

RESEARCH ARTICLE

WILEY

Optimizing multigrid reduction-in-time and Parareal coarse-grid operators for linear advection

Hans De Sterck¹  | Robert D. Falgout² | Stephanie Friedhoff³  |
Oliver A. Krzysik⁴  | Scott P. MacLachlan⁵ 

¹Department of Applied Mathematics,
University of Waterloo, Waterloo, Ontario,
Canada

²Center for Applied Scientific Computing,
Lawrence Livermore National Laboratory,
Livermore, California, USA

³Department of Mathematics, Bergische
Universität Wuppertal, Wuppertal,
Germany

⁴School of Mathematics, Monash
University, Clayton, Victoria, Australia

⁵Department of Mathematics and
Statistics, Memorial University of
Newfoundland, St John's, Newfoundland
and Labrador, Canada

Correspondence

Oliver A. Krzysik, School of Mathematics,
Monash University, Clayton, Vic.,
Australia.

Email: oliver.krzysik@monash.edu

Funding information

International Mobility Programme
funding from the ARC Centre of
Excellence for Mathematical and
Statistical Frontiers; Australian
Government Research Training Program
Scholarship; NSERC, Grant/Award
Numbers: RGPIN-2014-06032,
RGPIN-2019-04155, RGPIN-2019-05692;
U.S. Department of Energy by Lawrence
Livermore National Laboratory,
Grant/Award Number:
DE-AC52-07NA27344

Abstract

Parallel-in-time methods, such as multigrid reduction-in-time (MGRIT) and Parareal, provide an attractive option for increasing concurrency when simulating time-dependent partial differential equations (PDEs) in modern high-performance computing environments. While these techniques have been very successful for parabolic equations, it has often been observed that their performance suffers dramatically when applied to advection-dominated problems or purely hyperbolic PDEs using standard rediscritization approaches on coarse grids. In this paper, we apply MGRIT or Parareal to the constant-coefficient linear advection equation, appealing to existing convergence theory to provide insight into the typically nonscalable or even divergent behavior of these solvers for this problem. To overcome these failings, we replace rediscritization on coarse grids with improved coarse-grid operators that are computed by applying optimization techniques to approximately minimize error estimates from the convergence theory. One of our main findings is that, in order to obtain fast convergence as for parabolic problems, coarse-grid operators should take into account the behavior of the hyperbolic problem by tracking the characteristic curves. Our approach is tested for schemes of various orders using explicit or implicit Runge–Kutta methods combined with upwind-finite-difference spatial discretizations. In all cases, we obtain scalable convergence in just a handful of iterations, with parallel tests also showing significant speed-ups over sequential time-stepping.

KEYWORDS

high-order, hyperbolic, MGRIT, multigrid, parallel-in-time, Parareal

1 | INTRODUCTION

As compute clock speeds stagnate and core counts of parallel machines increase dramatically, parallel-in-time methods provide an attractive option for increasing concurrency when simulating time-dependent partial differential equations (PDEs).^{1,2} Two well-known parallel time integration methods are Parareal³ and multigrid reduction-in-time (MGRIT),⁴

which can be considered as multigrid methods, although Parareal also has other interpretations.⁵ For a wide variety of diffusion-dominated problems, these algorithms can achieve a significant reduction in wall clock time over sequential time-stepping methods, given enough parallel resources.^{2-4,6-9} However, despite this success for diffusion-dominated problems, MGRIT and Parareal (along with most other parallel-in-time methods) tend to perform quite poorly on hyperbolic PDEs. Specifically, they typically exhibit extremely slow convergence or even divergence when applied to advection-dominated PDEs.^{5,10-21} Moreover, many of these examples demonstrate a clear deterioration in convergence as the amount of dissipation in the underlying PDE and/or its discretization is decreased.^{16,20,21} Note that both MGRIT and Parareal converge to the exact solution of the discrete problem after a finite number of iterations due to sequential propagation of the initial condition across the temporal domain by the relaxation scheme.⁴ As such, in this paper, we use the term “divergence” to describe solvers that converge to the solution only in a number of iterations close to that for which they would reach the exact solution.

To date, attempts to address this problem have been largely unsuccessful. Several so-called stabilized variants of Parareal have been developed which overcome the typically divergent behavior of standard Parareal, but are significantly more expensive, and so their practicability is limited.^{10,11,18} Semi-Lagrangian coarse-grid operators were considered by Schmitt et al.²⁰ for the viscous Burgers equation, inspiring some of the work in this paper. Unfortunately, though, their numerical tests showed that convergence deteriorates significantly in the zero-viscosity/hyperbolic limit.²⁰ Coarsening also in space was considered by Howse et al.¹⁵ when applying MGRIT to the linear advection and Burgers’ equations to provide both cheaper multigrid cycles and overcome stability issues arising from coarsening only in time. Parallel speed-up was demonstrated for linear advection only, but convergence was, ultimately, slow and not scalable, and the approach did not work when applied to higher than first-order discretizations. Nielsen et al.¹⁷ achieved modest speed-ups for the shallow water equations, in part, by reducing the order of the discretizations in time and space, rather than coarsening the mesh. Small to moderate speed-ups for linear hyperbolic PDEs were obtained with ParaExp,²² but this algorithm falls outside of the class of MGRIT-like algorithms that are the subject of this paper. In all of these works, speed-ups over sequential time-stepping for hyperbolic PDEs are typically quite small (on the order of two to six), with slow convergence of the iteration ultimately inhibiting faster runtimes due to increased parallelism. For comparison, a speed-up on the order of 20 times was achieved for a diffusion-dominated parabolic problem in Falgout et al.⁹

A number of theoretical convergence analyses have been developed for Parareal and MGRIT,^{5,12,13,19,23-27} which have helped to explain numerical convergence results, and will likely play an important role in the design of new solvers. Furthermore, some theoretical studies have identified potential roadblocks for fast parallel-in-time convergence of hyperbolic PDEs.⁵ Nevertheless, there does not yet exist a general understanding of why the parallel-in-time solution of advection-dominated problems seems to be so much more difficult than for their diffusion-dominated counterparts.

The aim of this paper is to demonstrate that, in fact, MGRIT and Parareal, with the right choice of coarse-grid operator, can efficiently integrate hyperbolic PDEs. To do so, we work in an idealized environment, considering the constant-coefficient linear advection problem in one spatial dimension subject to periodic spatial boundary conditions, for which we can appeal to existing sharp MGRIT convergence theory. Informed by convergence theory and the PDE, we develop heuristics that coarse-grid operators should satisfy and we formulate optimization problems based on these to find “near-optimal” coarse-grid operators. For example, one such heuristic that we have developed is that coarse-grid operators should track information along characteristics, similar to the semi-Lagrangian schemes considered by Schmitt et al.²⁰ However, our optimization-based coarse-grid operators lead to robust solvers in the hyperbolic limit, unlike their semi-Lagrangian coarse-grid operators.²⁰ We demonstrate that these coarse-grid operators lead to scalable convergence, in the sense that the computational work is almost linear asymptotically as a function of the problem size (i.e., the solvers converge in approximately a constant number of multigrid iterations asymptotically). Moreover, convergence is achieved in just a handful of iterations, for both implicit and explicit discretizations, resulting in significant speed-ups in parallel over sequential time-stepping, comparable to what has been achieved for parabolic PDEs. Notably, our results include the use of high-order accurate discretizations (up to fifth order), which is important because many results reported in the literature for hyperbolic PDEs have used diffusive, low-order discretizations that have likely aided the convergence of the given parallel-in-time method. Additionally, our approach works for large coarsening factors, and we employ fine-grid CFL numbers that reflect what would realistically be used with sequential time-stepping.

The optimization approaches presented in this paper rely on calculations that are feasible specifically for the case of one-dimensional, constant-coefficient linear advection. This precludes direct application of these approaches to more complicated hyperbolic PDEs. Crucially, though, they give us powerful tools to demonstrate that, for this canonical hyperbolic PDE, it is possible to obtain highly efficient MGRIT and Parareal solvers. At the same time, our main finding that

coarse-grid operators should track characteristics is general, and we believe it will prove relevant to the design of solvers for more complicated hyperbolic PDEs. Practical methods for selecting coarse-grid operators for linear advection and other hyperbolic problems that follow this main insight are the subject of further research.

The remainder of this paper is organized as follows. In Section 2, the model problem and its discretizations are introduced, a brief overview of MGRIT and Parareal is given, and some motivating numerical examples are presented. A discussion on convergence theory and what it reveals about the difficulty of hyperbolic problems is given in Section 3. Section 4 develops an optimization-based approach for finding effective coarse-grid time-stepping operators. Parallel results are given in Section 5 for some of the newly developed coarse-grid operators. Concluding remarks and a discussion of future work is the subject of Section 6.

2 | PRELIMINARIES

In this section, we outline the model problem, its discretizations, and give a brief summary of MGRIT and Parareal. We then demonstrate the difficulty our seemingly simple model problem poses for these algorithms via some numerical examples.

2.1 | Model problem and discretizations

For the model problem, we consider the one-dimensional linear advection equation,

$$\partial_t u + \alpha \partial_x u = 0, \quad (x, t) \in [-1, 1] \times (0, T], \quad u(x, 0) = \sin^4(\pi x), \quad (1)$$

with constant wave speed $\alpha > 0$. While the exact solution of this canonical hyperbolic PDE is just the shifted initial condition, and its numerical approximation is easily obtained in the sequential time-stepping setting, it presents enormous difficulty for parallel-in-time solvers. In what follows, as throughout most of this paper, we consider periodic boundary conditions in space, but briefly turn our attention to inflow/outflow boundaries in Section 4.5.

To numerically approximate the solution of (1), finite-difference spatial discretizations are used with Runge–Kutta time integrators. As such, the spatial domain $x \in [-1, 1]$ is discretized with $n_x + 1$ equidistant points with spacing Δx , and the temporal domain $t \in [0, T]$ is discretized with $n_t + 1$ equidistant points having a spacing of Δt . We employ the method of lines to generate a semi-discretized representation. First, a p th-order upwind finite-difference spatial discretization is applied to (1), resulting in the system of ordinary differential equations (ODEs)

$$\mathbf{u}'(t) = \mathcal{L}\mathbf{u}, \quad t \in (0, T], \quad \mathbf{u}(0) = \mathbf{u}(x, 0), \quad (2)$$

in which $\mathcal{L} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ represents the discretization of $-\alpha \partial_x$ on the spatial mesh. Since α is constant and periodic boundaries are applied, \mathcal{L} is a circulant matrix and is, thus, unitarily diagonalized by the discrete Fourier transform (DFT). Specifically, we use upwind-finite-difference spatial discretizations of orders 1–5, which we denote as U1–U5. Letting v'_i denote the derivative of the single variable function $v(x)$ at point x_i , these are given by

$$v'_i = \frac{1}{\Delta x} [v_i - v_{i-1}] + \mathcal{O}(\Delta x), \quad (U1)$$

$$v'_i = \frac{1}{2\Delta x} [3v_i - 4v_{i-1} + v_{i-2}] + \mathcal{O}(\Delta x^2), \quad (U2)$$

$$v'_i = \frac{1}{6\Delta x} [2v_{i+1} + 3v_i - 6v_{i-1} + v_{i-2}] + \mathcal{O}(\Delta x^3), \quad (U3)$$

$$v'_i = \frac{1}{12\Delta x} [3v_{i+1} + 10v_i - 18v_{i-1} + 6v_{i-2} - v_{i-3}] + \mathcal{O}(\Delta x^4), \quad (U4)$$

$$v'_i = \frac{1}{60\Delta x} [-3v_{i+2} + 30v_{i+1} + 20v_i - 60v_{i-1} + 15v_{i-2} - 2v_{i-3}] + \mathcal{O}(\Delta x^5). \quad (U5)$$

These discretizations may be constructed using standard techniques, as described by Shu,²⁸ for example.

The ODE system (2) is then discretized using either a p th-order explicit Runge–Kutta (ERK) method, or a p th-order, L-stable singly diagonally implicit Runge–Kutta (SDIRK) method, with the resulting scheme denoted as either ERK p +Up, or SDIRK p +Up. We consider ERK schemes of orders 1–5 as follows: The first-order scheme is Euler’s method; the second- and third-order methods are the “optimal” strong-stability-preserving schemes (Reference 29, (9.7), (9.8)) the fourth-order scheme is the “classical Runge–Kutta method”; (Reference 30, p. 180) and finally, see Butcher (Reference 30, (236a)) for the fifth-order scheme. We consider SDIRK schemes of orders 1–4 as follows: The first-order scheme is Euler’s method; the second- and third-order methods can be found in Butcher (Reference 30, pp. 261–262) and the fourth-order scheme is given by Hairer and Wanner (Reference 31, (6.16)). Butcher tableaux for these Runge–Kutta schemes can be found in Appendix A1.

Upon application of a Runge–Kutta scheme to ODEs (2), their numerical solution may be written in the one-step form

$$\mathbf{u}^{n+1} = \Phi \mathbf{u}^n, \quad \mathbf{u}^0 = \mathbf{u}(0), \quad n = 0, \dots, n_t. \quad (3)$$

Note that Equation (3) can be written as a single, large space-time block lower bidiagonal linear system. Here, $\Phi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ is known as the “time-stepping operator,” as it steps the discrete solution from one time level to the next. The eigenvalues of Φ can be computed as a function of those of \mathcal{L} .^{12,24} In fact, it can be shown that Φ is a rational function (in a matrix sense) of \mathcal{L} , as in Lemma 1.

