Multilevel parallel-in-time methods for advection-dominated PDEs

Oliver Andrew Krzysik

ORCID iD: 0000-0001-7880-6512

September 2021

School of Mathematics
Faculty of Science
Monash University

A thesis submitted for the degree of Doctor of Philosophy

Copyright notice

Declaration of Authorship

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Oliver Andrew Krzysik
September 7, 2021

Abstract

The large-scale, parallel numerical simulation of partial differential equations (PDEs) is ubiquitous across the sciences. Developing efficient methods for these simulations is a core goal of the field of computational mathematics. A key class of such problems is those describing time-dependent phenomena. Traditional algorithms for simulating time-dependent PDEs do so sequentially in time; however, it is also possible to simulate these in a parallel-in-time fashion. The field of parallel-in-time integration has seen growing interest over the past two decades, motivated, in part, by an emergence of massively parallel supercomputers. Many successful parallel-in-time strategies have been developed, particularly for diffusion-dominated problems. On the other hand, there has been limited success for hyperbolic problems, and for advection-dominated problems more broadly. The overarching goal of this thesis is to investigate these apparent shortcomings, and to present new methods for the parallel-in-time solution of advection-dominated problems.

The parallel-in-time solvers we focus on in this work are those of Parareal and multigrid reduction-in-time, both of which use multilevel or multigrid techniques in time. We begin by identifying two principal reasons for the divergence these solvers typically experience on constant-coefficient linear advection problems. From these findings, we develop a heuristic-based optimization strategy for defining coarse-grid operators, which leads to fast parallel-in-time integration, even for high-order-accurate discretizations. We develop a general, closed-form convergence theory of these solvers using the tools of local Fourier analysis. We apply this theory to reveal that poor performance for advection problems is closely linked to that experienced by spatial multigrid solvers when applied to time-independent advection problems. Following this, we develop a novel coarse-grid operator for the solution of advection problems using semi-Lagrangian discretizations. The coarse-grid operator leads to fast convergence for variable-wave-speed problems.

In the final component of the thesis, we shift from parallel-in-time techniques to consider sequential time integration. We consider the numerical solution of fully implicit Runge-Kutta (FIRK) methods when applied in the method-of-lines solution of time-dependent PDEs. FIRK methods can have excellent accuracy and stability properties, but do not see wide-spread use in practice due to the difficulty of numerically solving the underlying algebraic stage equations. We present new algorithms for solving the stage equations, including for both linear and nonlinear problems. Numerical analysis is used to optimize and prove robustness of the linear preconditioning strategy of the proposed algorithms. Numerical tests demonstrate the efficacy of the proposed approach compared to existing widely used implicit time integration methods.

Contributions statement

While the majority of the work presented in this thesis is my own, some sections are the result of collaborations. Any work that is not my own is cited appropriately in the text. Additional details are outlined below.

- Chapter 2: This work is based on the publication [25]: H. De Sterck, R. D. Falgout, S. Friedhoff, O. A. Krzysik, and S. P. MacLachlan, *Optimizing multigrid reduction-in-time and Parareal coarse-grid operators for linear advection.*

  I am the primary author of this publication, with all other co-authors playing a supervisory role. Two contributions appearing in the chapter that I would like to explicitly acknowledge are as follows. Rob Falgout is responsible for developing the key idea that coarse-grid operators should track characteristics. Stephanie Friedhoff is largely responsible for generating the parallel computing results.
- Chapter 3: This work involved helpful discussions from Hans De Sterck, Stephanie Friedhoff, and Scott MacLachlan. A peer reviewer of our published article [25]—forming the basis of Chapter 2 of this thesis—is gratefully acknowledged for initially suggesting a potential link between poor MGRIT convergence and characteristic components, which is analysed in this chapter.
- Chapter 4: This work was conducted under the supervision of Hans De Sterck and Rob Falgout.
- Chapter 5: This chapter is based largely on the manuscripts [91]: B. S. Southworth, O. A. Krzysik, W. Pazner, and H. De Sterck. *Fast solution of fully implicit Runge-Kutta and discontinuous Galerkin in time for numerical PDEs, Part I: the linear setting*; and [90]: B. S. Southworth, O. A. Krzysik, and W. Pazner. *Fast solution of fully implicit Runge-Kutta and discontinuous Galerkin in time for numerical PDEs, Part II: nonlinearities and DAEs.* These manuscripts have been accepted for publication in SIAM Journal on Scientific Computing.

  I am the secondary author of the above manuscripts, and this chapter focuses on my contributions to this collaborative work, which include theoretical analysis and software implementation. In particular, I am primarily responsible for most of the theoretical analysis presented in the chapter, and for all of the numerical results. Ben Southworth is largely responsible for the conceptualization and design of the new algorithms presented in the chapter, and made key contributions to some of the theoretical analysis shown in the chapter. Hans De Sterck supervised my work in this collaboration. This chapter has been written in a self-contained manner so that the reader need not refer to [91, 90]; however, references to [91, 90] are included when further details about certain topics can be found there.

Acknowledgements

Hans, thank you! For all of the time you have given me over the years. For all of your patience, and for all of your advice. For affording me the freedom and the opportunity to research that which I have been interested in. I also greatly appreciate all of the conference and international travel that you have facilitated for me.

Thank you, TC, for taking over official supervisory duties after Hans' departure from Monash.

I would like to thank and acknowledge all of my collaborators from whom I have learned so much, and without whose help none of this would have been possible: Rob Falgout, Stephanie Friedhoff, Scott MacLachlan, and Ben Southworth. In particular, Rob, thank you for hosting me for two summers at LLNL, even if one of them was virtually from my lounge room in Australia.

To my mum, dad, and brother, Elliot: Thank you for all of your unconditional love and support throughout the years. To Nan and Pate, thank you, for all that you have done for me.

Maddie, for all of your love and devotion over the years I am extremely grateful. I am truly sorry for boring you senseless with my seemingly endless discussions of mathematics, but I cannot make any promises that they will end with the completing of this thesis!

I am indebted to all of the helpful and friendly administrative staff within the School of Mathematics, in particular to John Chan.

I gratefully acknowledge the ARC Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS) for providing me with funding through their International Mobility Programme to visit with colleagues in Canada.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship.

Contents

Chapter 1

Introduction and background

The goal of this opening chapter is to provide some high-level discussion on, and motivation for, the core topics studied throughout this thesis. Some of the simple partial differential equations (PDEs) we study and our motivations for doing so are described in Section 1.1. A short discussion on the types of discretizations we consider for advection-dominated problems is given in Section 1.2. A brief introduction to parallel-in-time methods is given in Section 1.3. A detailed description of the particular parallel-in-time methods that are studied throughout this work (Parareal and MGRIT) is given in Section 1.4. The chapter concludes with Section 1.5 providing an outline for the remainder of the thesis.

1.1 Model PDEs and their motivations

This body of work revolves around the numerical solution of time-dependent PDEs, and thus we begin with a high-level discussion of the PDEs that are of interest to us. Broadly speaking, time-dependent PDEs may be categorized into two classes: *hyperbolic* and *parabolic*. The prototypical example of a hyperbolic equation is that of the constant-coefficient or constant-wave-speed advection equation in one spatial dimension,

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0. \tag{1.1}$$

The prototypical example of a parabolic equation is the constant-coefficient diffusion equation in one spatial dimension,

$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2}, \quad \beta > 0. \tag{1.2}$$

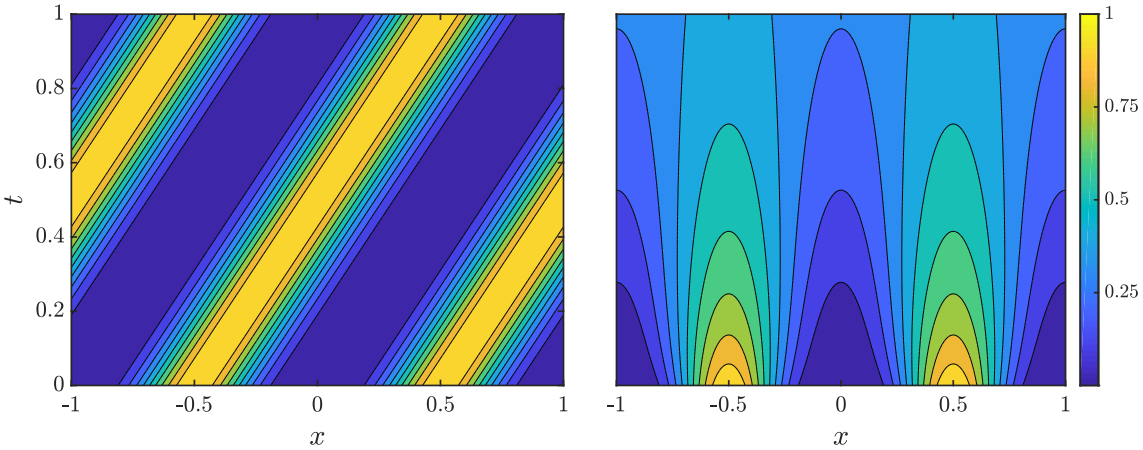

FIGURE 1.1: Contours of qualitatively representative PDE solutions on $(x,t) \in [-1,1] \times [0,1]$, subject to periodic boundary conditions in space, and the initial condition $u(x,0) = \sin^4(\pi x)$. **Left:** Linear advection equation (1.1) with $\alpha = 1$. **Right:** Diffusion equation (1.2) with $\beta = 0.05$.

