5 (SC-2 results not shown to declutter the table). 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.

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 since they can be executed in parallel on all but the coarsest level, whereas in 2-level methods the second level needs to be executed sequentially and, thus, forms a sequential bottleneck. 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 for all of cases A3, A4, and A5. The run-times (only shown for the largest grid sizes to reduce clutter in the tables) show that implicit MGRIT with adaptive spatial coarsening has the potential to speed up F-cycles compared to No SC. For explicit MGRIT runs, spatial coarsening is required for stability, and parallel speedup with respect to sequential time-stepping will be demonstrated in section 5. Section 5 will also demonstrate parallel speedup with respect to sequential time-stepping for implicit MGRIT runs.

The SC-A iterations in Table 2 show a moderate increase as a function of problem size. The near scalability for both implicit and explicit results is 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 time-stepping. Such a result would yield significant savings for both implicit and explicit schemes.

TABLE 3

Results for implicit 2-level MGRIT without spatial coarsening for $N_x \times N_t = 2^{13} \times 2^{13}$ with varying temporal coarsening factors.

	Coarsening factor m			
	2	4	8	16
Iter	14	25	44	81
Time	346.4	343.6	412.3	588.1

In the parallel results to be presented in section 5, we will compare MGRIT F-cycles with the 2-level method. To obtain parallel speedup for a 2-level method, it is crucial that the coarse-level solve is much cheaper than the fine-level solve, because the fine-level solve is performed in parallel, while the coarse-level solve needs to be done sequentially and forms a sequential bottleneck. In the setting of our 2-level method, this means that we should try to increase the coarsening factor as much as possible to make the coarse solve cheaper, but the results in Table 3 indicate that, for

our hyperbolic problems, increasing the coarsening factor results in very fast growth of the number of 2-level iterations required for convergence. This is contrary to the case of parabolic PDEs, where the coarsening factor can be taken much larger without incurring substantially increased iteration counts [13]. The increasing iteration count as in Table 3 is a manifestation of the difficulties that have been observed in obtaining speedup for parallel-in-time methods for hyperbolic PDEs [19, 10, 28]. For multilevel MGRIT in parallel, on the contrary, small coarsening factors are not a problem, since all coarse solves (except for the coarsest level) are executed in parallel, and the coarse solves don't form a sequential bottleneck. This is a major advantage of MGRIT over 2-level methods: MGRIT can employ very gradual changes between successively coarsened grids (e.g., a coarsening factor of 2) because the solves on all levels are parallel, and this tends to result in much lower iteration counts than for 2-level methods with sufficiently large coarsening factors to reduce the sequential bottleneck. In this way, while a multilevel MGRIT iteration is more expensive than a 2-level iteration, multilevel MGRIT can be an efficient parallel method that combines low iteration counts with full parallel scalability, due to the absence of a level-2 sequential bottleneck. This will be demonstrated in the parallel results of section 5.

Guided by the results in Table 3, we will in our parallel 2-level results in section 5 use a coarsening factor of $m = 4$, which results in a cheaper coarse-grid solver than for $m = 2$, while still maintaining a relatively small number of iterations, resulting in the best run-time in Table 3. For MGRIT F-cycles we will retain a coarsening factor of $m = 2$, which gives the lowest MGRIT iteration count.

4.2. Burgers' equation. We solve Burgers' equation for case B1 (defined in section 3.2) 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 [24]

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

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 Table 4.

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 the right half of Table 4 correspond to using twice as many time steps as those in the