Lemma 1 (Rational form of Φ). *Let $R(z) = P(z)/Q(z)$ denote the stability function (Reference 30, lemma 351A) of a Runge–Kutta scheme applied to (2), in which P and Q are polynomials derived from the Butcher tableau of the scheme. Then, for diagonalizable matrices $\mathcal{L} \in \mathbb{R}^{n_x \times n_x}$, the time-stepping operator in (3) is*

$$\Phi(\Delta t \mathcal{L}) = P(\Delta t \mathcal{L})[Q(\Delta t \mathcal{L})]^{-1}. \quad (4)$$

Proof. The proof essentially follows the same steps as analyzing the Runge–Kutta stability of ODE system (2) via decoupling it into a system of scalar ODEs, with the additional final step of recoupling the system. See, for example, Butcher (Reference 30, pp. 74–75) and Hairer and Wanner (Reference 31, pp. 15–16) for more details. ■

For an ERK scheme, $Q(z) = 1$ and so the Runge–Kutta stability function used in Lemma 1 is simply a polynomial, $R(z) = P(z)$.

Corollary 1. *For periodic boundary conditions applied to (1), the time-stepping operator Φ in (3) can be written as the product of a sparse circulant matrix and the inverse of a sparse circulant matrix. In the case of an ERK scheme, Φ is simply a sparse circulant matrix.*

Proof. For periodic boundaries, the finite-difference spatial discretizations \mathcal{L} are sparse and circulant, and noting that circulant matrices are closed under addition and multiplication, the result follows immediately from the rational form of Φ in (4). ■

The CFL number for Runge–Kutta finite-difference discretizations of (1) is defined as

$$c := \alpha \frac{\Delta t}{\Delta x}. \quad (5)$$

The explicit discretizations considered here suffer from a CFL limit, for which a necessary condition for numerical stability is $c \leq c_{\max}$. Values of c_{\max} can be computed from the Runge–Kutta stability function and the eigenvalues of $\Delta t \mathcal{L}$, and are given in Table 1. Throughout this paper, experiments using ERK discretizations will employ a CFL fraction—ratio of CFL number to CFL limit—of 85%, $c = 0.85c_{\max}$, since it is realistic of what would be used for regular time-stepping. In all SDIRK experiments, a CFL number of $c = 4$ is used. All of the SDIRK+U schemes considered here are unconditionally stable since the eigenvalues of the circulant matrices $\Delta t \mathcal{L} \in \mathbb{R}^{n_x \times n_x}$ lie in the left-half plane, which means they are contained in the stability region of any L-stable Runge–Kutta method. A wave speed of $\alpha = 1$ is used in all experiments.

To demonstrate the accuracy of the discretizations used here and to emphasize that the high-order methods faithfully represent the nondissipative nature of (1), computed discretization errors are shown in Figure 1. The high-order methods stand in stark contrast with the first-order SDIRK1+U1 (right-hand panel), which has yet to reach its asymptotic convergence rate of one, because it possesses significant numerical diffusivity.

TABLE 1 CFL limits c_{\max} for ERK+U discretizations of (1) with periodic boundary conditions

Scheme	ERK1+U1	ERK2+U2	ERK3+U3	ERK4+U4	ERK5+U5
c_{\max}	1	1/2	1.62589 ...	1.04449 ...	1.96583 ...

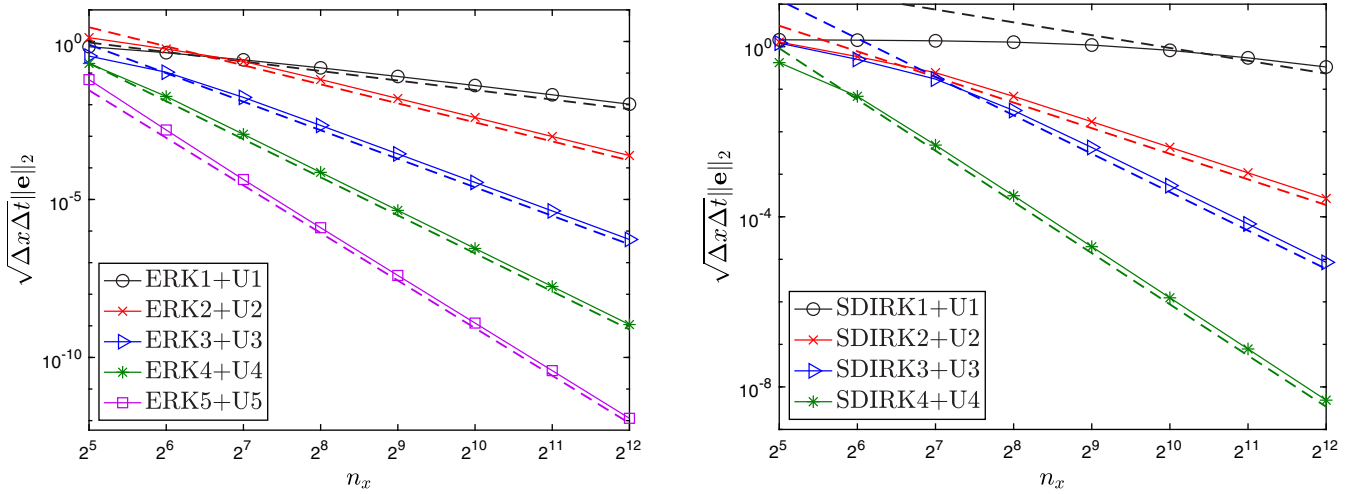


FIGURE 1 Space-time discretization errors for (1) measured in the discrete L_2 -norm. Left: ERK+U. Right: SDIRK+U. Plotted underneath each scheme's errors is a dashed line showing the theoretical convergence rate (order p for ERK p /SDIRK p +U p). Note the use of different scalings of the vertical axes in the plots

2.2 | MGRIT, Parareal, and numerical set-up

In this section, we give a brief overview of MGRIT and Parareal, and describe the set-up for our numerical tests. MGRIT and Parareal are parallel, multilevel, iterative methods for solving block lower triangular systems arising from evolutionary problems, such as (3); note, however, that Parareal is typically only thought of as a two-level method. In this paper, we make no significant differentiation between MGRIT and Parareal, since Parareal can be described in the MGRIT framework via certain choices of algorithmic parameters (as discussed below). As such, beyond the current section, we will focus on the MGRIT algorithm.

We now give an overview of the MGRIT algorithm as it applies to model problem (3); see Falgout et al.^{4,9} or Howse et al.¹⁵ for more detailed descriptions which apply to nonlinear problems, for example. The temporal mesh $(t^n)_{n=0}^{n_t} := (n\Delta t)_{n=0}^{n_t}$ on which the problem is posed is the “fine grid,” and an integer coarsening factor $m > 1$ is used to induce a “coarse grid” consisting of a subset of the fine-grid points, $(mn\Delta t)_{n=0}^{n_t/m}$. The set of points appearing exclusively on the fine grid are called “F-points,” while those appearing on both fine and coarse grids are “C-points.” An MGRIT iteration combines prereduction with a coarse-grid correction. The two fundamental types of relaxation are: F-relaxation, which is time-stepping starting from each C-point across the following F-interval; and C-relaxation, which is time-stepping from the last F-point in each interval to its following C-point. The standard relaxation sweeps performed in MGRIT are either: F-relaxation (commonly used in Parareal), or the stronger FCF-relaxation, which is an F-, followed by a C-, followed by an F-relaxation.

The coarse-grid problem is obtained by approximating the Schur complement system (with respect to C-points) of the fine-grid residual equation. This results in the algebraic error \mathbf{e}_c at C-points being approximated by the block lower-bidiagonal system

$$\mathbf{e}_c^{n+1} = \Psi \mathbf{e}_c^n + \mathbf{r}_c^{n+1}, \quad \mathbf{e}_c^0 = 0, \quad n = 0, \dots, n_t/m - 1, \quad (6)$$

in which \mathbf{r}_c^{n+1} is the fine-grid residual at the $n + 1$ st C-point, and $\Psi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ is the coarse-grid time-stepping operator, responsible for propagating the error from one C-point to the next. After solving (6), the coarse-grid error is interpolated to the fine grid via so-called ideal interpolation which corresponds to injection at C-points followed by an F-relaxation. To solve the coarse-grid problem (6), one can either do a sequential forward solve, or recursively apply the algorithm since

(6) has the same block lower-bidiagonal structure as fine-grid problem (3). A sequential coarse-grid solve is almost always used in Parareal algorithms, making them two-level solvers.

Taking $\Psi = \Psi_{\text{ideal}} := \Phi^m$ defines an *ideal* coarse-grid time-stepping operator in the sense that the exact solution of (3) is reached in a single MGRIT iteration. In this instance, coarse-grid problem (6) really is the Schur complement system of the fine-grid residual equation. However, no speed-up in parallel can be achieved with $\Psi = \Psi_{\text{ideal}}$ since the sequential coarse-grid solve is as expensive as the original problem of time-stepping with Φ across the entire fine grid. Instead, one should choose Ψ to be some approximation of Φ^m —or equivalently, it should approximate taking m steps with Φ on the fine grid—under the constraint that its action is significantly cheaper to compute so that speed-up can be achieved. Typically Ψ is chosen through the process of rediscrctizing Φ on the coarse grid, whereby it is employed with the enlarged coarse-grid time step, $m\Delta t$,^{4,12,23} but other techniques have also been considered.^{8,17}

For completeness, we now describe all of the settings used in our numerical tests. We exclusively use FCF-relaxation since we find that it generally leads to faster and more scalable solvers than using F-relaxation. The initial iterate for the space-time solution is uniformly random except at $t=0$, where it matches the prescribed initial condition. Unless otherwise noted, the metric used to report solver convergence is the number of iterations needed to achieve a space-time residual below 10^{-10} in the discrete L_2 -norm. This stopping criterion exceeds the accuracy of the underlying discretizations in almost all cases, and so its use typically leads to a dramatic “over solving” of the space-time system with respect to the discretization error. Nonetheless, we use such a small halting tolerance to highlight asymptotic convergence behavior. For all ERKp+Up tests, a spatial resolution is selected, and a number of points n_t in time is chosen to be the largest power of two such that $T = \Delta t n_t$ does not exceed 8 (note that requiring n_t to be a power of two simplifies the implementation because we always coarsen by a power of two). For $p = (1, 2, 3, 4, 5)$ and a CFL fraction of 85%, this results in final integration times $T \approx (6.8, 6.8, 5.5, 7.1, 6.7)$. For all SDIRK+U tests, $T=8$ and $n_t = n_x$ such that a CFL number (5) of $c=4$ results. Where scaling tests are presented, the mesh is refined uniformly in both space and time such that the CFL number of the fine-grid discretization remains constant.

2.3 | Failure of MGRIT with rediscrctization for the model problem

To provide a baseline for the numerical results shown later in this paper, we now present numerical results for model problem (1) using MGRIT with rediscrctized coarse-grid operators (i.e., Ψ is chosen as Φ with an enlarged time step of $m\Delta t$). For all ERK+U discretizations of (1), such a coarse-grid operator leads to divergent solvers for all m . This behavior is driven primarily by CFL instability: For coarsening factors $m > 1$, the coarse-grid CFL limit is violated (recalling fine-grid CFL numbers are set to 85% of their respective limits), and so the resulting (unstable) coarse-grid solution cannot accelerate convergence to the (stable) fine-grid solution. To overcome this instability, a possible strategy is to couple the explicit fine-level discretization with a stable, implicit coarse-grid discretization. In such cases, a large coarsening factor is required to amortize the increased cost of solving an implicit coarse-grid problem. However, in our numerical tests (not shown here), this technique seldom results in a good solver because the approximation it provides to $\Psi_{\text{ideal}} := \Phi^m$ is not good enough, even for small m ; furthermore, we are unaware of any results in the literature that have used this technique (either successfully or unsuccessfully) for hyperbolic problems. In the few instances where speed-up has been achieved for explicit discretizations of hyperbolic problems, alternative techniques, such as incorporating spatial coarsening,¹⁵ or reducing the order of the discretization in time¹⁷ have been used. While such techniques certainly avoid coarse-grid CFL instabilities, it is not clear that they result in efficient algorithms, since only small speed-ups have been observed in practice.^{15,17} Thus, we do not present numerical results for ERK discretizations here because the standard choice of rediscrctization is divergent for our time-only coarsening algorithm and, to the best of our knowledge, no other technique exists for developing efficient coarse-grid operators for explicit discretizations.

In contrast to explicit discretizations, unconditionally stable, implicit fine-grid discretizations can be rediscrctized on coarse grids to provide stable coarse-grid operators. Two-level MGRIT iteration counts for SDIRK+U discretizations of (1) using such coarse-grid operators are given in the left side of Table 2. All solvers, with the exception of SDIRK1+U1, are divergent since they converge in approximately the number of iterations for which the initial condition is sequentially propagated across the entire domain (if one uses exact arithmetic), as is done in sequential time-stepping. The relatively better performance of SDIRK1+U1 is attributable to the fact that it is highly diffusive (see Figure 1), but it still requires a number of iterations that is much higher than what we will achieve with the new approach introduced in this paper.