Qualitatively representative solutions of these two PDEs are shown in Figure 1.1. The two problems clearly permit solutions with very different behaviours. In the case of the advection equation (1.1), the initial data prescribed at $t = 0$ propagates through space-time along the *characteristics* of the PDE, $(x,t) = (\alpha t + c, t)$ for constant c. The solution does not decay or dissipate as it propagates along characteristics. In the case of the diffusion equation (1.2), the initial data prescribed at $t = 0$ propagates forward through time, but diffuses as it does so. Moreover, Fourier components of initial data diffuse in proportion to their frequency, with high-frequency components diffusing more quickly than those with low frequency. Globally, this results in the solution becoming smoother as time increases.

This work is primarily focused on hyperbolic equations. However, it will often be useful to think of these equations as arising in the advection-dominated limit or hyperbolic limit of certain parabolic PDEs. The prototypical example of such a problem is the advection-diffusion equation,

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = \beta \frac{\partial^2 u}{\partial x^2}, \quad \beta \geq 0. \tag{1.3}$$

This PDE is technically parabolic for any $\beta > 0$, but its behaviour is close to that of a hyperbolic equation in the advection-dominated regime, $|\alpha| \gg \beta$.

The numerical solution of hyperbolic PDEs is a rich and complex subject for a variety of reasons. Most famously, it is often the case for nonlinear hyperbolic PDEs that discontinuous solutions may form in finite time from smooth initial data. As one might anticipate, discontinuous solutions present enormous difficulty for numerical methods, creating non-trivial stability and accuracy issues. Despite much of the literature on numerical methods

for hyperbolic PDEs over the past half century concentrating on problems with discontinuous solutions, this thesis considers only smooth solutions of hyperbolic PDEs. In fact, a large part of the work here will be based on the constant-coefficient advection problem (1.1).

Undoubtedly, (1.1) is amongst the simplest of all PDEs, yet, since the inception of numerical methods for hyperbolic PDEs, considering it as a model problem for numerical method development has proven invaluable. The linear advection equation and/or the closely related two-way wave equation $\frac{\partial^2 u}{\partial t^2} - \alpha^2 \frac{\partial^2 u}{\partial x^2} = 0$ has appeared in many pioneering papers in the field, such as the famous 1928 paper [21] of Courant, Friedrichs, and Lewy that described the conditional stability of finite-difference schemes for hyperbolic problems, now dubbed the *CFL condition*.[1] Other examples include the papers [65, 60] on the now widely used *weighted essentially non-oscillatory* discretizations for hyperbolic problems with discontinuities. Moreover, one would struggle to find a textbook on the subject of numerical methods for hyperbolic PDEs that does not use at least one of these PDEs as a fundamental example [62, 63, 29, 31, 52].

All of the above brings us to the relevance of (1.1) being a primary focus of this work. For the most part, this thesis focuses on the parallel-in-time solution of hyperbolic problems, and, as we will come to learn, the efficient parallel-in-time solution of even the simple advection problem (1.1) is by no means simple. Developing an appreciation and understanding of why this is the case is a key step towards developing parallel-in-time methods for more complex problems. Moreover, it does not seem unreasonable to assume that one cannot develop truly efficient parallel-in-time methods for general nonlinear hyperbolic problems with discontinuous solutions if one cannot even efficiently solve (1.1) in a parallel-in-time manner. Finally, it is worth noting that while much of the literature for numerical methods for hyperbolic PDEs does focus on nonlinear problems, there remain many important applications that require the solution of linear hyperbolic problems, with common examples including passive tracer flow, acoustics, elasticity, and electromagnetics (see, e.g., [62]).

1.2 Discretizations of advection problems

There are two levels at which one can consider the numerical approximation of PDEs. The first is the discretization itself of the PDE, which is the approximation of the continuous problem with a discrete linear or nonlinear algebraic system. The second is the numerical solution of the approximate, discrete problem. We will focus on the latter part. However,

[1] See [22] for an English translation of [21].

developing efficient solvers for the discrete algebraic system will require us to utilize PDE- and discretization-specific knowledge.

We consider two classes or strategies of discretization for time-dependent, advection-dominated PDEs. The first is the method-of-lines technique. It works by first discretizing the spatial components of a PDE, resulting in a system of time-dependent ordinary differential equations (ODEs) that may be solved via a standard ODE solver. In the context of advection problems, method-of-lines discretizations are examples of *Eulerian* discretizations. For hyperbolic PDEs, Eulerian discretizations are the most widely used (see, e.g., [84, 62, 63, 29, 52]). This is in spite of many Eulerian methods suffering from a CFL limit, in particular, those that are explicit, which places a hard upper limit on the time-step size they may use. The principle underlying the CFL limit is that numerical stability requires the physical domain of dependence (i.e., the continuous one of the PDE) lie inside the numerical domain of dependence. In certain situations, this limit leads to inefficiency in the sense that it necessitates the use of restrictively small time-step sizes, which are otherwise smaller than one would like to use for accuracy reasons. As we will see, the CFL limit of explicit Eulerian methods presents somewhat of a barrier to their parallel-in-time solution.

Remaining in the context of advection problems, a contrast to Eulerian methods are methods in Lagrangian form. Lagrangian methods do not use a fixed mesh as Eulerian methods do. Instead, they consider how particles (or fluid elements) are advected by the PDE. We will consider what are known as *semi-Lagrangian* methods, which may be seen as a hybrid of Eulerian and Lagrangian methods. They too work by advecting particles, but they do so on a fixed Eulerian-like mesh. Semi-Lagrangian methods are less widely used that Eulerian methods; however, since they effectively track the flow of the PDE, they remain free of a CFL limit (for the most part). This is one of the reasons why these methods will be of great interest in the parallel-in-time solution of advection problems. For this reason also, semi-Lagrangian methods see widespread use in areas where the CFL limits of Eulerian methods impose the use of overly restrictive time-step sizes. One popular example is the area of numerical weather prediction [68, 95, 106]. They have also been used to simulate non-smooth particle transport [17, 74, 16], for example.

1.3 Parallel-in-time methods

The majority of this thesis focuses on the development of parallel-in-time methods for PDEs. We now provide an introduction to and overview of the area. The defining principle of a parallel-in-time method is that it computes the solution to a time-dependent problem at multiple points in time simultaneously and in parallel. For our purposes, this problem

will be a discretized time-dependent ODE or PDE. Parallel-in-time methods contrast with the traditional approach of time-stepping for solving time-dependent problems, in which the solution is advanced forwards in time sequentially, one point after another.

At first, the notion of parallel-in-time methods can be counter-intuitive due to the causal nature of time itself: The solution at any future time is determined only by the solution prior to it, so if the prior solution is not yet known, then how can a future solution be computed? Does the causality of time actually make this problem difficult? For example, consider a time-dependent ODE with a time-periodic condition rather than an initial value. There is no longer any causality in time (at least in a global sense) since the solution at any point in time depends on the solution at all other points in time. Would the parallel solution of this problem present more or less difficultly than its causal initial-value counterpart? In this sense, parallel-in-time methods are not so different conceptually from existing parallel solution strategies for time-independent problems. For example, for a steady-state boundary-value problem (e.g., a Poisson problem), the solution at any point in space depends on the solution at all other points in space, yet techniques have existed for many decades to compute solutions to such problems in parallel.

Parallel-in-time research dates back to at least 1964, with the work of Nievergelt [70], remarkably before the advent of parallel computers. Since Nievergelt's pioneering work, there has been significant progress made in the field, particularly within the past two decades. A history and broad survey of the field is given in the review [41], with the later review of [72] providing a summary of more recent developments. Architectural changes in large-scale parallel computing over the past two decades have led to an increased relevance for parallel-in-time methods. Specifically, speeds of computer processors have stagnated due to physical limitations, and in response, gains in the compute power of large-scale parallel machines has instead come from rapid increases in processor counts (see, e.g., [79, 72]). Therefore, decreases in wall-clock time of large-scale simulations can be made by the adoption of algorithms with increased parallelism or concurrency. This is precisely what parallel-in-time methods set out to do: They add parallelism into the time direction where the traditional or standard approach of time-stepping is inherently sequential in time.