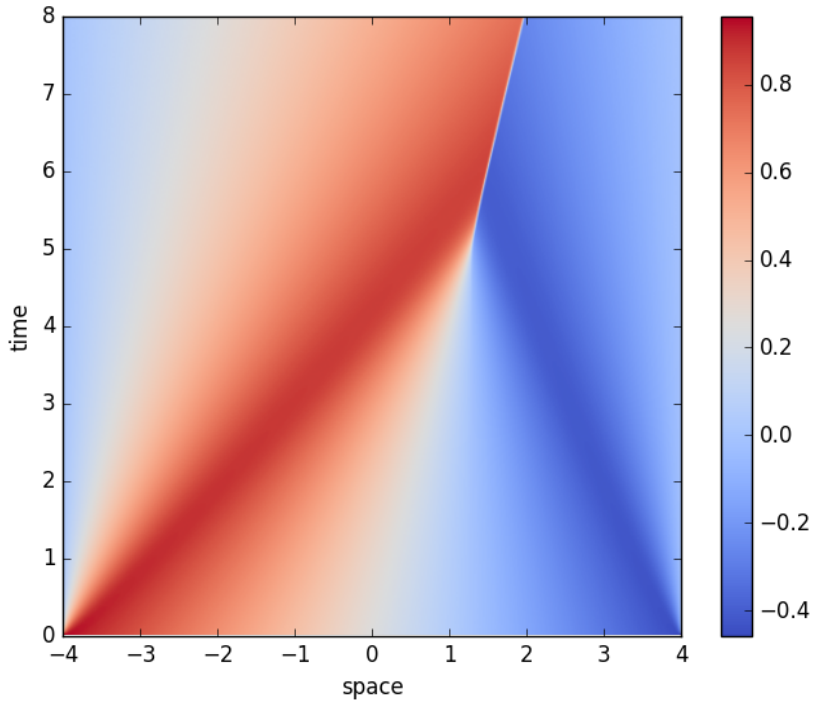

FIG. 8. *Burgers' equation, case* B1*: numerical solution on* $[-4,4]\times[0,8]$.

TABLE 4

Burgers' equation results (iteration counts, and run-times in seconds for the largest grid sizes). 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.

			Without shock: $t\in[0,4]$			With shock: $t\in[0,8]$		
$N_x\times N_t$			$2^7\times 2^7$	$2^9\times 2^9$	$2^{11}\times 2^{11}$	$2^7\times 2^7$	$2^9\times 2^9$	$2^{11}\times 2^{11}$
Implicit Max CFL ≈ 0.96	No SC	2-level	10	11	11 (296.2)	12	13	14 (444.0)
		F-cycle	11	12	16 (1496)	12	15	20 (2846)
	SC-G	2-level	25	28	29 (733.5)	26	28	29 (906.0)
		F-cycle	26	28	31 (1714)	26	28	30 (1961)
$N_x\times N_t$			$2^7\times 2^7$	$2^9\times 2^9$	$2^{11}\times 2^{11}$	$2^7\times 2^8$	$2^9\times 2^{10}$	$2^{11}\times 2^{12}$
Explicit Max CFL ≈ 0.48	SC-D	2-level	29	32	32 (19.4)	31	33	33 (36.9)
		F-cycle	30	36	42 (60.6)	32	37	49 (143.0)
$N_x\times N_t$			$2^7\times 2^8$	$2^9\times 2^{10}$	$2^{11}\times 2^{12}$	$2^7\times 2^9$	$2^9\times 2^{11}$	$2^{11}\times 2^{13}$
Explicit Max CFL ≈ 0.24	SC-D	2-level	19	21	22 (28.4)	20	22	22 (52.8)
		F-cycle	19	21	27 (97.3)	20	21	31 (217.5)

left half of Table 4 (to maintain the same 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 stability. 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.

The higher per-iteration cost observed for the case of implicit MGRIT F-cycles

without spatial coarsening over the longer time interval is due to a strong increase in the number of Newton iterations required for each time step once the shock has formed (from 2–4 preshock, to 5–20 postshock, with greater numbers required on coarser levels). On the other hand, no significant increase in Newton iterations is observed in the spatial coarsening case (3–5 iterations consistent across levels).

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, and hence are relegated to Supplementary Material sections SM3 and SM4. Results for explicit time integration are presented in section 5.1, followed by results for implicit time integration in section 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.

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 of providing fair, representative comparisons that 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.

TABLE 5
Best speedup achieved for explicit time-stepping strong scaling tests, $(N_x, N_t) = (2^n, 2^{n+1})$.