When using rediscrctization, MGRIT convergence rates for implicit discretizations of hyperbolic problems strongly depend on the CFL number, with smaller CFL numbers typically resulting in faster convergence, just as in the explicit

TABLE 2 Two-level iteration counts for SDIRK+U discretizations using a redscretized coarse-grid operator. Left: Measured iteration counts. Right: Iteration counts at which the exact solution is achieved using exact arithmetic, $n_t/(2m)$. The “ \times ” denotes a solve which suffered an overflow error at the 358th iteration where the residual norm was approximately 10^{303}

Scheme	$n_x \times n_t$	Iteration count		$n_t/(2m)$	
		$m = 2$	$m = 4$	$m = 2$	$m = 4$
SDIRK1+U1	$2^{10} \times 2^{10}$	18	38	256	128
	$2^{12} \times 2^{12}$	18	40	1024	512
SDIRK2+U2	$2^{10} \times 2^{10}$	241	128	256	128
	$2^{12} \times 2^{12}$	1008	514	1024	512
SDIRK3+U3	$2^{10} \times 2^{10}$	183	128	256	128
	$2^{12} \times 2^{12}$	891	507	1024	512
SDIRK4+U4	$2^{10} \times 2^{10}$	256	130	256	128
	$2^{12} \times 2^{12}$	\times	520	1024	512

case, even though there is no CFL limit to violate. This can be seen by contrasting the types of convergence rates reported in Dobrev et al.¹² for linear advection with those shown in Table 2 for larger, more realistic CFL numbers. This behavior stands in stark contrast with that for diffusion-dominated problems where convergence is typically achieved within 10 or so iterations, even for high-order discretizations and large coarsening factors.^{4,12}

3 | CONVERGENCE THEORY APPLIED TO HYPERBOLIC PROBLEMS

To better understand the reason for poor convergence of MGRIT applied to the model problem (as shown in the previous section), and hyperbolic PDEs more generally, we now recall the two-level MGRIT convergence theory of Dobrev et al.¹² and Hessenthaler et al.²⁴ and discuss some of its implications.

3.1 | Two-level convergence theory

The convergence behavior of MGRIT can be understood by analyzing its error propagation matrix, E . That is, if $\mathbf{e}^{(0)}$ is the initial space-time error, then after q MGRIT iterations, the error obeys $\|\mathbf{e}^{(q)}\| \leq \|E\|^q \|\mathbf{e}^{(0)}\|$. To analyze $\|E\|$, we assume that the fine-grid time stepper Φ and coarse-grid time stepper Ψ are simultaneously diagonalizable by a unitary transform, and denote their eigenvalues by $(\lambda_k)_{k=1}^{n_x}$, and $(\mu_k)_{k=1}^{n_x}$, respectively. These assumptions are satisfied by the Φ and Ψ considered here, since all circulant matrices are unitarily diagonalized by the DFT. Furthermore, the eigenvalues should satisfy $|\lambda_k|, |\mu_k| < 1 \forall k$, so that the time-stepping methods are stable; note $|\lambda_k| = |\mu_k| = 1$ is permissible, but the following result does not apply for such cases.

The assumption of diagonalizability allows error reduction for each spatial mode to be considered individually. That is, if E_k is the error propagator associated with the k th spatial mode, then we can consider $\|E_k\|$ for each k . Moreover, $\|E\|_2 = \max_k \|E_k\|_2$. For the case of FCF-relaxation considered in this paper, the 2-norm of E_k can be bounded as Reference 24, lemma 4.1, theorem 4.3 and Reference 12, theorem 3.3

$$\|E_k\|_2 \leq \sqrt{m} |\lambda_k|^m \frac{|\lambda_k^m - \mu_k|}{1 - |\mu_k|} (1 - |\mu_k|^{n_t/m-1}). \quad (7)$$

Southworth²⁵ showed that this bound is equal to $\|E_k\|_2$ up to $\mathcal{O}(m/n_t)$, and, so, it is sharp.

Given bound (7), the question is now: What is required of Ψ , by way of its eigenvalues $(\mu_k)_{k=1}^{n_x}$, for fast MGRIT convergence? Note that under the assumption $|\mu_k| < 1$, the last factor $1 - |\mu_k|^{n_t/m-1} \rightarrow 1$ as $n_t \rightarrow \infty$, meaning convergence is primarily determined by the preceding factors. Firstly, convergence of the k th mode is related to how closely $\mu_k \approx \lambda_k^m$. So, in general, the spectrum of Ψ should approximate that of Φ^m (but recall $\Psi = \Phi^m$ is not practically feasible). Secondly, from the denominator, error modes associated with $|\mu_k| \approx 1$ are potentially damped much slower than those having $|\mu_k| \ll 1$. This slow convergence must be rectified by ensuring the approximation $\mu_k \approx \lambda_k^m$ is more accurate for these modes. Thus, Ψ must most accurately approximate the largest (in magnitude) eigenvalues of Φ^m . As noted by Southworth,²⁵ the largest

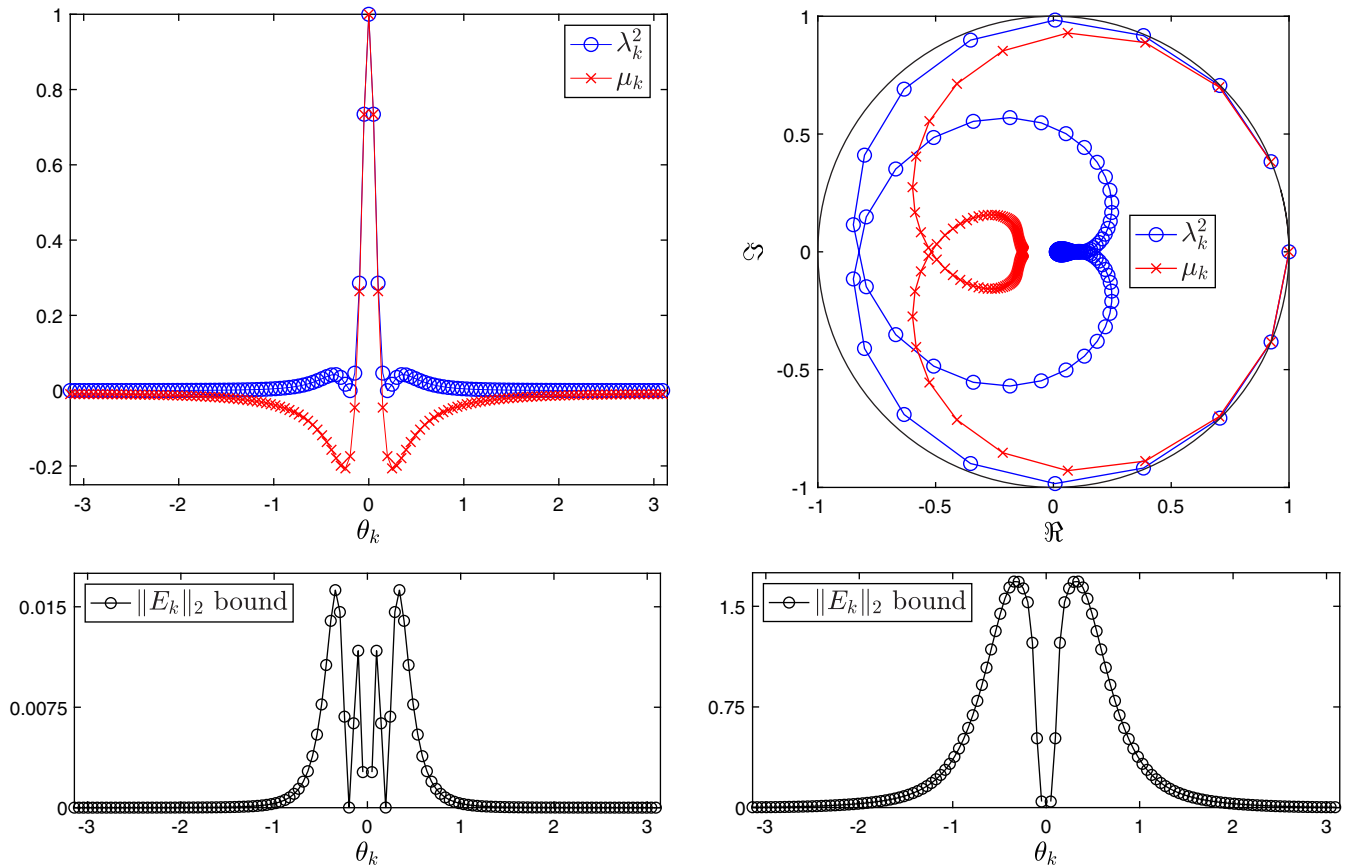


FIGURE 2 Left column: Diffusion equation $\partial_t u = \partial_{xx} u$ discretized with SDIRK2 in time and second-order central finite-differences in space. Right column: Advection equation $\partial_t u + \partial_x u = 0$ discretized with SDIRK3+U3. Top row: Eigenvalues λ_k^2 of Φ^2 , and μ_k of Ψ , with Ψ defined by rediscretizing Φ with $m = 2$. Bottom row: Error bound (7) for each problem as a function of Fourier frequency, θ_k . Both problems are subject to periodic boundary conditions in space, and are discretized on a space-time mesh covering $(x, t) \in (-1, 1) \times [0, 8]$ having $\Delta t = \Delta x = 1/64$, so that $\Phi, \Psi \in \mathbb{R}^{128 \times 128}$

(in magnitude) eigenvalues of Φ typically correspond to the smoothest spatial modes, and, so, equivalently, the action of Ψ must most accurately approximate that of Φ^m for spatially smooth modes. Moreover, the leading $|\lambda_k|^m$ factor provides an additional damping mechanism for modes having $|\lambda_k| \ll 1$; note this factor arises as a consequence of using FCF—rather than F-relaxation.^{12,24} In summary, fast convergence necessitates the approximation $\mu_k \approx \lambda_k^m$ to hold $\forall k$, and with increasing accuracy for $|\lambda_k|, |\mu_k| \rightarrow 1$.

3.2 | Implications of convergence theory

We now provide some insight as to why MGRIT convergence is typically much worse for advection-dominated problems compared with their diffusion-dominated counterparts. Discretizations of advection-dominated PDEs are (usually) much less dissipative than discretizations of diffusion-dominated PDEs since the PDEs themselves have little dissipation (or none in the hyperbolic limit). The amount of dissipation of the k th spatial mode for a given discretization Φ is directly related to the value of $|\lambda_k|$. Typically, for a discretization of a diffusion-dominated problem, there are very few $|\lambda_k| \approx 1$, and many $|\lambda_k| \ll 1$, while for an accurate discretization of an advection-dominated problem, there are many more $|\lambda_k| \approx 1$. In either case, $|\lambda_k| \approx 1$ typically correspond to smooth spatial modes. This behavior is seen in the top row of Figure 2, where the (square of the) eigenvalues of Φ for a purely diffusive and a purely advective PDE are shown.

Since diffusion-dominated problems have so few $|\lambda_k| \approx 1$, Ψ only has to accurately approximate very few eigenvalues of Φ^m to yield fast convergence. Conversely, since advection-dominated problems have many $|\lambda_k| \approx 1$, and very few $|\lambda_k| \ll 1$, Ψ has to accurately approximate a much greater proportion of the eigenvalues of Φ^m . In general, this makes

the task of identifying a good Ψ more difficult since it should have a simpler structure than Φ^m so that its action is less expensive to compute.

The properties just discussed, in conjunction with the plots in Figure 2, help to illuminate why rediscrretization of Φ with time step $m\Delta t$ typically leads to a good Ψ for diffusion-dominated problems, but is often a poor choice for advection-dominated problems. In Figure 2, notice the largest eigenvalues of Φ^2 are clustered around spatial frequency $\theta_k = 0$, noting that the $\theta_k = 0$ eigenvalue is at position (1,0) in the top right panel. In each instance, we see that μ_k provides a good approximation to λ_k^2 for the smoothest modes, $\theta_k \approx 0$. For the diffusion problem, this approximation is adequate to obtain fast convergence as the decay of $|\lambda_k|^2$ away from $\theta_k = 0$ is very rapid. However, for the advection problem, the approximation is inadequate since the decay of $|\lambda_k|^2$ away from $\theta_k = 0$ is more gradual and, so, the mismatch between λ_k^2 and μ_k is of much greater detriment for convergence. Error bounds (7) are shown in Figure 2 underneath the eigenvalue plots for each problem. In the diffusion case, the bound is very small and, so, convergence is fast while, in the advection case, the bound exceeds one, indicating that the solver will be divergent. It is the smooth modes that are not accurately captured by Ψ that cause the most issue.

In summary, rediscrretization essentially fails for advection-dominated problems since it does not provide an adequate approximation to Φ^m for smooth spatial modes, which tend to decay more slowly in time than for diffusion-dominated problems. Similar ideas were identified by Yavneh³² as being responsible for the breakdown of geometric multigrid for advection-dominated, steady-state PDEs. There, Fourier analysis showed that an inadequate coarse-grid correction of some asymptotically smooth error modes, so-called ‘characteristic components,’ is responsible for poor multigrid performance. It is conceivable that the problematic modes we identify here actually correspond to such space-time characteristic components, but lacking an analysis similar to that by Yavneh,³² this is difficult to say; this is an area of current research. Nonetheless, it is likely that some of the ideas proposed by Yavneh³² for improving spatial multigrid solvers will be useful for developing improved MGRIT solvers.

4 | COARSE-GRID OPERATORS BASED ON A LINEAR APPROXIMATION OF Ψ_{ideal}

From the discussion surrounding error estimates (7), the coarse-grid operator Ψ should approximately minimize the difference between its spectrum and that of Φ^m , in general, and particularly for larger $|\lambda_k|$. To this end, we consider Ψ as the solution of the minimization problem

$$\Psi := \arg \min_{\hat{\Psi} \in \mathbb{R}^{n_x \times n_x}} \left\| W_\lambda^{1/2} \left[\lambda^m - \mu(\hat{\Psi}) \right] \right\|_2^2, \quad (8)$$

where $\lambda = (\lambda_1, \dots, \lambda_{n_x})^\top$, $\mu = (\mu_1, \dots, \mu_{n_x})^\top \in \mathbb{C}^{n_x}$, and $(\lambda^m)_k \equiv \lambda_k^m$. Here, $W_\lambda := \text{diag}(\mathbf{w}) \in \mathbb{R}^{n_x \times n_x}$ is a weighting matrix, whose k th entry is $w_k := w(|\lambda_k|)$, in which $w : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a weighting function reflecting the heuristic that it is most important to minimize $\lambda_k^m - \mu_k$ for $|\lambda_k|, |\mu_k| \rightarrow 1$, and less important for $|\lambda_k|, |\mu_k| \ll 1$. One choice of weighting function that we have found to yield good results is

$$w(z) = \frac{1}{(1 - z + \epsilon)^2},$$

with $0 < \epsilon \ll 1$ a constant to avoid division by zero; we take $\epsilon = 10^{-6}$ in the numerical results shown here. Note that allowing a free choice for $\hat{\Psi}$ would naturally result in the choice $\Psi = \Phi^m$ and, so, the optimization in (8) is constrained by a pre-specified sparsity pattern of Ψ .