Parallel-in-time methods developed thus far fall into four classes [41, 72]: multiple shooting; waveform relaxation and domain decomposition; multigrid; and, direct methods. These classifications are loose, however, in the sense that some methods fall into multiple categories. Considering the numerical solution of a PDE in both space and time, there are two different approaches offering temporal parallelism. There are solvers that treat the whole of space and time simultaneously, or there are solvers that focus only on parallelising in the time direction. Some examples of methods that treat space and time simultaneously

are [101, 102, 54, 55, 30, 45, 96, 85]. Many of these methods couple waveform relaxation techniques in time with multigrid techniques in space, or use multigrid techniques in both space and time. A detailed comparison of several methods using multigrid-style techniques is given in [33].

In contrast to the above methods, Parareal [64] and multigrid reduction-in-time (MGRIT) [32], in their most basic form, are agnostic to the spatial components of a problem and focus on parallelizing in the time direction. Parareal was introduced in 2001 by Lions, Maday, and Turinici [64], and is perhaps the most well-known parallel-in-time method. Parareal was originally motivated as a multiple shooting method, but also can be interpreted as a two-level, iterative multigrid method [46]. MGRIT was introduced in 2014 by Falgout et al. [32] and is a truly multilevel method that uses reduction-based strategies to define a sequence of coarser-in-time problems. In fact, MGRIT can be seen as a multilevel generalization of Parareal [32] (see also [44] for more information on this connection). For this reason, throughout this thesis, we will make no significant differentiation between MGRIT and Parareal, and will typically refer collectively to these algorithms as 'MGRIT.' When discussing literature, however, we will adopt the nomenclature used by the authors. The parallel-in-time work in this thesis focuses exclusively on the MGRIT algorithm, and, as such, a detailed algorithmic description of it is now provided in Section 1.4. Further discussion of Parareal and MGRIT literature will be made in individual chapters where it is most relevant.

1.4 Algorithmic description of MGRIT

Consider an initial-value, time-dependent PDE problem posed on the finite time interval $t \in [0, T]$. Let the interval be discretized with $n_t + 1$ equidistant points $(t_n)_{n=0}^{n_t}$, where $t_n := n\delta t$. Suppose the PDE has been discretized in both space and time, with the only restriction being that the resulting method is of *one-step* form in time. The fully discrete problem can then be expressed as

$$\boldsymbol{u}_{n+1} = \Phi \boldsymbol{u}_n + \boldsymbol{g}_{n+1}, \quad n = 0, \ldots, n_t - 1, \tag{1.4}$$

where $\boldsymbol{u}_n$ is used to denote the approximation of the PDE solution at time $t = t_n$, which is a vector since the discretization presumably approximates the PDE solution on some spatial mesh. Here, Φ is dubbed the time-stepping operator, since it steps the discrete solution from one time point to the next. For example, Φ could arise through a method-of-lines discretization, or a semi-Lagrangian discretization. In general, Φ could be a nonlinear function, and it could also explicitly depend on time. However, to simplify our presentation, let us assume that Φ is linear (i.e., it is a matrix), and time-independent. We direct

the reader to one of [32, 35, 58] for descriptions of MGRIT for nonlinear problems. In (1.4), the $\boldsymbol{g}$ term contains any solution-independent terms such as boundary conditions or source terms.

Consider writing the equations (1.4), supplemented with their initial condition, as a block system

$$A\boldsymbol{u} := \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{n_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{n_t} \end{bmatrix} =: \boldsymbol{b}. \tag{1.5}$$

The solution of this system is trivially obtained through (block) forward substitution, but this is an inherently sequential procedure, offering no possibility for parallelization. In fact, a block forward solve of (1.5) is equivalent to sequential time-stepping. In contrast, a parallel-in-time method solves this system in parallel, or for all values of $\boldsymbol{u}$ concurrently. A block forward solve of this system is optimal in the sense that it requires only $\mathcal{O}(n_t)$ floating point operations (FLOPs). Often, parallel-in-time methods introduce parallelism at the expense of additional computation. The parallel-in-time solution of (1.5) will therefore typically require substantially more FLOPs than a forward solve, although this would ideally still only be $\mathcal{O}(n_t)$, just with a larger constant than forward substitution. For certain problems, MGRIT has been shown to be an $\mathcal{O}(n_t)$ solver [32].

One way in which parallel-in-time algorithms, such as MGRIT, conceptually differ from traditional parallel solvers is that they introduce a *lot* of additional computation. For this reason, they typically have low parallel efficiencies. In essence, this is because parallel-in-time algorithms do not simply parallelize an existing sequential algorithm, as many parallel solvers do. Instead, they replace a sequential algorithm (time-stepping), that is already optimal in terms of the amount of work it does, with an algorithm that does significantly more work, with the goal of reducing wall-clock time.

MGRIT is an iterative multigrid method for solving (1.5) in a parallel manner. For a large class of problems, multigrid methods are among the most efficient solvers, being able solve a problem with n unknowns in approximately $\mathcal{O}(n)$ FLOPs. Furthermore, multigrid methods are typically highly parallelizable, and, so, are ubiquitous throughout scientific computing. See the textbooks [13, 99] for introductions to the subject area. The methods were originally developed to solve structured discretizations of elliptic boundary value problems, of which the Poisson equation is the prototypical example, and have since been developed for a larger class of problems. While there has undoubtedly been much progress in extending their applicability, most of their success has been for symmetric, or near-symmetric problems, with the efficient multigrid solution of many highly non-symmetric

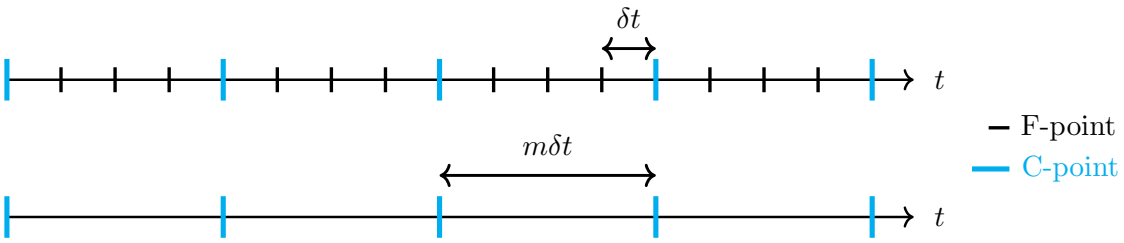

FIGURE 1.2: A fine grid with time-step δt (**top**), and coarse grid with time-step $m\delta t$, with $m = 4$ (**bottom**). F-points appear exclusively on the fine grid, while C-points appear on both the fine and coarse grids.

problems, such as (1.5), or implicit time discretizations of (1.1), remaining an active area of research.

Returning to the MGRIT solution of (1.5), suppose that the time points $(n\delta t)_{n=0}^{n_t}$ constitute a 'fine grid,' and let a coarsening factor $m \in \mathbb{N}$ induce a 'coarse grid' that consists of every mth fine-grid point, $(mn\delta t)_{n=0}^{n_t/m}$ (assuming for simplicity that n_t is divisible by 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. See Figure 1.2 for an example.

As is standard for a multigrid method, an MGRIT iteration carries out pre-relaxation, a coarse-grid correction, and then post-relaxation. Suppose that we have an approximation to the solution of (1.5) given by $\boldsymbol{v} \approx \boldsymbol{u}$, with the algebraic residual defined by $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{v}$. Then, there are two fundamental types of relaxation carried out on the system $A\boldsymbol{v} \approx \boldsymbol{b}$: F-relaxation, which uses $\boldsymbol{v}$ at C-points to solve for the solution at the intervening F-points; and, C-relaxation, which uses $\boldsymbol{v}$ at the F-points that precede C-points to solve for the solution at C-points. See Figure 1.3 for a schematic example. Therefore, F-relaxation sets the residual $\boldsymbol{r}$ to zero at F-points, and C-relaxation sets the residual $\boldsymbol{r}$ to zero at C-points. Equivalently, as pictured in Figure 1.3, F-relaxation can be thought of as time-stepping $\boldsymbol{v}$ from each C-point across the interval of $m-1$ F-points that follow it, and C-relaxation can be thought of as time-stepping $\boldsymbol{v}$ from the last F-point in each interval to its neighbouring C-point. F- and C-relaxations are completely local operations and are therefore highly parallelizable.