		(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

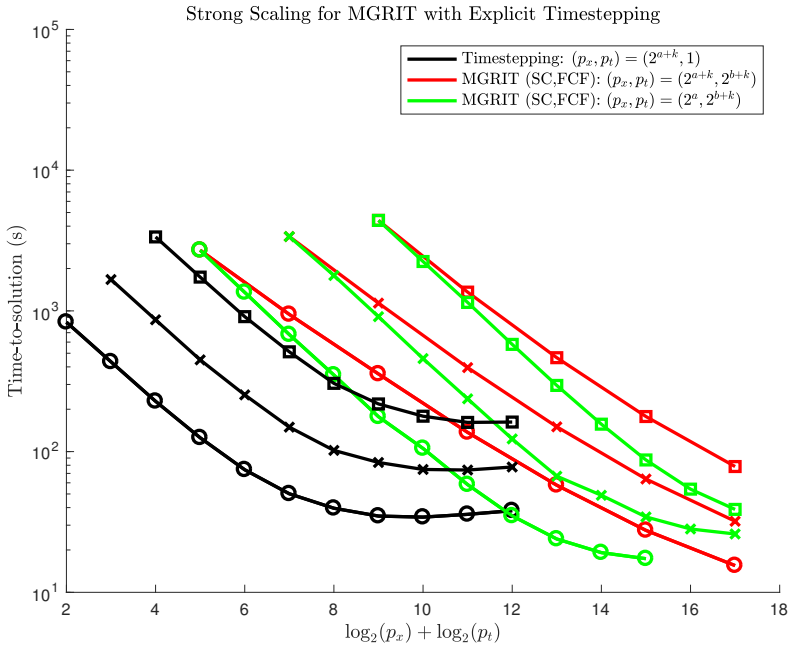

FIG. 9. *Comparison of serial time-stepping with spatial parallelism to MGRIT with FCF-relaxation and different combinations of space-time parallelism for three different problem sizes on up to* 131072 *cores. These results correspond to Tables* SM1–SM4 *in the Supplementary Materials.*
○*:* $(N_x, N_t) = (2^{14}, 2^{15})$*,* $a = 2$*,* $b = 3$*.*
×*:* $(N_x, N_t) = (2^{15}, 2^{16})$*,* $a = 3$*,* $b = 4$*.*
□*:* $(N_x, N_t) = (2^{16}, 2^{17})$*,* $a = 4$*,* $b = 5$*. (Figure in color online.)*

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 of Supplementary Results 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 speedup 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) is recorded in Table 5. Numerical values corresponding to these plots are recorded in Tables SM2–SM4 of Supplementary Results section SM1. 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 maintaining the original initial condition $u_0(x) = \sin(\pi x/2)$ with fixed frequency as spatial resolution increases, we also consider initial conditions $u_0(x) = \sin(2\pi\xi x)$, where ξ is chosen so that the frequency increases as the spatial grid is refined.

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 6) or double N_t while leaving N_x fixed (Table 7); 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-two 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 with frequency increasing as a function of grid resolution (Table 6). The higher-frequency tests in Table 7, for fixed resolution in space, show that convergence is not hampered by increasing the frequency.

TABLE 6
Weak scaling for explicit MGRIT F-cycles with $u_0(x) = \sin(2\pi\xi x)$ *and increasing* N_x *and* N_t.

					Original			Increased frequency		
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	170.08	1/4	31	169.28
2	11	12	1	2	1/4	33	220.90	1/2	33	220.62
3	12	13	2	3	1/4	34	232.30	1	34	231.88
4	13	14	3	4	1/4	36	291.86	2	36	291.64
5	14	15	4	5	1/4	37	334.94	4	37	334.82

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

					Original			Increased frequency		
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	170.02	1	31	169.41
2	10	12	0	2	1/4	14	100.81	1	14	100.57
3	10	13	0	3	1/4	11	89.37	1	11	89.29
4	10	14	0	4	1/4	9	79.96	1	9	79.92
5	10	15	0	5	1/4	7	69.59	1	7	69.63
6	10	16	0	6	1/4	6	65.25	1	6	65.25
7	10	17	0	7	1/4	5	62.02	1	5	61.98

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 section 5.1 for the explicit case, except that we start with $a = b = 2$.