In general the eigenvalues of matrix Ψ depend nonlinearly on its entries and thus (8) constitutes a nonlinear minimization problem. However, for the special case of circulant Ψ , (8) reduces to a linear least squares problem because the eigenvalues of a circulant matrix depend only linearly on its entries (they are given by the DFT—a linear operator—applied to its first column). For explicit temporal discretizations of model problem (1) with periodic boundaries, it is reasonable to impose that Ψ is a sparse circulant matrix because Φ is and, hence, so too is Φ^m (Corollary 1). In Section 4.6, we will also show that a sparse Ψ may be used with implicit temporal discretizations of (1), for which Φ and Φ^m are dense matrices. Thus, for the remainder of this section, we focus exclusively on the case in which Ψ is a sparse circulant matrix such that (8) reduces to a linear least squares problem.

We have also formulated and solved a nonlinear least squares problem that is based on a more direct minimization of error estimates (7) than the heuristic-based (8). This more elaborate approach, however, gives similar results for the simple model problem considered here and so it has been omitted for brevity.

4.1 | Linear least squares formulation

Let $\tilde{\phi}^m, \tilde{\psi} \in \mathbb{R}^{n_x}$ denote the first columns of the circulant matrices Φ^m and Ψ , respectively, and note that a circulant matrix can be fully specified by its first column. Assuming the sparsity pattern of Ψ is given, we let $R \in \mathbb{R}^{\nu \times n_x}$ be the restriction operator that selects these ν nonzero entries from $\tilde{\psi}$, where $\nu \ll n_x$ since the column is sparse. Further details on the choice of R (or equivalently, the sparsity pattern of Ψ) will be given in the following sections. The unknowns are thus the nonzero components of $\tilde{\psi}$, which we denote as $\psi := R\tilde{\psi} \in \mathbb{R}^\nu$. Finally, let $F \in \mathbb{C}^{n_x \times n_x}$ be the DFT matrix, then by the properties of circulant matrices, $\lambda^m = F\tilde{\phi}^m$, and $\mu = F\tilde{\psi} = FR^\top \psi$ since $R^\top R$ has ones on the diagonal in rows where $\tilde{\psi}$ has nonzeros and zeros everywhere else. Thus, (8) can be written as a linear least squares problem for the nonzero entries in the first column of Ψ :

$$\psi := \arg \min_{\psi \in \mathbb{R}^\nu} \left\| W_\lambda^{1/2} F (\tilde{\phi}^m - R^\top \tilde{\psi}) \right\|_2^2. \quad (9)$$

Remark 1 (Minimizing $\|\Phi^m - \Psi\|_2^2$ is not sufficient to obtain a good solver). For weighting function $w=1$, or $W_\lambda = I$, (9) corresponds to minimizing the difference between the spectra of Φ^m and Ψ in the two-norm. This is equivalent to minimizing the difference between Φ^m and Ψ in the two-norm since they are both diagonalized by the same unitary transform. In this instance, the solution of (9) can be computed explicitly as $\psi = R\tilde{\phi}^m$, which means that Ψ is given by truncating Φ^m in the sparsity pattern of Ψ . We have found that this choice of Ψ typically does not lead to a fast or scalable solver for model problem (1) because it does not adequately capture the dominant eigenvalues of Φ^m (see Section 3).

Lemma 2. *The solution of (9) is real valued.*

Proof. The normal equations of (9) are

$$(RF^*W_\lambda FR^\top) \psi = (RF^*W_\lambda F) \tilde{\phi}^m.$$

Since R and $\tilde{\phi}^m$ are real, ψ is real if the circulant matrix $\mathcal{A} := F^*W_\lambda F = F^*\text{diag}(\mathbf{w})F$ is real. Letting $\tilde{\mathbf{a}}$ denote the 1st column of \mathcal{A} , then, because \mathcal{A} is circulant, $\tilde{\mathbf{a}} = F^*\mathbf{w}$; that is, $\tilde{\mathbf{a}}$ is the inverse DFT of \mathbf{w} . Appealing to properties of the inverse DFT, since \mathbf{w} is real, $\tilde{\mathbf{a}}$ will be real if \mathbf{w} has even symmetry, meaning that $w_k = w_{n_x-k}$. Using the explicit formula for the eigenvalues of circulant matrices, it is easy to verify that eigenvalues λ_k of any real-valued circulant matrix Φ must satisfy $|\lambda_k|^2 = \lambda_k \lambda_k^* = \lambda_k \lambda_{-k} = \lambda_k \lambda_{n_x-k} = |\lambda_{n_x-k}|^2$. It follows that \mathbf{w} is even since $w_k = w(|\lambda_k|) = w(|\lambda_{n_x-k}|) = w_{n_x-k}$ and, thus, \mathcal{A} is real. ■

In practice, the numerical solution of (9) is found to have some small imaginary components since F is complex and the problem is ill-conditioned. We simply truncate these components from the solution, as is justified by Lemma 2. In some cases, the imaginary components can become large and simply truncating them from the solution has never been found to result in a good solver; see Table 3. This is also observed for some other choices of the weight matrix W leading to particularly ill-conditioned matrices in (9). In practice, if an imaginary component larger than 10^{-8} is detected, we flag the results and do not accept the resulting Ψ as a coarse-grid operator. We note, however, that this does not happen for the sparsity patterns of Ψ that we advocate in the following sections.

4.2 | Explicit schemes: Selection of Ψ 's nonzero pattern

Before solving (9), we must first decide how to constrain the nonzero pattern of Ψ . Our goal is to develop coarse-grid operators Ψ that result in convergence in a small number of multigrid iterations, but that are sufficiently sparse to obtain a low cost per iteration. In multigrid, the cost per iteration is quantified by the so-called operator complexity, which is defined in the case of MGRIT as the total amount of work done in time-stepping on all levels, relative to the time-stepping

TABLE 3 Two-level iteration counts for ERK+U discretizations with Ψ as linear least squares solution (9). Left: Sparsity pattern of Ψ is equal to that of Φ . Right: Sparsity pattern of Ψ is based on that of Φ^m . An “ \times ” denotes a solve that did not converge to the required tolerance in significantly fewer than $n_t/(2m)$ iterations (i.e., the number of iterations at which the exact solution is reached). An “ \times^* ” denotes a solve in which the least squares solution had imaginary components larger than 10^{-8} , as another indication of divergence, and an indication that the least squares problem was severely ill-conditioned

Scheme	$n_x \times n_t$	$m(\Phi\text{-based sparsity})$						$m(\Phi^m\text{-based sparsity})$					
		2	4	8	16	32	64	2	4	8	16	32	64
ERK1+U1	$2^8 \times 2^{10}$	\times	\times	\times	\times	\times	\times	11	6	6	7	6	5
	$2^{10} \times 2^{12}$	\times	\times	\times	\times	\times	\times	11	6	6	7	6	5
	$2^{12} \times 2^{14}$	\times	\times	\times	\times	\times	\times	11	6	6	7	6	5
ERK2+U2	$2^8 \times 2^{11}$	10	\times	\times	\times	\times	\times	10	7	9	8	7	7
	$2^{10} \times 2^{13}$	10	\times	\times	\times	\times	\times	10	7	9	8	7	7
	$2^{12} \times 2^{15}$	10	\times	\times	\times	\times	\times	10	7	9	8	7	7
ERK3+U3	$2^8 \times 2^9$	9	\times	\times^*	\times^*	\times^*	\times^*	7	6	5	6	5	3
	$2^{10} \times 2^{11}$	9	\times	\times^*	\times^*	\times^*	\times^*	7	6	5	6	5	4
	$2^{12} \times 2^{13}$	9	\times	\times^*	\times^*	\times^*	\times^*	7	6	5	6	5	4
ERK4+U4	$2^8 \times 2^{10}$	6	4	8	\times	\times^*	\times^*	5	4	4	4	5	5
	$2^{10} \times 2^{12}$	6	4	8	\times	\times^*	\times^*	5	4	4	4	5	6
	$2^{12} \times 2^{14}$	6	4	8	\times	\times^*	\times^*	5	4	4	4	5	6
ERK5+U5	$2^8 \times 2^9$	3	3	7	\times^*	\times^*	\times^*	3	3	3	4	4	3
	$2^{10} \times 2^{11}$	3	3	7	\times^*	\times^*	\times^*	3	3	3	4	5	4
	$2^{12} \times 2^{13}$	3	3	7	\times^*	\times^*	\times^*	3	3	3	4	5	4

work on the finest level. The operator complexity depends on the sparsity of the coarse-grid operators Ψ . Clearly, we require Ψ to be significantly sparser than the exact coarse-grid operator Φ^m , so that time-stepping on the coarse grid is substantially less expensive than on the fine grid. Ideally, we would like the coarse operator Ψ not to be denser than the fine-level operator Φ , as would result from rediscretizing the PDE on the coarse grid (where we note that the exact coarse propagator Φ^m is much denser than Φ), but our numerical results will show that constraining Ψ to have as few nonzeros as Φ does not yield good solvers in general. Still, it is useful to consider the case where $\text{nnz}(\Psi) = \text{nnz}(\Phi)$, with $\text{nnz}(A)$ denoting the number of nonzeros of matrix A , as a reference case that we will compare the per-iteration cost of our operators Ψ with. To compute the operator complexity, let Φ_ℓ denote the time-stepping operator on level $1 \leq \ell \leq L$ of a multilevel hierarchy with $L > 1$ levels, meaning that $\Phi \equiv \Phi_1$ and $\Psi \equiv \Phi_2$ in the two-level notation we have been using so far. Now, assuming Φ_ℓ is a sparse operator, the work required to time-step with it is proportional to $\text{nnz}(\Phi_\ell)n_t m^{1-\ell}$, assuming a constant coarsening factor of m on all levels. Thus, the operator complexity is given by

$$\text{operator complexity} := \frac{1}{\text{nnz}(\Phi_1)} \sum_{\ell=1}^L m^{1-\ell} \text{nnz}(\Phi_\ell). \quad (10)$$

An efficient multigrid cycle should have an operator complexity that is bounded independently of L (so that the cost of the work on the coarse levels relative to the fine-level work is bounded by a constant independent of the problem size and the number of levels). In fact, if one uses ideal coarse-grid operators, $\Phi_{\ell+1} = \Phi_\ell^m \forall \ell$, the solver will have an operator complexity equal to L . In contrast, in our reference case where $\text{nnz}(\Psi) = \text{nnz}(\Phi)$, a two-level solver has an operator complexity bounded by $1 + 1/m$, and a multilevel solver obeying this condition on all levels has a complexity bound of $1 + 1/(m-1)$. In the following sections, we will compare measured operator complexities for the coarse operators Ψ we derive with these reference complexities.

Next, we discuss how to choose the locations of the nonzeros in Ψ . Note that simply rediscretizing Φ on a temporally coarsened mesh leads to Ψ having the same nonzero pattern as Φ . To motivate a better choice of sparsity pattern, we consider the effects of temporal coarsening on the exact solution of (1) when it is sampled on a space-time mesh; a schematic

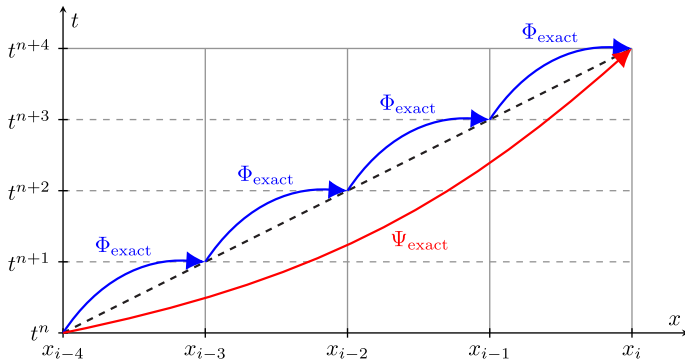


FIGURE 3 Exact fine- and coarse-grid time-stepping operators, Φ_{exact} and Ψ_{exact} , propagate the solution of $\partial_t u + \alpha \partial_x u = 0$ along one of its characteristics (thick, dashed line) on a fine and coarse grid, respectively. The fine grid has a temporal mesh spacing of $\Delta t = \Delta x / \alpha$, and the coarse grid of $4\Delta t$, with coarse-grid points at t^n and t^{n+4}