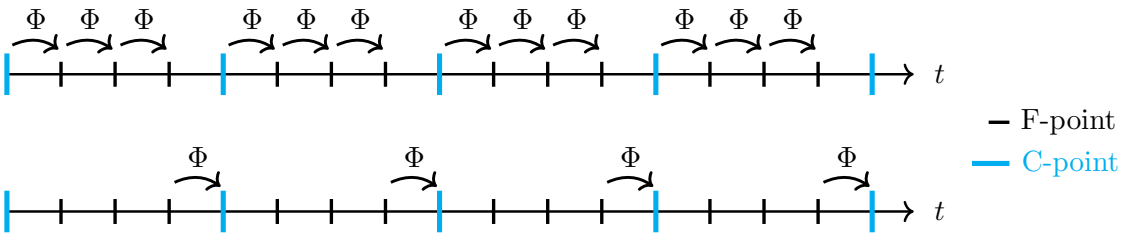

FIGURE 1.3: The two fundamental types of relaxation used in MGRIT. **Top:** F-relaxation. **Bottom:** C-relaxation.

The standard pre-relaxation sweep performed in MGRIT is either: F-relaxation, or the stronger FCF-relaxation, which is an F-, followed by a C-, then by an F-relaxation. Throughout this thesis, we exclusively use FCF-relaxation since we typically find that it gives more robust convergence (see also [32]). Note that in the Parareal literature, an F-relaxation rather than an FCF-relation is most commonly used. After the coarse-grid correction, post-relaxation is performed, which is simply an F-relaxation. Note that one FCF relaxation is already almost a factor of two times as expensive as solving (1.5) by sequential time-stepping (the exact factor is $2 - \frac{1}{m}$, $m > 1$). Therefore, even if MGRIT converges in a small number of iterations, say $\mathcal{O}(10)$, for example, this means that solving (1.5) with MGRIT will always be much more expensive in terms of FLOPs than sequential time-stepping. However, the crucial advantage of MGRIT is that its iterations can be parallelized efficiently, whereas time-stepping is inherently sequential.

In MGRIT, the coarse-grid correction problem is derived from the Schur complement of the fine-grid residual equation with respect to the C-points, or equivalently, by eliminating the F-point variables from the residual equation (the process of algebraically eliminating a subset of the variables is known as reduction, and hence the name multigrid reduction-in-time). Recall for an approximation of (1.5) given by $\boldsymbol{v} \approx \boldsymbol{u}$, the residual equation relates the algebraic error and the residual by $A\boldsymbol{e} = \boldsymbol{r}$, where $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{v}$, and $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{v}$. Now, suppose pre-relaxation is carried out using $\boldsymbol{v}$, then a new approximation, error, and residual will be generated, all of which we will still denote by $\boldsymbol{v}$, $\boldsymbol{e}$, and $\boldsymbol{r}$, respectively. After pre-relaxation (which always ends with an F-relaxation), the residual will be zero at F-points, and therefore the residual equation is

$$\begin{bmatrix} I & & & & & & & & & & \\ -\Phi & I & & & & & & & & & \\ & \ddots & \ddots & & & & & & & & \\ & & -\Phi & I & & & & & & & \\ & & & -\Phi & I & & & & & & \\ & & & & -\Phi & I & & & & & \\ & & & & & \ddots & \ddots & & & & \\ & & & & & & \ddots & \ddots & & & \\ & & & & & & & -\Phi & I & & \\ & & & & & & & & -\Phi & I & \\ & & & & & & & & & -\Phi & I \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_0 \\ \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_{m-1} \\ \boldsymbol{e}_m \\ \boldsymbol{e}_{m+1} \\ \vdots \\ \vdots \\ \boldsymbol{e}_{n_t-1} \\ \boldsymbol{e}_{n_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_0 \\ 0 \\ \vdots \\ 0 \\ \boldsymbol{r}_m \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \boldsymbol{r}_{n_t} \end{bmatrix}. \tag{1.6}$$

Simple algebra reveals that the error at the kth C-point can be expressed in terms of the error at the $k-1$st C-point according to $\boldsymbol{e}_{km} = \boldsymbol{r}_{km} + \Phi^m \boldsymbol{e}_{(k-1)m}$. Therefore, the error at

C-points satisfies the global system

$$A_{\text{ideal coarse}}\boldsymbol{e}_{\text{ideal coarse}} := \begin{bmatrix} I & & & \\ -\Phi^m & I & & \\ & \ddots & \ddots & \\ & & -\Phi^m & I \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_0 \\ \boldsymbol{e}_m \\ \vdots \\ \boldsymbol{e}_{n_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{r}_m \\ \vdots \\ \boldsymbol{r}_{n_t} \end{bmatrix} =: \boldsymbol{r}_{\text{coarse}} \tag{1.7}$$

In other words, (1.7) is the Schur complement of the fine-grid residual equation $A\boldsymbol{e} = \boldsymbol{r}$ that results after pre-relaxation. Rather than solving the Schur complement system (1.7), MGRIT instead solves the system

$$A_{\text{coarse}}\boldsymbol{e}_{\text{coarse}} := \begin{bmatrix} I & & & \\ -\Psi & I & & \\ & \ddots & \ddots & \\ & & -\Psi & I \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{e}}_0 \\ \hat{\boldsymbol{e}}_m \\ \vdots \\ \hat{\boldsymbol{e}}_{n_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{r}_m \\ \vdots \\ \boldsymbol{r}_{n_t} \end{bmatrix} = \boldsymbol{r}_{\text{coarse}}, \tag{1.8}$$

in which $\Psi \approx \Phi^m$ is the *coarse-grid time-stepping operator*, and gives rise to the approximate coarse-grid error $\boldsymbol{e}_{\text{coarse}} \approx \boldsymbol{e}_{\text{ideal coarse}}$. After solving (1.8), the coarse-grid error $\boldsymbol{e}_{\text{coarse}}$ is simply injected to the fine grid (i.e., added to C-point values of $\boldsymbol{v}$), and then post-relaxation propagates the correction on to F-points with an F-relaxation.

The coarse-grid problem (1.8) can either be solved sequentially with a forward solve, resulting in a two-level method, or the algorithm can be applied recursively since (1.8) has the same block lower-bidiagonal structure as the original fine-grid problem (1.5), resulting in a multilevel method. Recall that Parareal is a two-level method, and thus it solves (1.8) via sequential forward substitution. Parallelism is limited in the two-level case because the coarse-grid problem size of n_t/m may still be large enough to present a serial bottleneck.

Taking $\Psi = \Phi^m =: \Psi_{\text{ideal}}$ in (1.8) defines an *ideal* coarse-grid time-stepping operator in the sense that the exact solution of (1.5) is reached in a single MGRIT iteration, since then (1.8) really is the Schur complement system (1.7). However, no speed-up in parallel can be achieved with $\Psi = \Phi^m$ since computations on the coarse grid are as expensive as on the fine grid. For example, if a two-level method used $\Psi = \Phi^m$, then the sequential coarse-grid solve requires as much work as time-stepping across the whole fine grid with Φ. One should therefore 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. Most commonly in the literature, Ψ is chosen through the process of rediscretizing Φ on the coarse grid. That is, if Φ represents a specific time discretization, for example, implicit Euler, with a time-step size of δt, then Ψ is also chosen as implicit Euler, with the enlarged coarse-grid time step $m\delta t$ [26, 28, 32]. However, other techniques have also been considered, such as

coarsening in the order of accuracy of the discretization, akin to a p-multigrid method, rather than coarsening the time-step which is the classical h-multigrid approach [34, 69].

To summarise, the pseudo code for two-level MGRIT is given in Algorithm 1. Note that the algorithm is easily extended to multiple levels by replacing Line 4 with a recursive call to the next coarsest level, if not already on the coarsest level.

Algorithm 1 Two-level MGRIT for $A\boldsymbol{u} = \boldsymbol{b}$, with initial iterate $\boldsymbol{v} \approx \boldsymbol{u}$.

1: **while** residual norm larger than tolerance **do**
2: F(CF) relax on $A\boldsymbol{v} = \boldsymbol{b}$ ▷ Pre-relaxation using Φ
3: Compute C-point residual $\boldsymbol{r}_{\text{coarse}}$
4: Solve $A_{\text{coarse}}\boldsymbol{e}_{\text{coarse}} = \boldsymbol{r}_{\text{coarse}}$ ▷ Sequential forward solve using Ψ
5: Correct C-point approximation with $\boldsymbol{e}_{\text{coarse}}$ ▷ Injection interpolation
6: F-relax on $A\boldsymbol{v} = \boldsymbol{b}$ ▷ Post-relaxation using Φ

MGRIT is an iterative algorithm. Its convergence behaviour is governed by the properties of the coarse-grid operator Ψ, and how accurately this approximates the ideal coarse-grid operator, Φ^m. While MGRIT has been shown to be efficient for many diffusion-dominated problems, it has had markedly less success for advection-dominated problems. The key to obtaining an efficient MGRIT solver is identifying a *good* coarse-grid operator, which is what the majority of this thesis is dedicated to, in the context of advection-dominated problems (see Chapters 2 to 4).