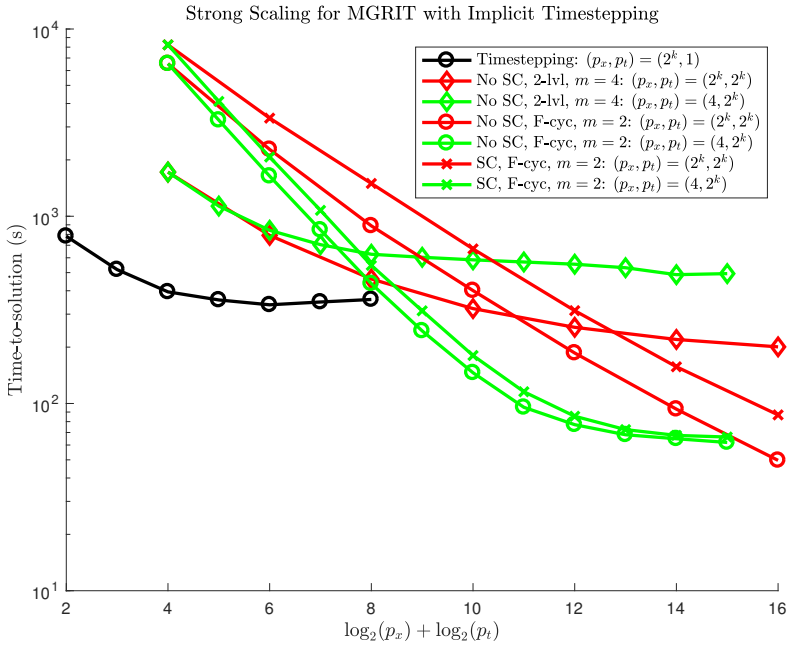

FIG. 10. *Comparison of serial time-stepping with spatial parallelism to MGRIT using FCF-relaxation with or without spatial coarsening for different combinations of space-time parallelism on up to* 65536 *cores. These results correspond to Tables* SM5–SM7 *in the supplementary materials. Two-level results with coarsening factor* 4 *are also shown for comparison.*

TABLE 8
Best speedup achieved for implicit time-stepping strong scaling tests, $(N_x, N_t) = (2^{14}, 2^{14})$.

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

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 of Supplementary Results SM2. 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 is recorded in Table 8.

Figure 10 also shows 2-level results with coarsening factor 4. As expected from the discussion in section 4.1, the 2-level parallel-in-time method scales substantially less favorably than the multilevel MGRIT method, because the coarse-level solve needs to be done sequentially and forms a sequential bottleneck. In contrast, the multilevel MGRIT algorithm is parallel on all coarse levels, and does not have the level-2 sequential bottleneck. Due to this bottleneck, the scalability of the 2-level method levels off much sooner than for the multilevel method, and only the version of the 2-level method that increases parallelism both in space and in time (diamond curve) manages to eke out a small amount of speedup compared to sequential time-stepping.

In the supplemental materials Figures SM1 and SM2, we give the parallel efficiency analogue of Figures 9 and 10, where parallel efficiency is measured as $T(1)/(P \cdot T(P)) \cdot 100$, where $T(P)$ is the wall-clock time required for solution on P processors. As determining $T(1)$ (for sequential time-stepping without spatial parallelism) is impractical for these large problem sizes, we approximate it by assuming near-perfect strong scaling for the case of spatial parallelism with small processor counts: we take $T(1) \approx T(P) \cdot 1.9^{\log_2(P)}$, where we use the smallest P for which we have time-stepping results. We observe similar efficiency cross-over points as in Figures 9 and 10, where, past a certain core count, multilevel MGRIT becomes more efficient than time-stepping and the 2-level method.

TABLE 9

Time to solution (s) (and parallel efficiency (%)) for explicit time-stepping strong scaling tests, $(N_x, N_t) = (2^{16}, 2^{17})$. Times with asterisks are constant extrapolations from the last known value (see also Figure SM1*). Row* 1 *corresponds to the black curve from Figure* 9 *(serial time-stepping), and rows* 2 *and* 3 *correspond to the red and green curves (MGRIT), respectively. The MGRIT efficiency numbers reflect that MGRIT uses highly redundant computations to achieve scalability on very large core counts.*

		$\log_2(p_x \cdot p_t)$				
		9	11	13	15	17
(p_x, p_t)	$(2^{k+4}, 1)$	218 (39.4)	161 (13.3)	162* (3.1)	162* (0.7)	162* (0.2)
	$(2^{k+4}, 2^{k+5})$	4395 (2.0)	1360 (1.6)	465 (1.2)	178 (0.8)	78 (0.4)
	$(2^4, 2^{k+5})$	4395 (2.0)	1148 (1.9)	295 (1.8)	87 (1.5)	39 (0.9)

Selected execution times from Figure 9 and efficiencies from Figure SM1 are highlighted in Table 9. Since methods like parareal and MGRIT achieve run-time speedup by employing computations that are highly redundant but are, at the same time, amenable to parallel execution in a scalable manner, the parallel efficiencies of these parallel-in-time methods are generally low, reflecting the number of redundant computations that are needed to obtain an algorithm that can scale well when large amounts of parallelism are used. Parareal and MGRIT really target strong-scaling run-time speedups by exploiting parallelism in time, and, as shown in Figures 9 and 10, and in the complementary supplementary figures, Figures SM1 and SM2, the cost for this speedup is the use of additional resources. Time-stepping methods saturate and cannot efficiently use these additional resources to further reduce run-times, but methods like parareal and MGRIT can, by employing redundant computations that make the methods more scalable in parallel.