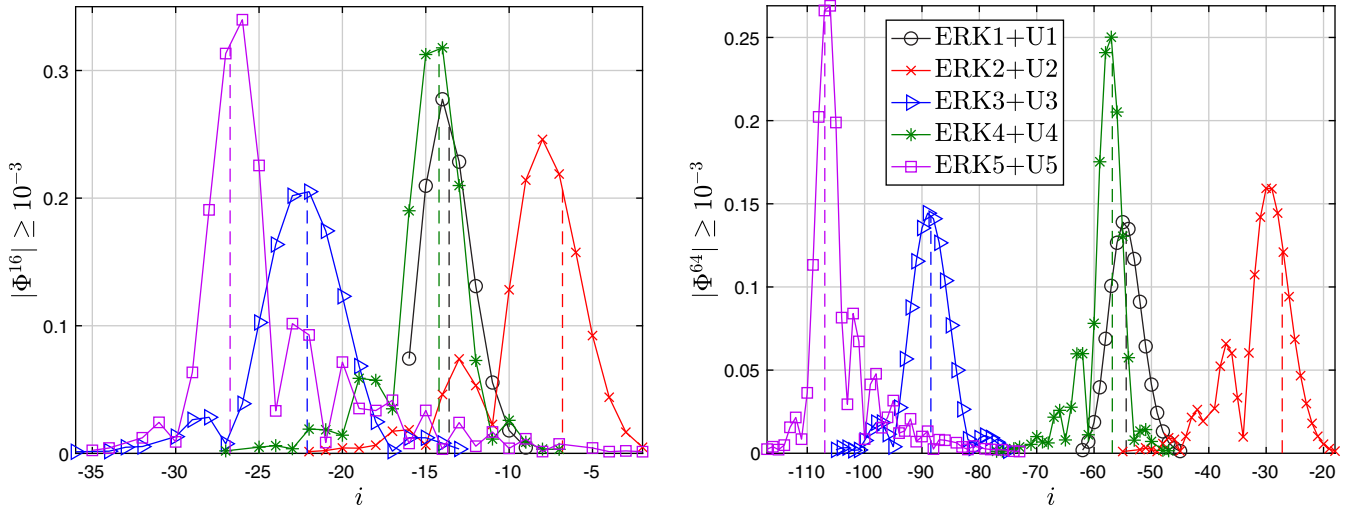


FIGURE 4 Magnitude of diagonal entries of $\Psi_{\text{ideal}} := \Phi^m$, $m = 16$ (left), and $m = 64$ (right), that are larger than 10^{-3} , as a function of their diagonal index, i . Fine-level discretizations, Φ , are $\text{ERK}p + \text{U}p$, $p \in \{1, 2, 3, 4, 5\}$. Dashed vertical lines for each discretization are included to represent a distance of $-mc\Delta x$ from $i = 0$. Note that $c = 0.85c_{\text{max}}$ is different for each scheme

diagram of this example is shown in Figure 3. The solution of a hyperbolic PDE is propagated through space-time along its characteristics, $x(t)$. Advection problem (1) simply has characteristics that are straight lines with slope $dx/dt = \alpha$. Now, say we have an exact fine-grid time-stepping operator, Φ_{exact} , that advects the PDE solution along characteristics from one time level to the next. From the diagram, it is clear that Φ_{exact} propagates the solution not only a distance of Δt in time, but also a distance of Δx in space. Considering semi-coarsening in time, by a factor of $m = 4$, for example, the resulting exact coarse-grid time-stepping operator is $\Psi_{\text{exact}} = \Phi_{\text{exact}}^4$. By definition, Ψ_{exact} propagates the solution forward in time by a distance of $4\Delta t$; however, we see that it also propagates the solution a distance of $4\Delta x$ in space. Thus, coarsening in the time direction, but not in space, has shifted the spatial stencil of Ψ_{exact} (which reaches back four points in space) with respect to that of Φ_{exact} (which reaches back one point in space).

From an algebraic perspective, it seems reasonable to consider $\Psi \approx \Phi^m$ having its sparsity pattern based on the largest nonzeros of Φ^m . To assess this, we compute Φ^m and examine its nonzeros as a function of their diagonal index i (recall entries of Φ^m are constant along its diagonals since it is circulant). We define diagonal index i to be 0 on the main diagonal, negative below the main diagonal, and positive above the main diagonal. For $m \in \{16, 64\}$, these are shown in Figure 4. There is clearly a well-defined distribution in the magnitude of these nonzeros for each scheme. The distributions peak at different i essentially because the time step is chosen differently for each scheme, since $c = 0.85c_{\text{max}}$ and c_{max} is different for each scheme (see Table 1). In the plots, dashed lines represent $mc\Delta x$, which is the spatial distance travelled along a characteristic departing from t^n and arriving at t^{n+m} . This illustrates, not unexpectedly, that the discretizations provide an approximation to the advection of the solution along characteristics that occurs at the PDE level.

From our previous intuitive arguments involving Figure 3, it is clear that the nonzero pattern of Ψ should reflect the characteristic nature of the PDE; this is also the conclusion reached from an algebraic perspective of approximating

Φ^m (Figure 4). We note that Gander¹³ also argued that making use of characteristic information may be important. The specific sparsity patterns used for the ERK+U schemes will be discussed further in Section 4.3.

4.3 | Explicit schemes: Two-level results

In this section, we show MGRIT convergence with Ψ as the solution of least squares problem (9) for ERK discretizations. To demonstrate the validity of the ideas outlined in the previous section, we solve the least squares problem for Ψ having a sparsity pattern equal to that of the fine-level operator Φ , and for it having a sparsity pattern based on the largest nonzeros of Φ^m . We will first compare numerical results for these two approaches, and will then provide details on how exactly we choose a sparsity pattern based on the largest nonzeros of Φ^m .

Solver iteration counts for Ψ having the same sparsity pattern as Φ are shown in the left side of Table 3. A convergent solver was not obtained for any m for the first-order scheme, convergent schemes were obtained only for $m = 2$ for the second- and third-order schemes, and convergent schemes with $m \in \{2, 4, 8\}$ were obtained for the fourth- and fifth-order schemes. In all cases where convergent solvers were found, the iteration counts remain constant as the mesh is refined. For the cases where the solvers converge, these results are certainly an improvement on those using rediscritization, which are divergent in this setting due to coarse-level CFL instability (see Section 2.3), attesting to the power of the optimization approach. However, for many coarsening factors and discretizations, the results are significantly worse than those obtained when using a sparsity pattern based on Φ^m , as shown in the right side of Table 3. When considering the magnitude of entries in Φ and Φ^m for $m = 2, 4, 8$ (plots not shown here for brevity), the locations of nonzeros in Φ correspond primarily to the largest nonzeros in Φ^m for the cases in which the Φ -based sparsity patterns yield convergent solvers. Furthermore, least squares problem (9) was severely ill-conditioned for many instances in which Ψ and Φ shared a sparsity pattern (see Table 3), but never when Ψ and Φ^m shared a similar sparsity pattern, further supporting our argument that using a characteristic-based sparsity pattern is the better choice.

We now explain in detail how the sparsity patterns were chosen that lead to the results in the right-hand side of Table 3, and then give a general discussion about the solvers. To select this sparsity pattern for a given discretization and coarsening factor, we first look at the locations of the largest nonzeros in Ψ_{ideal} (as in Figure 4, for example). As a first approximation, we choose a contiguous subset of the locations of the largest $\text{nnz}(\Phi)$ nonzeros of Φ^m (even if the locations of the largest nonzeros are not contiguous). The solver is then tested at multiple grid resolutions to determine if it is scalable. If the solver is not scalable, then an extra nonzero is included in a contiguous fashion and it is retested; this process is repeated until a scalable solver is obtained.

Additionally, once a scalable solver has been found, if it is determined that the convergence is significantly improved by including a relatively small number of additional nonzeros (e.g., two or three), then that is done also. Note, however, that the convergence rate has not been rigorously optimized as a function of the number of nonzeros. As an example, the left panel of Figure 5 shows the nonzero patterns of Ψ selected for ERK3+U3 as a function of coarsening factor, m . Also plotted is the coarse-grid characteristic departure point of $-mc$, demonstrating how the sparsity patterns are correlated with the departure points. Figure 5 also shows (right panel), for each discretization, the operator complexities (10) of the resulting solvers along with the operator complexity of $1 + 1/m$ that results when $\text{nnz}(\Psi) = \text{nnz}(\Phi)$ in a two-level method (see Section 4.2).

We find that, in general, to obtain convergent and scalable solvers there has to be a slight increase in the number of nonzeros in Ψ as the coarsening factor is increased, as can be seen for ERK3+U3 in Figure 5 (left panel), for example. This behavior appears consistent with that seen in Yavneh³² for the multigrid solution of steady-state advection-dominated PDEs, in which it was shown that coarse-grid operators may require a wider stencil than fine-grid operators. The number of additional nonzeros required is smaller for higher-order discretizations, as seen in the right panel of Figure 5, where operator complexities tend to be smaller for higher-order methods. Notice that ERK4+U4 and ERK5+U5 essentially have operator complexities of $1 + 1/m$ since very few (if any) additional nonzeros were needed.

The results at the right of Table 3 show that it is possible to overcome the CFL instability that arises from rediscritizing the fine-grid discretization on a temporally coarsened mesh and to obtain very fast two-level convergence and, therefore, show the significance of using a characteristic-based sparsity pattern for Ψ . Notably, the convergence rates of the solvers shown in Table 3 are comparable to those for model diffusion problems using rediscritized coarse-grid operators.^{4,12} To the best of our knowledge, these are the first scalable results obtained with a two-level time-coarsening algorithm for the explicit discretization of any hyperbolic PDE using realistic CFL numbers, and also for moderately large coarsening factors. Interestingly, convergence rates tend to be faster for higher-order discretizations compared with those of lower

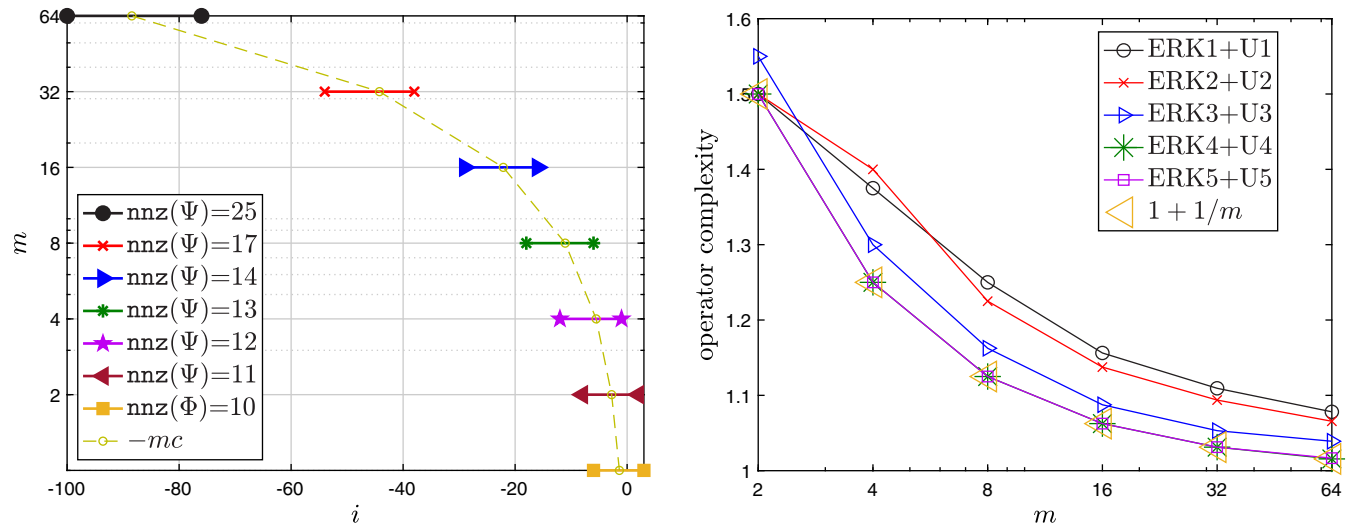


FIGURE 5 Two-level solvers for $m \in \{2, 4, 8, 16, 32, 64\}$ with Ψ as linear least squares solution (9). Left: Sparsity patterns chosen for Ψ for ERK3+U3, as represented by the nonzero diagonal indices i for each value of m . Also plotted is $-mc$, which represents the characteristic departure point on a grid using a time-step of $m\Delta t$. Right: Operator complexities (10) for all discretizations; shown also is the reference operator complexity of $1 + 1/m$

order. When combined with the trend in Figure 5 (right panel) that operator complexities are smaller for higher-order discretizations, this suggests that higher-order discretizations of model problem (1) likely benefit more from parallel-in-time integration.

Finally, an example of the eigenvalues and entries of Ψ for ERK3+U3 with $m = 8$ is shown in Figure 6. In this example, the eigenvalues of Φ^8 are clearly very well approximated by the eigenvalues of Ψ when they are of order one (in magnitude), and not so well approximated when they are smaller. Given this behavior, it is unsurprising that the solver converges quickly, and that the associated error bounds are small (bottom right panel of Figure 6). The entries of the least squares Ψ (upper right panel of Figure 6) are clearly correlated with those of the ideal operator.

Remark 2 (Relation between Ψ and semi-Lagrangian discretizations). It is clear that the optimized coarse operators Ψ with Φ^m -based sparsity have a nonlocal stencil structure that has a very different support than standard Eulerian discretizations like the ERK+U schemes. However, the stencil support of these coarse operators Ψ is similar to the stencil support of semi-Lagrangian schemes, since it tracks the characteristic curves of the PDE. This begs the question whether the optimized operators Ψ we obtain with Φ^m -based sparsity imposed may be close to semi-Lagrangian discretizations. It would indeed be useful for the sake of developing more practical approaches if our optimized operators Ψ could be replaced by semi-Lagrangian discretizations, without substantially degrading the convergence speed. However, since the stencil of Ψ has to be increased with coarsening factor m to get a scalable solver (see Figure 5), Ψ clearly does not just represent a particular semi-Lagrangian discretization of the PDE on coarse grids. This suggests that using standard semi-Lagrangian coarse operators may by itself not be sufficient to obtain scalable solvers, which is consistent with the findings in Schmitt et al.,²⁰ where fast convergence was not obtained for a hyperbolic problem using a semi-Lagrangian coarse-grid operator. Nonetheless, exploring semi-Lagrangian coarse-grid operators presents a promising area for future research.