Somewhat atypical of iterative solvers, MGRIT converges 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 fine-grid relaxation scheme [32]. It is easy to see in Figure 1.3 that, starting from the initial condition (i.e., the left-most C-point), a single iteration of an MGRIT algorithm using F-relaxation will obtain the exact solution at all points upto and including the second C-point. After another iteration, the exact solution will be obtained at all points upto and including the third C-point, and so on. However, should MGRIT converge only in the regime where the initial solution has been propagated across much of the (fine-grid) domain, then it is of no practical interest since no speed-up in parallel can possibly be obtained. As such, throughout this thesis, we use the term 'divergence' to describe this type of convergence behaviour.

Definition 1.1 (Divergence of MGRIT)**.** *If MGRIT converges only in a number of iterations which is close to that for which it propagates the initial condition exactly across the entire fine-grid interval, that is, $\mathcal{O}(n_t/(2m))$ iterations for FCF pre-relaxation (see [32]), then we say that it diverges.*

Note that our word choice of 'divergence' to describe the above scenario is motivated by the observation that often, but not always, the residual grows initially in such cases,

and sometimes very strongly, before eventually decaying to zero. In the literature, this behaviour is also commonly referred to as instability of the solver [83, 80].

Finally, let us conclude with a numerical example that serves to contextualize results throughout the remainder of the thesis. We will solve the diffusion equation $\frac{\partial u}{\partial t} = 0.05\frac{\partial^2 u}{\partial x^2}$, $(x,t) \in (-1,1) \times (0,1]$ subject to periodic boundary conditions in space, and the initial condition in time $u(x,0) = \sin^4(\pi x)$. The exact solution of this PDE is plotted in the right panel of Figure 1.1. The PDE is discretized in space using 2nd-order central finite differences, and in time with a 2nd-order L-stable singly diagonally implicit Runge-Kutta (SDIRK) method (see Appendix A.1). A time-step of $\delta t = h$ is used, for spatial mesh width h. We consider two different MGRIT solvers: One using only two levels and a coarsening factor of $m = 16$, and a second using a multilevel V-cycle with a coarsening factor of $m = 4$ on each level, with coarsening performed until there are just two points on the coarsest level. On coarse levels, the time-stepping operator is given by rediscretizing the fine-grid problem (that is, reapplying the same discretization as on the fine grid, just with the enlarged time-step of $m\delta t$). The initial guess of the space-time solution is chosen to be uniformly random.

Let us focus on the convergence speed of MGRIT rather than its parallel performance (convergence speed of the MGRIT algorithm in terms of the number of iterations is identical whether it is run serially or in parallel). To do so, we consider the two-norm of the residual of the space-time system (1.5) as a function of the number of MGRIT iterations, which is shown in Figure 1.4. Observe from the plots that convergence is very fast, reaching a

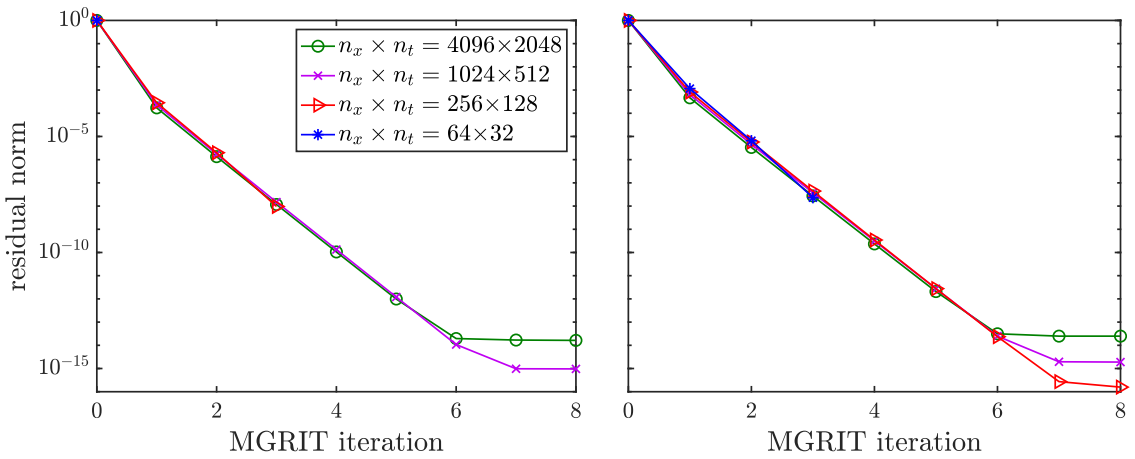

FIGURE 1.4: Two-norm of the space-time residual (relative to its initial value) for the MGRIT solution of the diffusion equation problem from Figure 1.1 discretized with $n_x \times n_t$ points in space-time. **Left:** Two-level solver using coarsening factor $m = 16$. **Right:** Multilevel V-cycles using $m = 4$ on every level.

machine zero residual norm in ~ 7 iterations.[2] The convergence is scalable in the sense that the asymptotic convergence rate is effectively constant as the problem size grows. In other words, the problems are solved with $\mathcal{O}(n_t)$ work, and, so, we say that the solver scales optimally for this problem. Another important observation is that the convergence appears robust with respect to parameters of the algorithm, since there is no significant change in going from two levels to multiple levels, or in changing the coarsening factor. In summary, MGRIT convergence is fast, scalable, and robust on this diffusion equation, and would likely result in large speed-ups over sequential time-stepping if the solves were run on a parallel machine with a sufficient number of processors.

For hyperbolic equations, and advection-dominated equations more broadly, this approach of rediscretizing the fine-grid problem does not give a usable solver, even for simple problems such as the one-dimensional, constant-coefficient equation (1.1). Obtaining scalable MGRIT convergence as in Figure 1.4 for these equations is an open problem that this thesis will solve.

1.5 Thesis outline

We now provide an outline for the remaining chapters of the thesis.

In Chapter 2, *Optimizing MGRIT coarse-grid operators for linear advection*, we explore the application of MGRIT to the constant-coefficient linear advection equation. It is demonstrated that efficiently solving this problem is very difficult. We derive a heuristic optimization strategy to produce coarse-grid operators that are approximately optimal in some sense. Detailed numerical examples show the coarse-grid operators yield highly efficient MGRIT convergence.

In Chapter 3, *Closed-form Fourier analysis of MGRIT with applications to advection-dominated problems*, we analyse the convergence behaviour of MGRIT through the lens of Fourier analysis. Analytical expressions are derived that approximately describe MGRIT convergence behaviour for certain initial-value problems. We then apply the Fourier theory to develop a better understanding of what it is about advection-dominated problems that makes them difficult to efficiently solve with MGRIT.

In Chapter 4, *Fast MGRIT for advection via dissipatively corrected coarse-grid operators*, based, in part, on findings from previous chapters, we develop a novel coarse-grid operator

[2] There are some residual curves that vanish from the plots after some number of iterations; for example, the $n_t = 32$ and $n_t = 128$ curves in the left panel are not shown after the 0th and 3rd iterations, respectively. This is because the residual is exactly zero after 1 and 4 iterations for these values of n_t when $m = 16$ (see the discussion before Definition 1.1).

for advection-dominated problems discretized with semi-Lagrangian methods. The operator generalizes the one presented in Chapter 2, with it being applicable to a much wider set of problems. Numerical tests demonstrate the efficacy of the operator in a wide variety of settings, including for high-order discretizations and variable-coefficient problems.

Chapter 5, *Fully implicit Runge-Kutta methods for method of lines*, considers sequential time-stepping rather than parallel-in-time integration. Here, we study the numerical solution of fully implicit Runge-Kutta methods for method-of-lines discretizations. New algorithms for the solution of this problem are presented. In-depth theoretical analyses are conducted to assess the efficacy of the linear preconditioners that lie at the center of the proposed algorithms. Detailed numerical experiments are given which confirm the theoretical analyses. While this chapter does not consider parallel-in-time integration, our theoretical analysis is motivated by the time integration of advection-dominated problems. In particular, we analyse the proposed algorithms for cases in which the spatial discretization is non-symmetric and/or non-normal, which is common for hyperbolic problems. Moreover, our numerical tests focus on linear and nonlinear advection-diffusion problems.