The multilevel strong scaling results in Figure 10 are significant because they are the first to demonstrate parallel speedup over sequential time-stepping for the case

of MGRIT applied to implicit discretizations of hyperbolic problems (where spatial coarsening is not required for stability), and they are among the first to demonstrate speedup in the implicit hyperbolic case for any parallel-in-time method. While our implicit serial results in Table 2 indicated that adaptive spatial coarsening has the potential to further speed up implicit MGRIT for hyperbolic PDEs, we do not observe speedup resulting from spatial coarsening in the parallel results of Figure 10, compared with MGRIT without spatial coarsening. In Tables SM6 and SM7 of Supplementary Results SM2 we tabulate the results from Figure 10. 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 by improving aspects of our MGRIT algorithm with adaptive spatial coarsening for the implicit hyperbolic case, we could see significant improvements in the SC time to solution compared to No SC also in parallel. Reducing this increase in iteration counts is a topic of further research. Nevertheless, we demonstrate substantial speedups of up to 6 and more using the MGRIT approach for implicit hyperbolic problems.

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			Increased frequency		
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	244.22	1/4	21	244.49
2	11	11	1	1	1/4	25	587.80	1/2	25	587.05
3	12	12	2	2	1/4	29	850.81	1	29	849.76
4	13	13	3	3	1/4	37	1230.55	2	37	1230.05
5	14	14	4	4	1/4	46	1465.35	4	46	1465.28

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

					Original			Increased frequency		
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	244.22	1	21	244.38
2	10	11	0	1	1/4	25	313.50	1	25	312.96
3	10	12	0	2	1/4	13	209.91	1	13	209.72
4	10	13	0	3	1/4	10	177.37	1	10	177.20
5	10	14	0	4	1/4	9	166.02	1	9	165.93
6	10	15	0	5	1/4	7	138.55	1	7	138.44
7	10	16	0	6	1/4	6	125.33	1	6	125.29
8	10	17	0	7	1/4	5	112.91	1	5	112.92
9	10	18	0	8	1/4	4	101.24	1	4	101.25

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 initial condition $u_0(x) = \sin(2\pi\xi x)$ with frequency growing as spatial resolution increases, 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

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

					Original			Increased frequency		
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	243.96	1/4	21	243.80
2	11	10	1	0	1/4	22	444.45	1/2	22	444.17
3	12	10	2	0	1/4	24	585.94	1	24	586.29
4	13	10	3	0	1/4	24	630.92	2	24	630.87
5	14	10	4	0	1/4	25	703.53	4	25	703.59
6	15	10	5	0	1/4	25	758.26	8	27	813.10
7	16	10	6	0	1/4	25	812.73	16	30	958.41
8	17	10	7	0	1/4	22	727.93	32	26	914.58
9	18	10	8	0	1/4	22	784.93	64	24	928.37

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 time-stepping 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 $\Delta t : \Delta x$ (as in Table 10).

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 the best of our 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 over sequential time-stepping when spatial parallelism alone saturates. For explicit methods, this requires our adaptive spatial coarsening strategy to retain stability on coarse levels. For implicit methods, we also demonstrate, for the first time, speedup for the case of MGRIT applied to implicit discretizations of hyperbolic problems, where spatial coarsening is not required for stability. Weak scaling results show that the MGRIT scalability is robust for solutions with oscillation frequency increasing as a function of grid resolution.

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 two dimensions and three dimensions, and will also apply to higher-order methods, future plans for solving hyperbolic problems with MGRIT involve implementing adaptive spatial coarsening for problems in two or more spatial dimensions, higher-order methods, and systems of hyperbolic equations. Hyperbolic systems pose a challenge due to multiple wave speeds that may be nonlinearly coupled. One approach that

may be considered for linear hyperbolic systems is to apply the scalar approach to the characteristic variables, but the nonlinear case will require more elaborate techniques. It is also important to note that the diffusive nature of the first-order discretizations used in this paper is likely beneficial for the efficient convergence of our MGRIT methods. For this reason, extending the methods proposed in this paper to higher-order methods remains an important open research challenge.