4.4 | Explicit schemes: Multilevel results

In this section, we extend the results from above to define an effective hierarchy of coarse-grid operators for multilevel solvers. The scalability of the two-level solvers considered in the previous section is limited because they require the sequential solve of a large coarse-grid problem. Conversely, multilevel solvers are more scalable because the temporal grid can be coarsened gradually over many levels until the coarsest level contains sufficiently few degrees of freedom that a sequential solve there does not present a significant bottleneck.

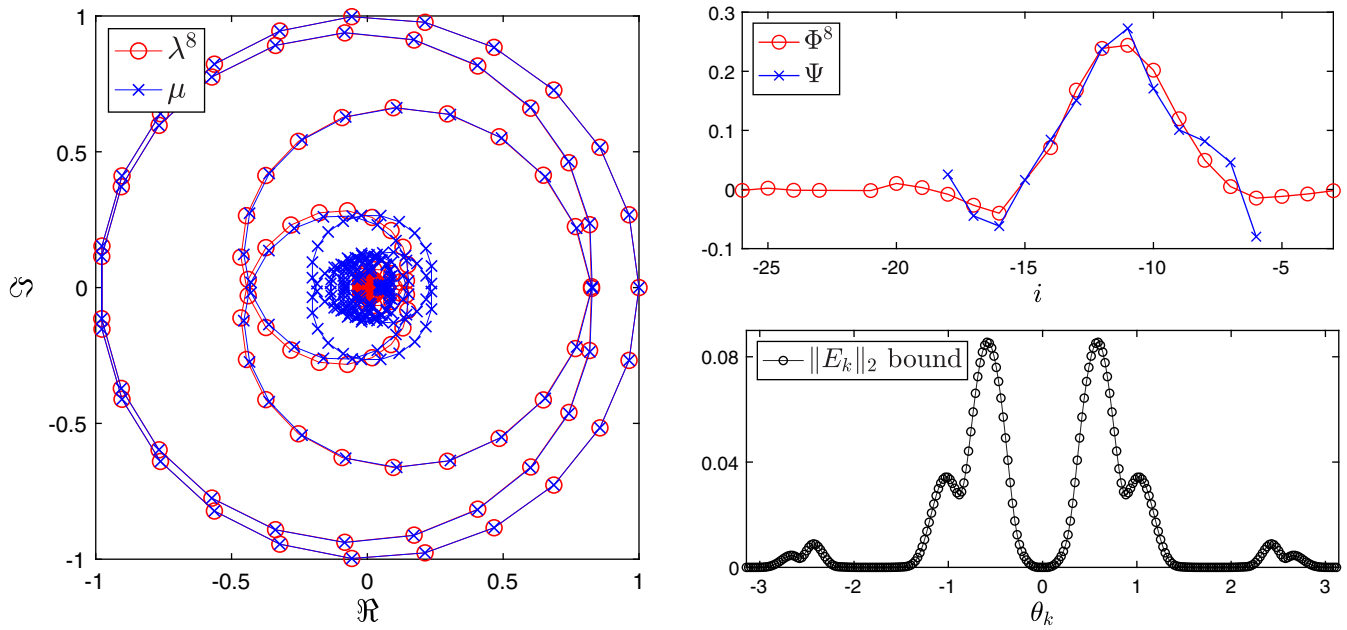


FIGURE 6 Linear least squares solution (9) for ERK3+U3 with $n_x \times n_t = 2^8 \times 2^9$, coarsening factor $m = 8$, and the sparsity pattern of the least-squares determined Ψ based on that of $\Psi_{\text{ideal}} = \Phi^8$. Left: Eigenvalues λ^8 of Φ^8 , and μ of Ψ . Top right: Entries of Φ^8 with magnitude larger than 10^{-3} as a function of their diagonal index, i , and all entries of Ψ . Bottom right: Error bound (7) as a function of spatial Fourier frequency

Convergence theory of multilevel MGRIT is significantly more complicated than in the two-level setting²⁴ and, so, rather than approximately minimizing a multilevel convergence estimate akin to what we did in the two-level case, we simply consider applying our previous two-level strategy in a recursive fashion. That is, if level ℓ uses a time-stepping operator Φ_ℓ , and coarsens by a factor of m , then the ideal time-stepping operator on level $\ell + 1$, $\Psi_{\text{ideal}, \ell+1} := \Phi_\ell^m$, is approximated with linear least squares problem (9). As previously, sparsity patterns of coarse-grid operators are selected by roughly choosing some subset of the locations of the largest nonzeros in the corresponding ideal coarse-grid operator. Again, we try to strike some balance between the overall convergence rate of the solver and the amount of fill-in of the coarse-grid operators.

For the sake of brevity, we only show results for ERK p +Up, $p \in \{1, 3, 5\}$. We have considered both V- and F-cycles using coarsening factors of both $m = 2$ and $m = 4$. However, only results for V-cycles using $m = 4$ coarsening are shown here because we found that this combination typically resulted in the fastest parallel solvers (see Section 5). For the case of ERK1+U1, we coarsen down to a minimum of just four points on the coarsest grid in time, and for ERK3+U3 and ERK5+U5, we coarsen down to a minimum of eight points.

The iteration counts for the resulting solvers are shown in Table 4 as a function of mesh resolution and number of grid levels. For all three discretizations, the solvers appear scalable with respect to the number of levels in the grid hierarchy and the mesh size, and they are very fast. We find that to obtain scalable solvers, the number of nonzeros in coarse-grid operators has to increase relative to that of the operator on the previous level. Similarly to the two-level case (see Figure 5), the amount of fill-in required decreases with increasing discretization order, as seen by the operator complexities also shown in the table. Importantly, the operator complexities converge to a constant as the number of levels is increased, which, when taken with the scalable iteration counts, indicates that the amount of work to solve a given problem is independent of the number of grid levels.

To the best of our knowledge, this is the first time that scalable multilevel results have been obtained for the explicit discretization of any hyperbolic PDE using a realistic CFL fraction. For example, Howse et al.¹⁵ is the only other work to show multilevel MGRIT results (with spatial coarsening) for explicit discretizations of hyperbolic problems, yet results presented there were limited to first-order accuracy, used a smaller CFL fraction, and even with the use of F-cycles, were not scalable with respect to mesh size. Furthermore, convergence was slow, with on the order of 40 iterations required to reach convergence for the mesh sizes considered here.

Scheme	$n_x \times n_t$	Number of levels					
		2	3	4	5	6	7
ERK1+U1	$2^8 \times 2^{10}$	6	6	6	6	–	–
	$2^{10} \times 2^{12}$	6	7	7	7	7	–
	$2^{12} \times 2^{14}$	6	7	7	7	7	7
	OC	1.38	1.56	1.65	1.69	1.71	1.71
ERK3+U3	$2^8 \times 2^9$	6	7	7	–	–	–
	$2^{10} \times 2^{11}$	6	7	7	7	–	–
	$2^{12} \times 2^{13}$	6	7	7	8	8	–
	OC	1.28	1.35	1.38	1.38	1.39	–
ERK5+U5	$2^8 \times 2^9$	3	4	4	–	–	–
	$2^{10} \times 2^{11}$	3	4	5	5	–	–
	$2^{12} \times 2^{13}$	3	4	5	5	5	–
	OC	1.25	1.31	1.33	1.33	1.33	–

TABLE 4 Multilevel iteration counts as a function of number of grid levels for V-cycles with $m = 4$ coarsening on each level. Operator complexities (OC) (10) are also given for each discretization. Note the reference operator complexity for a multilevel method is bounded above by $1 + 1/(m - 1) = 1.33 \dots$ for $m = 4$ (see Section 4.2). A “–” denotes a hierarchy that would have coarsened to fewer than the prescribed minimum number of allowable points (four for ERK1+U1, and eight for ERK3+U3 and ERK5+U5)

4.5 | Explicit schemes: Application to inflow/outflow boundaries

Discretizations of (1) subject to inflow/outflow spatial boundary conditions result in noncirculant Φ . Therefore, the optimization techniques discussed in previous sections cannot be rigorously applied since they rely on Φ and Ψ being circulant. Moreover, such Φ are nonnormal and likely not even diagonalizable, and, so, the rigorous optimization of the corresponding Ψ would require more sophisticated convergence theory than in both Dobrev et al.¹² and Hessenthaler et al.,²⁴ such as that of Southworth,²⁵ for example, and would certainly be highly nonlinear.

In the spirit of local Fourier analysis of multigrid methods,³³ we hypothesize that the Ψ we have previously designed for the periodic problem may work well for the inflow/outflow problem since the fine-grid operators Φ share the same Toeplitz structure away from the boundaries. To test this, we apply MGRIT to inflow/outflow problems using coarse-grid operators as described in the previous sections that were designed for analogous periodic problems (i.e., same discretizations, CFL numbers, n_x , and m), and we truncate these at the boundaries such that they are no longer circulant but remain Toeplitz. Note that not truncating the operators leads to similar, but slightly less satisfactory results. Given their nonzero pattern, truncating these operators at boundaries results in strictly lower triangular matrices in almost all cases.

In our tests, an inflow boundary condition $u(-1, t) = \sin^4(\pi t)$ is prescribed at $x = -1$. At the outflow boundary $x = 1$, no boundary condition is specified since the solution simply propagates out of the domain along characteristics. To implement the outflow numerically, however, we extrapolate from inside the computational domain to ghost points since solution data at ghost points is required by the third-order and higher spatial discretizations. To preserve the global p th-order accuracy of each spatial discretization, we employ p th-order extrapolation. While the inflow condition leads to a solution that mimics the periodic solution (they converge to the same solution as the mesh is refined), this does not influence the convergence of MGRIT, which is independent of the solution for linear problems.²⁵ For the numerical implementation of boundary conditions, sufficiently accurate extrapolation is used at the outflow boundary; at the inflow boundary, sufficiently accurate ERK stage values are computed using ideas similar to those in Carpenter et al.,³⁴ except we elect to use the same spatial discretization right up to the boundary rather than switching to a compact one. To approximate solution and ERK stage values at ghost points, we employ truncated Taylor series about the boundary and use the PDE with the ‘inverse Lax–Wendroff’ procedure (Reference 29, p. 364). For each scheme, numerical tests (not shown here) verify that convergence at the theoretically predicted rates is achieved. Tests also indicate that CFL limits are very similar to their analogues with periodic boundaries (Table 1).

Iteration counts for the inflow/outflow problem discretized with ERK p +Up, $p \in \{1, 3, 5\}$, are given in Table 5. Two-level solvers using different coarsening factors m , and multilevel V-cycles using $m = 4$ on each level are considered. On average, the results are very similar to those for the analogous periodic problems (right side of Table 3 for two level, and Table 4 for multilevel). The optimized coarse-grid operators for the periodic problem therefore also make excellent coarse-grid operators for the inflow/outflow problem despite them not being designed to do so in any rigorous sense. These results

TABLE 5 Iteration counts for ERK+U discretizations of (1) with inflow/outflow boundaries; Ψ is given by truncating the circulant matrix resulting from linear least squares solution (9), with its sparsity pattern based on that of Ψ_{ideal} . Left: Two-level solves as a function of coarsening factor m . Right: ℓ -level V-cycle solves using $m = 4$ coarsening on each level. A “—” denotes a hierarchy that would have coarsened to fewer than the prescribed minimum number of allowable points (four for ERK1+U1, and eight for ERK3+U3 and ERK5+U5)

		Two level, m						Multilevel, ℓ					
Scheme	$n_x \times n_t$	2	4	8	16	32	64	2	3	4	5	6	7
ERK1+U1	$2^8 \times 2^{10}$	10	6	6	6	5	3	6	6	6	6	–	–
	$2^{10} \times 2^{12}$	11	6	6	7	6	5	6	6	6	6	6	–
	$2^{12} \times 2^{14}$	11	6	6	7	6	5	6	7	7	7	7	7
ERK3+U3	$2^8 \times 2^9$	7	6	5	5	4	2	6	6	6	–	–	–
	$2^{10} \times 2^{11}$	7	6	5	6	5	4	6	7	7	7	–	–
	$2^{12} \times 2^{13}$	7	6	5	6	5	4	6	7	7	7	7	–
ERK5+U5	$2^8 \times 2^9$	8	5	4	4	4	2	5	5	5	–	–	–
	$2^{10} \times 2^{11}$	7	5	3	4	4	4	5	5	5	5	–	–
	$2^{12} \times 2^{13}$	7	5	3	4	4	4	5	5	5	5	5	–

indicate that the issues hindering convergence for hyperbolic problems in the simpler periodic setting, where Φ and Ψ are normal matrices, are also responsible for poor convergence in this more complicated setting.

4.6 | Implicit schemes

We now consider linear least squares problem (9) for the computation of suitable coarse-grid operators Ψ for SDIRK+U discretizations of (1). For such discretizations, Φ is a rational function of sparse matrices and so, too, is $\Psi_{\text{ideal}} := \Phi^m$. Naturally, one might seek a Ψ that is also of this form. However, it is not obvious how this should be done, with one complication being the choice of sparsity patterns for the numerator and denominator. Consequently, we take a different approach here.