Finally, the main findings of the thesis are summarized in Chapter 6, and avenues of future research are discussed.

Chapter 2

Optimizing MGRIT coarse-grid operators for linear advection

2.1 Literature survey and outline

For a wide variety of diffusion-dominated problems, MGRIT and Parareal can achieve a significant reduction in wall clock time over sequential time-stepping methods, given enough parallel resources [64, 2, 32, 33, 34, 35, 72]. 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 [19, 23, 28, 46, 40, 50, 58, 57, 69, 81, 80, 83, 97] (recall our definition of divergence given in Definition 1.1.) 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 [57, 83, 97].

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 behaviour of standard Parareal, but are significantly more expensive, and so their practicability is limited [19, 23, 81]. The coupling of semi-Lagrangian coarse-grid operators with Eulerian fine-grid operators was considered in [83] for the viscous Burgers equation. Unfortunately, though, their numerical tests showed that convergence deteriorates significantly in the zero-viscosity/hyperbolic limit [83]. Coarsening also in space was considered in [58] when applying MGRIT to the linear advection and Burgers' equations to provide both cheaper multigrid cycles and overcome CFL 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. In [69], modest speed-ups were achieved 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 have been obtained with ParaExp [42], but this algorithm falls outside of the class of MGRIT-like algorithms that we study in this thesis. 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 [35].

A number of theoretical convergence analyses have been developed for Parareal and MGRIT, which have helped to explain numerical convergence results, and will likely play an important role in the design of new solvers [26, 28, 46, 40, 51, 80, 88, 44, 38]. Furthermore, some theoretical studies have identified potential roadblocks for fast parallel-in-time convergence of hyperbolic PDEs [46]. 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 chapter is to demonstrate that, in fact, Parareal and MGRIT, with the right choice of coarse-grid operator, can efficiently integrate hyperbolic PDEs. To do so, we work in an idealized environment, in which the constant-coefficient linear advection problem in one spatial dimension is subject to periodic spatial boundary conditions, such that existing sharp MGRIT convergence theory can be appealed to. Informed by convergence theory and the PDE, heuristics are developed that coarse-grid operators should satisfy and optimization problems based on these are formulated to find 'near-optimal' coarse-grid operators. For example, one such heuristic we develop here is that coarse-grid operators should track information along characteristics, similar to the semi-Lagrangian schemes considered in [83]. However, our optimization-based coarse-grid operators lead to robust solvers in the hyperbolic limit, unlike the semi-Lagrangian coarse-grid operators in [83]. It is demonstrated 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 diffusion-dominated 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 fine-grid CFL numbers are used that reflect what would realistically be used with sequential time-stepping.

The remainder of this chapter is organized as follows. In Section 2.2, the model problem and its discretizations are introduced. Section 2.3 provides some motivating numerical examples that highlight the difficulty of solving the seemingly simple model problem. A discussion on convergence theory and what it reveals about the difficulty of hyperbolic problems is given in Section 2.4. Section 2.5 develops an optimization-based approach for finding effective coarse-grid time-stepping operators. Parallel results are given in Section 2.6 for some of the newly developed coarse-grid operators. Concluding remarks and a discussion of future work is the subject of Section 2.7.

2.2 Preliminaries

In this section, the model problem and its discretizations are outlined.

2.2.1 Model problem and discretizations

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

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0, \quad (x,t) \in [-1,1] \times (0,T], \quad u(x,0) = \sin^4(\pi x), \tag{2.1}$$

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 thesis, periodic boundary conditions in space will be considered, but inflow/outflow boundaries will be considered in Section 2.5.5.

To numerically approximate the solution of (2.1), 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. The method of lines is employed to generate a semi-discretized representation. First, a pth-order upwind finite-difference spatial discretization is applied to (2.1), resulting in the system ODEs

$$\boldsymbol{u}'(t) = \mathcal{L}\boldsymbol{u}(t), \quad t \in (0,T], \quad \boldsymbol{u}(0) = u(\boldsymbol{x},0), \tag{2.2}$$

in which $\mathcal{L}\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ represents the discretization of $-\alpha\frac{\partial}{\partial x}$ 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, upwind-finite-difference spatial discretizations of orders 1–5 are used, which will be denoted as U1–U5. Letting v_i' denote the derivative of the single variable function $v(x)$ at point x_i, these are given by

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

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

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

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

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

These discretizations may be constructed using standard polynomial interpolation techniques, as described in [84], for example.

The ODE system (2.2) is then discretized using either a pth-order explicit Runge-Kutta (ERK) method, or a pth-order, L-stable singly diagonally implicit Runge-Kutta (SDIRK) method, with the resulting scheme denoted as either ERKp+Up, or SDIRKp+Up. Specifically, the following ERK schemes of orders 1–5 are considered: The 1st-order scheme is Euler's method; the 2nd- and 3rd-order methods are the 'optimal' strong-stability-preserving schemes [52, (9.7), (9.8)]; the 4th-order scheme is the 'classical Runge-Kutta method' [14, p. 180]; and finally, see [14, (236a)] for the 5th-order scheme. The following SDIRK schemes of orders 1–4 are considered: The 1st-order scheme is Euler's method; the 2nd- and 3rd-order methods can be found in [14, pp. 261–262]; and the 4th-order scheme is given by [48, (6.16)]. For completeness, Butcher tableaux for these Runge-Kutta schemes can be found in Appendix A.1.

Upon application of a Runge-Kutta scheme to ODEs (2.2), their numerical solution may be written in the one-step form

$$\boldsymbol{u}_{n+1} = \Phi \boldsymbol{u}_n, \quad \boldsymbol{u}_0 = \boldsymbol{u}(0), \quad n = 0, \ldots, n_t - 1, \tag{2.3}$$

where $\Phi\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ is the (fine-grid) time-stepping operator. Note that equations (2.3) are equivalent to the block lower bidiagonal linear system given in (1.5). As will be prove important shortly, the eigenvalues of Φ can be computed as a function of those of $\mathcal{L}$ [28, 51]. In fact, it can be shown that Φ is a rational function (in a matrix sense) of $\mathcal{L}$, as in Lemma 2.1.

Lemma 2.1 (Rational form of Φ)**.** *Let $R(z) = P(z)/Q(z)$ denote the stability function [14, Lemma 351A] of a Runge-Kutta scheme applied to (2.2), 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 (2.3) is

$$\Phi(\Delta t\mathcal{L}) = P(\Delta t\mathcal{L})[Q(\Delta t\mathcal{L})]^{-1}. \tag{2.4}$$

Proof. Let $\mathcal{X}$ denote the matrix having eigenvectors of $\mathcal{L}$ as its columns. Substituting $\boldsymbol{u} = \mathcal{X}\boldsymbol{v}$ into (2.2) and left multiplying by $\mathcal{X}^{-1}$ yields a system of n_x decoupled ODEs of the form $\mathrm{d}v_k/\,\mathrm{d}t = \xi_k v_k$, $k = 1, \ldots, n_x$, in which ξ_k is the kth eigenvalue of $\mathcal{L}$. Using the Runge-Kutta stability function, the one-step numerical solution of the kth component of the decoupled ODEs is $v_k^{n+1} = R(\Delta t\xi_k)v_k^n \equiv P(\Delta t\xi_k)/Q(\Delta t\xi_k)v_k^n$. The one-step solution of the system of decoupled ODEs can then be written as $\boldsymbol{v}^{n+1} = P(\Delta t\,\mathrm{diag}(\boldsymbol{\xi}))[Q(\Delta t\,\mathrm{diag}(\boldsymbol{\xi}))]^{-1}\boldsymbol{v}^n$. Making the substitution $\boldsymbol{v} = \mathcal{X}^{-1}\boldsymbol{u}$, left multiplying by $\mathcal{X}$, and noting $\mathcal{X}G\left(\Delta t\,\mathrm{diag}(\boldsymbol{\xi})\right)\mathcal{X}^{-1} = G\left(\Delta t\mathcal{X}\,\mathrm{diag}(\boldsymbol{\xi})\mathcal{X}^{-1}\right) = G\left(\Delta t\mathcal{L}\right)$ for any rational function G yields the result. ■

For an ERK scheme, $Q(z) = 1$ and so the Runge-Kutta stability function used in Lemma 2.1 is simply a polynomial, $R(z) = P(z)$.

Corollary 2.2. *For periodic boundary conditions applied to (2.1), the time-stepping operator Φ in (2.3) 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 (2.4). ■

The CFL number for Runge-Kutta finite-difference discretizations of (2.1) is defined as

$$c := \alpha\frac{\Delta t}{\Delta x}. \tag{2.5}$$

The explicit discretizations considered here suffer from a CFL limit, for which a necessary (and sufficient) 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

TABLE 2.1: CFL limits $c_{\max}$ for ERK+U discretizations of constant-coefficient advection problem (2.1) 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...