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] P. Benedusi, D. Hupp, P. Arbenz, and R. Krause, *A parallel multigrid solver for time-periodic incompressible Navier–Stokes equations in* 3*D*, in Numerical Mathematics and Advanced Applications ENUMATH 2015, Lect. Notes Comput. Sci. Eng. 112, Springer, Cham, 2016, pp. 265–273.

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

[5] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719505.

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

[7] F. Chen, J. S. Hesthaven, and X. Zhu, *The use of reduced basis methods to accelerate and stabilize the parareal method*, in Reduced Order Methods for Modeling and Computational Reduction, Springer, Cham, 2014, pp. 187–214.

[8] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, *Algorithm* 887*: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate*, ACM Trans. Math. Software, 35 (2008), 22.

[9] J. Cortial and C. Farhat, *A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems*, Internat. J. Numer. Methods Engrg., 77 (2009), pp. 451–470.

[10] X. Dai and Y. Maday, *Stable parareal in time method for first-and second-order hyperbolic systems*, SIAM J. Sci. Comput., 35 (2013), pp. A52–A78, https://doi.org/10.1137/110861002.

[11] T. A. Davis, *Algorithm* 832: *UMFPACK V*4.3*—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199.

[12] M. Emmett and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Commun. Appl. Math. Comput. Sci., 7 (2012), pp. 105–132.

[13] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661, https://doi.org/10.1137/130944230.

[14] 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*, Tech. Rep. LLNLJRNL-663416, Lawrence Livermore National Laboratory, Livermore, CA, 2015.

[15] 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., 39 (2017), pp. S298–S322, https://doi.org/10.1137/16M1082330.

[16] C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators: Theory and feasibility studies for fluid, structure, and fluid–structure applications*, Internat. J. Numer. Methods Engrg., 58 (2003), pp. 1397–1434.

[17] M. Gander and M. Petcu, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, in Paris-Sud Working Group on Modelling and Scientific Computing 2007–2008, ESAIM Proc. 25, EDP Sciences, 2008, pp. 114–129.

[18] M. J. Gander, 50 *Years of Time Parallel Time Integration*, in Multiple Shooting and Time Domain Decomposition Methods, Springer, Cham, 2015, pp. 69–113.

[19] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578, https://doi.org/10.1137/05064607X.

[20] S. Güttel, *A Parallel Overlapping Time-Domain Decomposition Method for ODEs*, in Domain Decomposition Methods in Science and Engineering XX, Springer, Heidelberg, 2013, pp. 459–466.

[21] G. Horton, *The time-parallel multigrid method*, Comm. Appl. Numer. Methods, 8 (1992), pp. 585–595.

[22] W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer Ser. Comput. Math. 33, Springer-Verlag, Berlin, 2013.

[23] 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), IEEE, 2018, pp. 1–5.

[24] R. J. LeVeque, *Numerical Methods for Conservation Laws*, 2nd ed., Birkhäuser Verlag, Basel, 1992.

[25] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.

[26] A. S. Nielsen, G. Brunner, and J. S. Hesthaven, *Communication-aware Adaptive Parareal with Application to a Nonlinear Hyperbolic System of Partial Differential Equations*, Tech. Report, EPFL-ARTICLE-228189, Swiss Federal Institute of Technology, Lausanne, 2017.

[27] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.

[28] D. Ruprecht, *Wave Propagation Characteristics of Parareal*, preprint, https://arxiv.org/abs/1701.01359, 2017.

[29] D. Ruprecht and R. Krause, *Explicit parallel-in-time integration of a linear acoustic-advection system*, Comput. & Fluids, 59 (2012), pp. 72–83.

[30] B. Southworth, T. Manteuffel, S. McCormick, S. Munzenmaier, and J. Ruge, *Non-symmetric Reduction-Based Algebraic Multigrid for Upwind Discretizations*, preprint, https://arxiv.org/abs/1704.05001, 2017.

[31] J. Steiner, D. Ruprecht, R. Speck, and R. Krause, *Convergence of parareal for the Navier-Stokes equations depending on the Reynolds number*, in Numerical Mathematics and Advanced Applications—ENUMATH 2013, Springer, Cham, 2015, pp. 195–202.

[32] N. P. V. Dobrev, Tz. Kolev and J. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM J. Sci. Comput., 39 (2017), pp. S501–S527, https://doi.org/10.1137/16M1074096.

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