Since Φ is a rational function of sparse matrices, it can also be written as a dense matrix. To assess to what extent Φ and Φ^m do globally couple the solution, we consider the magnitude of their entries as a function of their diagonal index, as pictured in Figure 7 for $m \in \{16, 64\}$. These plots show that, despite Φ and Φ^m being dense, they effectively act as sparse matrices, with their largest nonzeros having a sharp peak that is correlated with the characteristic departure point (shown as the dashed vertical line). The one exception here is SDIRK1+U1, whose entries are significantly less peaked than the other discretizations. As in previous examples, this is consistent with this discretization being very dissipative and not capturing the nondissipative nature of (1) well. Indeed, the plots in Figure 7 are qualitatively similar to their analogues for the ERK schemes in Figure 4 (noting the curves sit over the top of one another in the SDIRK case because the same CFL number of $c = 4$ is used for every implicit discretization).

The effectively sparse structure of Ψ_{ideal} —as shown in Figure 7—begs the question: Can a sparse (or equivalently, explicit) coarse-grid operator Ψ be used to approximate it? The use of an explicit coarse-grid operator with an implicit fine-grid discretization is certainly not standard, and in fact, the reverse case has been used elsewhere in the literature: An implicit coarse-grid operator has been coupled with an explicit fine-grid discretization since it is a natural way of ensuring that the coarse-grid operator is stable.⁴ However, quasi-tracking the solution of the PDE along characteristics—as done in the previous sections—is another way of ensuring the coarse-grid operator is stable, since the physical domain of dependence is included in the numerical domain of dependence.

We now test the idea of using a sparse Ψ to approximate a dense Φ^m . As for the ERK discretizations, we place a restriction on the number of nonzeros in Ψ . To do so, we compute the entries in the first column of Φ^m (this can be done using the FFT and its inverse), and then we select a nonzero pattern using thresholding. That is, recalling $\tilde{\phi}_k^m$ is the (dense) first column of Φ^m , we take the nonzero pattern to be that of the entries with magnitude at least equal to $\eta_{\text{tol}} \times \max_k |\tilde{\phi}_k^m|$, in which $\eta_{\text{tol}} < 1$. We find that smaller values of η_{tol} lead to more quickly converging MGRIT solvers. As for the ERK

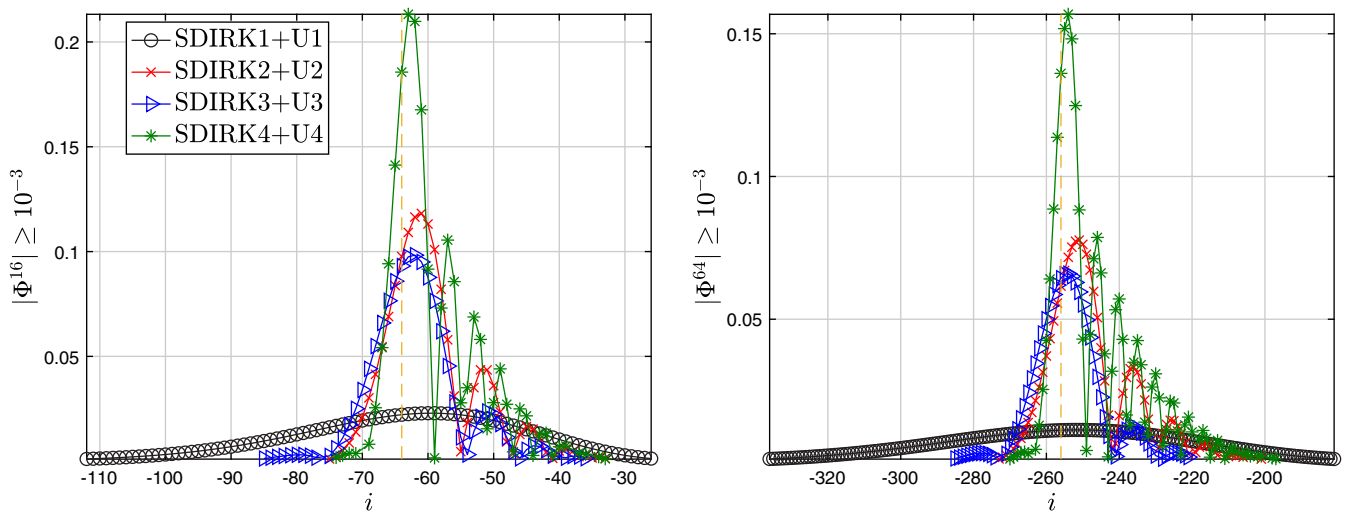


FIGURE 7 Magnitude of diagonal entries of the dense matrices $\Psi_{\text{ideal}} := \Phi^m$, $m = 16$ (left), and $m = 64$ (right), that are larger than 10^{-3} , as a function of their diagonal index, i . Fine-level discretizations, Φ , are SDIRK p +U p , $p \in \{1, 2, 3, 4\}$. A value of $n_x = 2^{10}$ has been used here. In each plot, a dashed vertical line is included to represent a distance of $-4m\Delta x$ from $i = 0$ (these schemes use CFL number of $c = 4$)

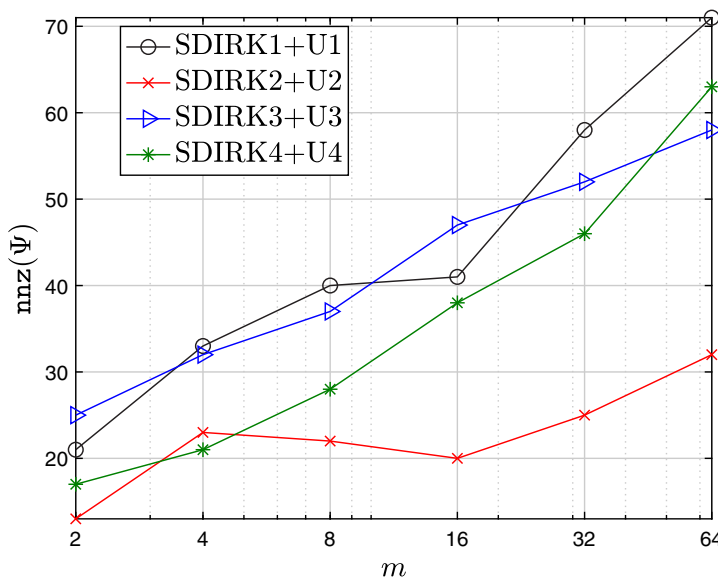


FIGURE 8 Number of nonzeros per row of Ψ for SDIRK+U discretizations as a function of coarsening factor, m

schemes, we have loosely tried to achieve some balance between the rate of convergence and the number of nonzeros in Ψ , but this has not been fully optimized. For each discretization and coarsening factor, m , we allow for a different value of η_{tol} . For $m = (2, 4, 8, 16, 32, 64)$ the values for the p th-order SDIRK+U scheme are: $p = 1$, $\eta_{\text{tol}} = (.1, .125, .25, .5, .5, .6)$; $p = 2$, $\eta_{\text{tol}} = (.05, .1, .1, .2, .2, .2)$; $p = 3$, $\eta_{\text{tol}} = (.005, .01, .02, .02, .02, .04)$; and $p = 4$, $\eta_{\text{tol}} = (.005, .01, .01, .01, .02, .02)$. These choices of η_{tol} result in coarse-grid operators that have on the order of the same number of entries shown in the plots in Figure 7. Figure 8 shows the number of nonzeros per row of Ψ as a function of the coarsening factor and how there is, in general, some growth in this number with m , just as there is in the number of nonzeros in Φ^m whose magnitude is significant (Figure 7).

The iteration counts for the solvers are shown in Table 6. Convergence is fast for all coarsening factors, and the solvers appear scalable as the mesh is refined. This is in stark contrast to the results in Table 2 where rediscretizing Φ on the coarse grid resulted in a divergent solver for all discretizations except for SDIRK1+U1, reinforcing the idea that there exist significantly better coarse-grid operators for advection problem (1) than those offered by rediscretizing the PDE on the coarse grid. Furthermore, these results confirm that despite Φ^m being a dense operator for the implicit temporal discretizations considered here, it can be well approximated by a sparse one.

TABLE 6 Two-level iteration counts for SDIRK+U discretizations with Ψ given as linear least squares solution (9)

Scheme	$n_x \times n_t$	m					
		2	4	8	16	32	64
SDIRK1+U1	$2^{10} \times 2^{10}$	10	7	8	10	8	7
	$2^{12} \times 2^{12}$	10	7	8	11	9	9
SDIRK2+U2	$2^{10} \times 2^{10}$	10	8	7	8	8	7
	$2^{12} \times 2^{12}$	11	8	7	8	8	8
SDIRK3+U3	$2^{10} \times 2^{10}$	5	5	5	4	4	4
	$2^{12} \times 2^{12}$	5	5	5	4	4	4
SDIRK4+U4	$2^{10} \times 2^{10}$	6	6	5	5	5	5
	$2^{12} \times 2^{12}$	6	6	5	5	5	5

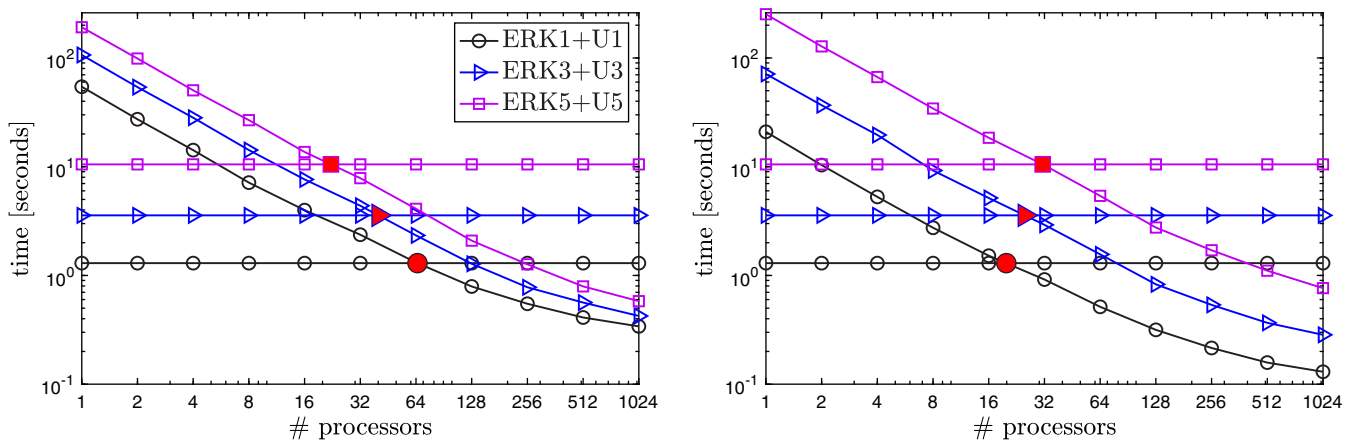


FIGURE 9 Strong parallel scaling: Runtimes of MGRIT V-cycles with $m = 4$ coarsening and using time-only parallelism for $\text{ERK}p+\text{U}p$ discretizations on space-time grids of size $n_x \times n_t = 2^{12} \times (2^{14}, 2^{13}, 2^{13})$ for $p = (1, 3, 5)$. Left: Fixed residual stopping tolerance of 10^{-10} . Right: Residual stopping tolerance based on the discretization error. Dashed lines represent runtimes of time-stepping on one processor for reference purposes. Solid red markers represent crossover points

5 | PARALLEL RESULTS

In this section, we present strong parallel scaling results for the $\text{ERK}p+\text{U}p$, with $p \in \{1, 3, 5\}$, multilevel solvers developed in Section 4.4. We show that they can lead to significant speed-ups over sequential time-stepping. We have obtained analogous parallel results for the two-level solvers developed in Section 4.3, but these lead to smaller speed-ups and so have been omitted here.

The implementations use the open-source package XBraid.³⁵ The results were generated on Quartz, a Linux cluster at Lawrence Livermore National Laboratory consisting of 2688 compute nodes, with dual 18-core 2.1 GHz Intel Xeon processors per node. For each discretization, we consider the strong scaling of a single problem whose space-time grid is the largest from Table 4, and the number of levels in the solver is taken as the maximum shown in this table. Since we want to demonstrate the benefits of parallelization in time, we only consider parallelization in the time direction. As throughout the rest of this paper, we first consider a stopping criterion based on achieving a space-time residual below 10^{-10} in the discrete L_2 -norm, but a stopping criterion based on achieving discretization error accuracy is also considered.

In our parallel tests, we have considered both V- and F-cycles with coarsening factors of $m = 2$ and $m = 4$. We find that F-cycles require fewer iterations to converge than V-cycles, but this of course comes at the cost of added work and communication. Accordingly, we typically find that the best results arise from the use of V-cycles with $m = 4$ coarsening and, thus, results for this configuration are shown here, in Figure 9. The plots show good parallel scaling with benefit over sequential time-stepping when using at least 32 processors in almost all cases, which is on par with what has been achieved for model diffusion-dominated problems using time-only parallelism.⁷ The largest speed-up achieved over sequential

time-stepping is at 1024 processors, where MGRIT is faster by a factor of about 3.8, 8.4, and 18.1 when solving up to 10^{-10} residual tolerance, and of about 10.0, 12.6, and 13.7 when solving up to discretization error (for the discretizations in the order of increasing accuracy).

The relative speed-ups shown here further demonstrate the improvements given by this work over existing parallel-in-time strategies for hyperbolic PDEs. For example, achieving MGRIT speed-up with high-order discretizations of any hyperbolic problem is unheard of in the literature, and so the fact that we have been able to achieve a speed-up on the order of 15 times for a highly accurate explicit fifth-order discretization run at a realistic CFL fraction is significant.