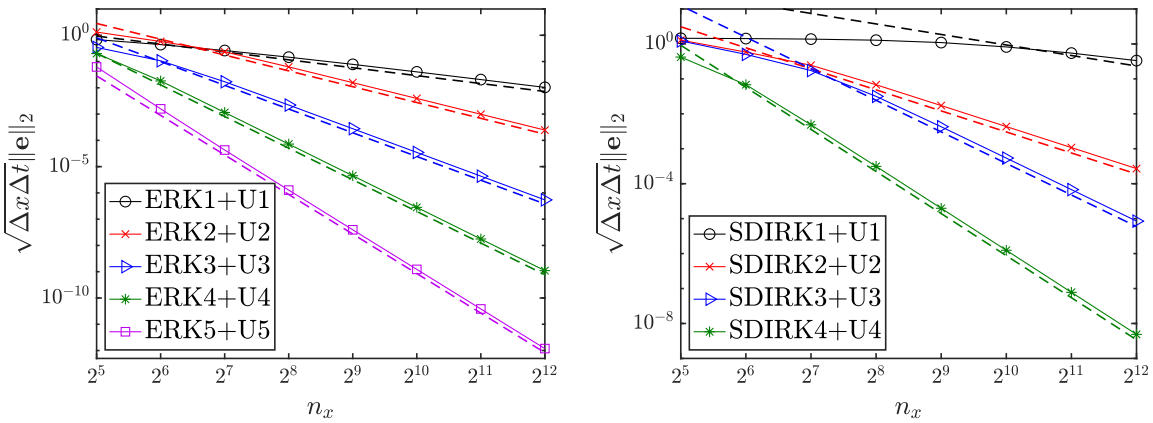

FIGURE 2.1: Space-time discretization errors for (2.1) measured in the discrete ℓ^2-norm. **Left:** ERK+U. **Right:** SDIRK+U. Plotted underneath each scheme's errors is a dashed line showing the theoretical convergence rate (order p for ERKp/SDIRKp+Up). Note the use of different scalings of the vertical axes in the plots.

given in Table 2.1. Throughout this chapter, 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 it can be shown that the eigenvalues of the circulant matrices $\mathcal{L} \in \mathbb{R}^{n_x \times n_x}$ lie in the closed left half plane, which means that the eigenvalues of $\Delta t\mathcal{L}$ 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 non-dissipative nature of (2.1), computed discretization errors are shown in Figure 2.1. 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.

2.2.2 Numerical set-up

For completeness, all of the settings used in the numerical tests in this chapter are now described. 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 behaviour. 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 only coarsening by a power of two is considered here). 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 (2.5) 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 for the model problem

To provide a baseline for the numerical results shown later in this chapter, we now present numerical results for model problem (2.1) using MGRIT with rediscretized 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 (2.1), such a coarse-grid operator leads to divergent solvers for all m. This behaviour 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. See the left panel of Figure 2.2 for a specific example.

To overcome the instability of rediscretizing an explicit method on the coarse grid, 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

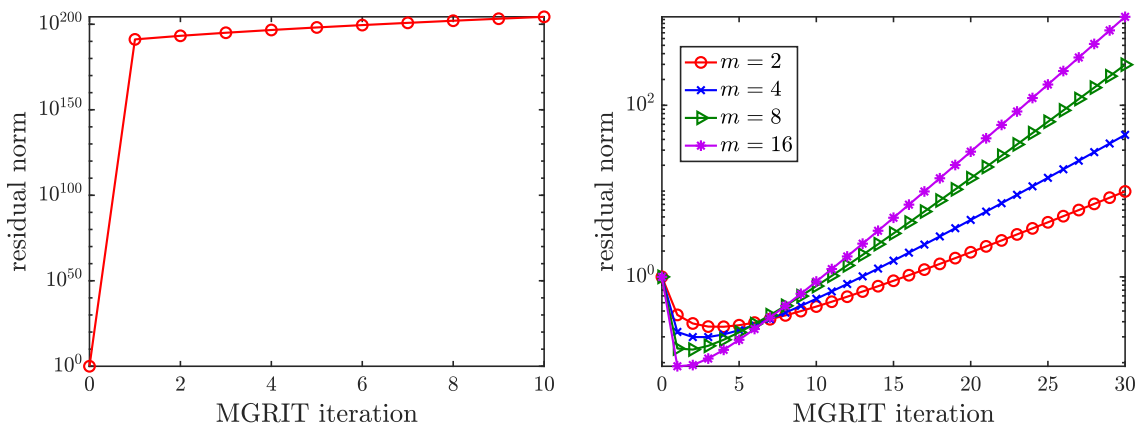

FIGURE 2.2: Two-norm of space-time residual (relative to its initial value) as a function of MGRIT iteration for the fine-grid operator Φ = ERK1+U1. **Left:** Coarse-grid operator is the unstable rediscretization Ψ = ERK1+U1 with coarsening factor $m = 2$. The problem size is $n_x \times n_t = 2^8 \times 2^{10}$. **Right:** Coarse-grid operator is the stable implicit discretization Ψ = SDIRK1+U1 for coarsening factors $m = 2, 4, 8, 16$. The problem size is $n_x \times n_t = 2^{10} \times 2^{12}$.

tests, 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. See the examples in the right panel of Figure 2.2. 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 [58], or reducing the order of the discretization in time [69] 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 [58, 69]. Thus, we do not present numerical results for ERK discretizations here because the standard choice of rediscretization 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 rediscretized on coarse grids to provide stable coarse-grid operators. Two-level MGRIT iteration counts for SDIRK+U discretizations of (2.1) using such coarse-grid operators are given in the left side of Table 2.2. 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 2.1), but it still requires a number of iterations that is much higher than what we will achieve with the new approach introduced in this chapter.

TABLE 2.2: Two-level iteration counts for SDIRK+U discretizations using a rediscretized coarse-grid operator. **Left:** Measured iteration counts. **Right:** Iteration counts at which the exact solution is achieved using exact arithmetic, $n_t/(2m)$. The '✗' 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
	$2^{12} \times 2^{12}$	✗	520	1024	512

When using rediscretization, MGRIT convergence rates for implicit discretizations of hyperbolic problems strongly depend on the CFL number, with smaller CFL numbers typically resulting in faster convergence, even though there is no CFL limit to violate. This can be seen by contrasting the types of convergence rates reported in [28] for linear advection with those shown in Table 2.2 for larger, more realistic CFL numbers. This behaviour 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 [28, 32]; see also our example in Figure 1.4 for the diffusion equation.

2.4 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, let us now recall the two-level MGRIT convergence theory of [28, 51] and discuss some of its implications.

2.4.1 Two-level convergence theory

The convergence behaviour of MGRIT can be understood by analyzing its error propagation matrix, $\mathcal{E}$. That is, if $\boldsymbol{e}^{(0)}$ is the initial space-time error, then after q MGRIT iterations, the error obeys $\left\|\boldsymbol{e}^{(q)}\right\| \leq \|\mathcal{E}\|^q \left\|\boldsymbol{e}^{(0)}\right\|$. To analyze $\|\mathcal{E}\|$, let us assume that the fine-grid time-stepping operator Φ and coarse-grid time-stepping operator 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 that $|\lambda_k| = |\mu_k| = 1$ is also sufficient for stability, but the following theoretical result does not apply to such cases.

The assumption of diagonalizability allows error reduction for each spatial mode to be considered individually. That is, if $\mathcal{E}_k$ is the error propagator associated with the kth spatial mode, then we can consider $\|\mathcal{E}_k\|$ for each k. Moreover, $\|\mathcal{E}\|_2 = \max_k \|\mathcal{E}_k\|_2$. For the case of FCF-relaxation used here, the 2-norm of $\mathcal{E}_k$ can be bounded as [51, Lemma 4.1, Theorem 4.3], [28, Theorem 3.3]:

$$\|\mathcal{E}_k\|_2 \leq \sqrt{m}\, |\lambda_k|^m \frac{|\lambda_k^m - \mu_k|}{1 - |\mu_k|} \left(1 - |\mu_k|^{n_t/m-1}\right). \tag{2.6}$$

In [88], it was shown that this bound is equal to $\|\mathcal{E}_k\|_2$ up to $\mathcal{O}(m/n_t)$, and, so, it is sharp.