6 | CONCLUSIONS

In this paper, we consider the parallel-in-time integration of the one-dimensional, constant-coefficient linear advection problem using the MGRIT and Parareal algorithms. This PDE represents the simplest of all hyperbolic problems, yet, to the best of our knowledge, no parallel-in-time solvers have been successfully applied to accurate discretizations of this problem, yielding inexpensive solvers that achieve fast and scalable convergence for realistic CFL fractions close to one. We use existing convergence theory to explain why this problem is so difficult, and what is required of coarse-grid operators for its efficient solution. Convergence hinges on the coarse-grid operator accurately propagating slowly decaying spatial modes through time very similarly to the fine-grid operator. The larger number of such modes for advection-dominated problems compared with diffusion-dominated problems means that even small differences between fine- and coarse-grid operators typically result in extremely poor convergence.

For this PDE, we develop “near-optimal” coarse-grid operators through the approximate minimization of two-level error estimates. We use these coarse-grid operators for both explicit and implicit discretizations of low- and high-order accuracy and demonstrate that they lead to solvers with fast and scalable convergence that is on par with performance typically seen from these parallel-in-time algorithms when applied to diffusion-dominated problems. For explicit discretizations, we show that it is possible to overcome the CFL-driven divergence that arises from naively applying a conditionally stable discretization on the coarse grid. Primarily, this is done by tracking information along characteristic curves of the PDE on the coarse grid. Moreover, we show that observing this characteristic nature is also important for unconditionally stable implicit discretizations. Finally, our results indicate that higher-order discretizations of this problem benefit more from parallel-in-time integration compared with those of lower order.

While the optimization approach proposed here is very effective, it also has clear limitations, requiring the solution of expensive optimization problems. Further, it is not directly generalizable to more complicated PDEs than constant-coefficient linear advection. However, we expect that the heuristic insights developed here will be important for more complicated hyperbolic problems. Future work will focus on developing practical coarse-grid operators for linear advection as well as for more complicated hyperbolic problems which are of practical interest, such as those with non-constant coefficients and nonlinearities. Our major finding, that coarse operators need to track characteristics, will be a central guiding principle in this future work; indeed, this has already led to some promising results in initial research in these directions. Furthermore, the spatial parallelization of the coarse-grid operators developed here will also be considered, which is not straightforward due to their coupling of connections over longer distances in space.

ACKNOWLEDGEMENTS

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-789101). The work of Oliver A. Krzysik was supported by an Australian Government Research Training Program Scholarship, and was partially supported by International Mobility Programme funding from the ARC Centre of Excellence for Mathematical and Statistical Frontiers. The work of Scott P. MacLachlan and Hans De Sterck was partially supported by NSERC discovery grants RGPIN-2014-06032, RGPIN-2019-05692, and RGPIN-2019-04155. The authors are grateful to Matt Menickelly for helpful conversations regarding the optimization aspects of this work, and would like to thank Irad Yavneh for his insightful comments on connections between this problem and that considered in Yavneh.³² The authors are grateful to the anonymous reviewers for their constructive feedback, which has significantly improved this article. This study does not have any conflicts of interest.

DATA AVAILABILITY STATEMENT

No electronic material or data from publicly available data repositories has been used in this paper.

ORCID

Hans De Sterck  <https://orcid.org/0000-0002-1641-932X>

Stephanie Friedhoff  <https://orcid.org/0000-0001-5527-3354>

Oliver A. Krzysik  <https://orcid.org/0000-0001-7880-6512>

Scott P. MacLachlan  <https://orcid.org/0000-0002-6364-0684>

REFERENCES

- Gander MJ. 50 years of time parallel time integration. Contributions in mathematical and computational sciences. Berlin, Germany: Springer International Publishing; 2015. p. 69–113.
- Ong BW, Schroder JB. Applications of time parallelization. *Comput Vis Sci*. 2020;23(1):11.
- Lions JL, Maday Y, Turinici G. Résolution d'EDP par un schéma en temps pararéel. *C R Acad Sci Paris Sér I Math*. 2001;332(7):661–8.
- Falgout R, Friedhoff S, Kolev T, MacLachlan S, Schroder JB. Parallel time integration with multigrid. *SIAM J Sci Comput*. 2014;14(1):951–2.
- Gander MJ, Vandewalle S. Analysis of the parareal time-parallel time-integration method. *SIAM J Sci Comput*. 2007;29(2):556–78.
- Bal G, Maday YA. "Parareal" time discretization for non-linear PDE's with application to the pricing of an american put. Lecture notes in computational science and engineering. Berlin, Heidelberg/Germany: Springer; 2002. p. 189–202.
- Falgout RD, Friedhoff S, Kolev TV, MacLachlan SP, Schroder JB, Vandewalle S. Multigrid methods with space–time concurrency. *Comput Vis Sci*. 2017;18(4–5):123–43.
- Falgout RD, Lecouvez M, Woodward CS. A parallel-in-time algorithm for variable step multistep methods. *J Comput Sci*. 2019;37:101029.
- Falgout RD, Manteuffel TA, O'Neill B, Schroder JB. Multigrid reduction in time for nonlinear parabolic problems: a case study. *SIAM J Sci Comput*. 2017;39(5):S298–322.
- Chen F, Hesthaven JS, Zhu X. On the use of reduced basis methods to accelerate and stabilize the parareal method. Reduced order methods for modeling and computational reduction. New York, NY: Springer; 2014. p. 187–214.
- Dai X, Maday Y. Stable parareal in time method for first-and second-order hyperbolic systems. *SIAM J Sci Comput*. 2013;35(1):A52–78.
- Dobrev VA, Kolev T, Petersson NA, Schroder JB. Two-level convergence theory for multigrid reduction in time (MGRIT). *SIAM J Sci Comput*. 2017;39(5):S501–27.
- Gander MJ. Analysis of the parareal algorithm applied to hyperbolic problems using characteristics. *Boletín de la Sociedad Española de Matemática Aplicada*. 2008;42:21–35.
- Hessenthaler A, Nordsletten D, Röhrle O, Schroder JB, Falgout RD. Convergence of the multigrid reduction in time algorithm for the linear elasticity equations. *Numer Linear Algebra Appl*. 2018;25(3):e2155.
- Howse AJM, De Sterck H, Falgout RD, MacLachlan S, Schroder J. Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid Burgers equations. *SIAM J Sci Comput*. 2019;41(1):A538–65.
- Howse A. Nonlinear preconditioning methods for optimization and parallel-in-time methods for 1d scalar hyperbolic partial differential equations. Waterloo, Canada: University of Waterloo; 2017.
- Nielsen AS, Brunner G, Hesthaven JS. Communication-aware adaptive Parareal with application to a nonlinear hyperbolic system of partial differential equations. *J Comput Phys*. 2018;371:483–505.
- Ruprecht D, Krause R. Explicit parallel-in-time integration of a linear acoustic-advection system. *Comput Fluids*. 2012;59:72–83.
- Ruprecht D. Wave propagation characteristics of Parareal. *Comput Vis Sci*. 2018;19(1–2):1–17.
- Schmitt A, Schreiber M, Peixoto P, Schäfer M. A numerical study of a semi-Lagrangian Parareal method applied to the viscous Burgers equation. *Comput Vis Sci*. 2018;19(1–2):45–57.
- Steiner J, Ruprecht D, Speck R, Krause R. Convergence of Parareal for the Navier-Stokes equations depending on the Reynolds number. Numerical mathematics and advanced applications-ENUMATH 2013. New York, NY: Springer; 2015. p. 195–202.
- Gander MJ, Güttel S. PARAEXP: a parallel integrator for linear initial-value problems. *SIAM J Sci Comput*. 2013;35(2):C123–42.
- De Sterck H, Friedhoff S, Howse AJM, MacLachlan SP. Convergence analysis for parallel-in-time solution of hyperbolic systems. *Numer Linear Algebra Appl*. 2020;27(1):e2271.
- Hessenthaler A, Southworth BS, Nordsletten D, Röhrle O, Falgout RD, Schroder JB. Multilevel convergence analysis of multigrid-reduction-in-time. *SIAM J Sci Comput*. 2020;42(2):A771–96.
- Southworth BS. Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time. *SIAM J Matrix Anal Appl*. 2019;40(2):564–608.
- Gander MJ, Kwok F, Zhang H. Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. *Comput Vis Sci*. 2018;19(3–4):59–74.
- Friedhoff S, MacLachlan S. A generalized predictive analysis tool for multigrid methods. *Numer Linear Alg Appl*. 2015;22:618–47.
- Shu CW. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. Advanced numerical approximation of nonlinear hyperbolic equations. New York, NY: Springer; 1998. p. 325–432.
- Hesthaven JS. Numerical methods for conservation laws: from analysis to algorithms. Philadelphia, PA: SIAM; 2017.
- Butcher JC. Numerical methods for ordinary differential equations. Hoboken, NJ: John Wiley & Sons Ltd; 2003.
- Hairer E, Wanner G. Solving ordinary differential equations. II. vol. 14 of Springer Series in Computational Mathematics. 2nd ed. Berlin, Germany: Springer-Verlag; 1996.
- Yavneh I. Coarse-grid correction for nonelliptic and singular perturbation problems. *SIAM J Sci Comput*. 1998;19(5):1682–99.
- Trottenberg U, Oosterlee CW, Schuller A. Multigrid. Academic Press; 2001.

34. Carpenter MH, Gottlieb D, Abarbanel S. The stability of numerical boundary treatments for compact high-order finite-difference schemes. *J Comput Phys.* 1993;108(2):272–95.
35. XBraid: parallel multigrid in time; Available from: <http://llnl.gov/casc/xbraid>.

How to cite this article: De Sterck H, Falgout RD, Friedhoff S, Krzysik OA, MacLachlan SP. Optimizing multigrid reduction-in-time and Parareal coarse-grid operators for linear advection. *Numer Linear Algebra Appl.* 2021;e2367. <https://doi.org/10.1002/nla.2367>

APPENDIX A. RUNGE–KUTTA BUTCHER TABLEAUX

For completeness, Butcher tableaux are given here for the Runge–Kutta methods used in the main text. ERK methods are given in Table A1, and L-stable SDIRK methods are given in Table A2.

This document was prepared as an account of work sponsored by an agency of the U. S. government. Neither the U.S. government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

TABLE A1 Butcher tableaux for explicit Runge–Kutta (ERK) methods of orders 1–5

ERK1	ERK2 (Reference 29, (9.7))	ERK3 (Reference 29, (9.8))	ERK4 (Reference 30, p. 180)	ERK5 (Reference 30, (236a))																																																												
<table><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	0	0	1	1	<table><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr><tr><td></td><td>$\frac{1}{2}$</td><td>$\frac{1}{2}$</td></tr></table>	0	0	0	1	1	0		$\frac{1}{2}$	$\frac{1}{2}$	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>$\frac{1}{2}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{4}$</td><td>0</td></tr><tr><td></td><td>$\frac{1}{6}$</td><td>$\frac{1}{6}$</td><td>$\frac{2}{3}$</td></tr></table>	0	0	0	0	1	1	0	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0		$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\frac{1}{2}$</td><td>$\frac{1}{2}$</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\frac{1}{2}$</td><td>0</td><td>$\frac{1}{2}$</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td></td><td>$\frac{1}{6}$</td><td>$\frac{1}{3}$</td><td>$\frac{1}{3}$</td><td>$\frac{1}{6}$</td></tr></table>	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	1	0	0	1	0		$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0	0	0	0	0	0	0
0	0																																																															
1	1																																																															
0	0	0																																																														
1	1	0																																																														
	$\frac{1}{2}$	$\frac{1}{2}$																																																														
0	0	0	0																																																													
1	1	0	0																																																													
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0																																																													
	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$																																																													
0	0	0	0	0																																																												
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0																																																												
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0																																																												
1	0	0	1	0																																																												
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$																																																												
				$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	0																																																						
				$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	0	0																																																						
				$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	0																																																						
				$\frac{3}{4}$	$\frac{3}{16}$	$-\frac{3}{8}$	$\frac{3}{8}$	$\frac{9}{16}$	0	0																																																						
				1	$-\frac{3}{7}$	$\frac{8}{7}$	$\frac{6}{7}$	$-\frac{12}{7}$	$\frac{8}{7}$	0																																																						
					$\frac{7}{90}$	0	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$																																																						

TABLE A2 Butcher tableaux for L-stable singly diagonally implicit Runge–Kutta (SDIRK) methods of orders 1–4. Constants used in SDIRK3 are: $\zeta = 0.43586652150845899942 \dots$, $\alpha = \frac{1+\zeta}{2}$, $\beta = \frac{1-\zeta}{2}$, $\gamma = -\frac{3}{2}\zeta^2 + 4\zeta - \frac{1}{4}$, $\epsilon = \frac{3}{2}\zeta^2 - 5\zeta + \frac{5}{4}$

SDIRK1	SDIRK2 (Reference 30, p. 261)	SDIRK3 (Reference 30, p. 262)	SDIRK4 (Reference 31, (6.16))
$\begin{array}{c c} 1 & 1 \\ \hline & 1 \end{array}$	$\begin{array}{c cc} 1 - \frac{\sqrt{2}}{2} & 1 - \frac{\sqrt{2}}{2} & 0 \\ \hline 1 & \frac{\sqrt{2}}{2} & 1 - \frac{\sqrt{2}}{2} \\ & \frac{\sqrt{2}}{2} & 1 - \frac{\sqrt{2}}{2} \end{array}$	$\begin{array}{c ccc} \zeta & \zeta & 0 & 0 \\ \hline \alpha & \beta & \zeta & 0 \\ 1 & \gamma & \epsilon & \zeta \\ \hline & \gamma & \epsilon & \zeta \end{array}$	$\begin{array}{c cccc} \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \hline \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ \frac{11}{20} & \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & 0 \\ \hline \frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} \\ 1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} \\ \hline & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} \end{array}$