Given bound (2.6), 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 $1 - |\mu_k|^{n_t/m-1} \to 1$ as $n_t \to \infty$, meaning convergence is primarily determined by the preceding factors. Firstly, convergence of the kth 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 in [88], the largest (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 [28, 51]. In summary, fast convergence necessitates the approximation $\mu_k \approx \lambda_k^m$ to hold $\forall k$, and with increasing accuracy for $|\lambda_k|, |\mu_k| \to 1$.

2.4.2 Implications of convergence theory

Let us 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 kth 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 behaviour is seen in the top row of Figure 2.3, 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.

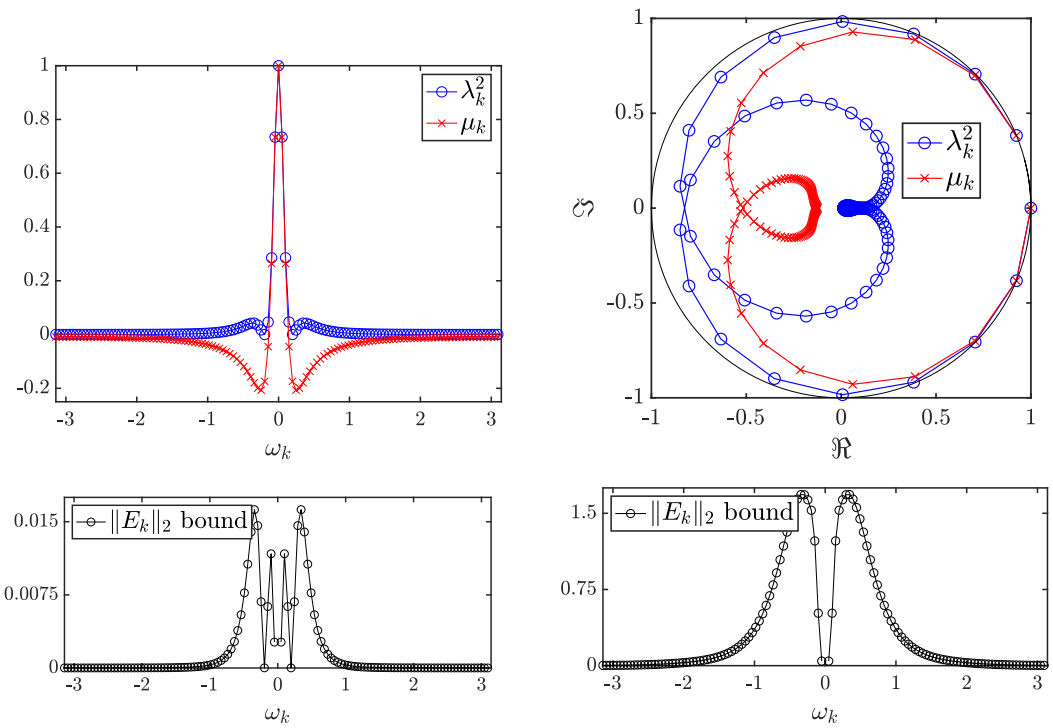

FIGURE 2.3: **Left column:** Diffusion equation $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ discretized with SDIRK2 in time and 2nd-order central finite-differences in space. **Right column:** Advection equation $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$ 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 (2.6) for each problem as a function of Fourier frequency, ω_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 $\Phi, \Psi \in \mathbb{R}^{128\times 128}$.

The properties just discussed, in conjunction with the plots in Figure 2.3, help to illuminate why rediscretization 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 2.3, notice the largest eigenvalues of Φ^2 are clustered around spatial frequency $\omega_k = 0$, noting that the $\omega_k = 0$ 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, $\omega_k \approx 0$. For the diffusion problem, this approximation is adequate to obtain fast convergence as the decay of $|\lambda_k|^2$ away from $\omega_k = 0$ is very rapid. However, for the advection problem, the approximation is inadequate since the decay of $|\lambda_k|^2$ away from $\omega_k = 0$ is more gradual and, so, the mismatch between λ_k^2 and μ_k is of much greater detriment for convergence. Error bounds (2.6) are shown in Figure 2.3 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, rediscretization 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 [108] 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 identified above actually correspond to space-time characteristic components, and this will be investigated further in Chapter 3. In any event, it is likely that some of the ideas proposed in [108] for improving spatial multigrid solvers will be useful for developing improved MGRIT solvers.

2.5 Coarse-grid operators based on a linear approximation of Ψ_{ideal}

From the discussion surrounding error estimates (2.6), 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, let us consider Ψ as the solution of the minimization problem

$$\Psi := \underset{\widehat{\Psi}\in\mathbb{R}^{n_x\times n_x}}{\arg\min} \left\| W_{\boldsymbol{\lambda}}^{1/2}\left[\boldsymbol{\lambda}^m - \boldsymbol{\mu}\big(\widehat{\Psi}\big)\right]\right\|_2^2, \tag{2.7}$$

where $\boldsymbol{\lambda} = \big(\lambda_1,\ldots,\lambda_{n_x}\big)^\top$, $\boldsymbol{\mu} = \big(\mu_1,\ldots,\mu_{n_x}\big)^\top \in \mathbb{C}^{n_x}$, and $(\boldsymbol{\lambda}^m)_k \equiv \lambda_k^m$. Here, $W_{\boldsymbol{\lambda}} := \operatorname{diag}(\boldsymbol{w}) \in \mathbb{R}^{n_x\times n_x}$ is a diagonal weighting matrix, whose kth entry is $w_k := w(|\lambda_k|)$, in which $w\colon \mathbb{R}_+ \to \mathbb{R}_+$ is a weighting function reflecting the heuristic that it is most important to minimize $\lambda_k^m - \mu_k$ for $|\lambda_k|, |\mu_k| \to 1$, and less important for $|\lambda_k|, |\mu_k| \ll 1$. One choice of weighting function 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; in the numerical results shown here, $\epsilon = 10^{-6}$. Note that allowing a free choice for $\widehat{\Psi}$ would naturally result in the choice $\Psi = \Phi^m$ and, so, the optimization in (2.7) is constrained by a pre-specified sparsity pattern of Ψ.

In general the eigenvalues of a matrix Ψ depend nonlinearly on its entries and thus (2.7) constitutes a nonlinear minimization problem. However, for the special case of circulant Ψ, (2.7) 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 (2.1) 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 (see Corollary 2.2). In Section 2.5.6, we will also show that a sparse Ψ may be used with implicit temporal discretizations of (2.1), for which Φ and Φ^m are dense matrices. Thus, for the remainder of this chapter, let us focus exclusively on the case in which Ψ is a sparse circulant matrix such that (2.7) 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 (2.6) than the heuristic-based (2.7). This more elaborate approach, however, gives similar results for the simple model problem considered here. For completeness, this approach is given in Appendix A.2.

2.5.1 Linear least squares formulation

Let $\tilde{\boldsymbol{\phi}}^m, \tilde{\boldsymbol{\psi}} \in \mathbb{R}^{n_x}$ denote the first columns of the circulant matrices Φ^m and Ψ, respectively, and recall that a circulant matrix can be fully specified by its first column. Assuming the sparsity pattern of Ψ is given, let $R \in \mathbb{R}^{\nu \times n_x}$ be the restriction operator that selects these ν non-zero entries from $\tilde{\boldsymbol{\psi}}$, 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 non-zero components of $\tilde{\boldsymbol{\psi}}$, which are denoted as $\boldsymbol{\psi} := R\tilde{\boldsymbol{\psi}} \in \mathbb{R}^{\nu}$. Finally, let $\mathcal{F} \in \mathbb{C}^{n_x \times n_x}$ be the DFT matrix, then by the properties of circulant matrices, $\boldsymbol{\lambda}^m = \mathcal{F}\tilde{\boldsymbol{\phi}}^m$, and $\boldsymbol{\mu} = \mathcal{F}\tilde{\boldsymbol{\psi}} = \mathcal{F}R^\top\boldsymbol{\psi}$ since $R^\top R$ has ones on the diagonal in rows where $\tilde{\boldsymbol{\psi}}$ has non-zeros and zeros everywhere else. Thus, (2.7) can be written as a linear least squares problem for the non-zero entries in the first column of Ψ:

$$\boldsymbol{\psi} := \underset{\hat{\boldsymbol{\psi}} \in \mathbb{R}^{\nu}}{\arg\min} \left\| W_{\boldsymbol{\lambda}}^{1/2} \mathcal{F} \left(\tilde{\boldsymbol{\phi}}^m - R^\top \hat{\boldsymbol{\psi}} \right) \right\|_2^2 . \tag{2.8}$$

Remark 2.3 (The coarse-grid operator $\Psi = \underset{\widehat{\Psi} \in \mathbb{R}^{n_x \times n_x}}{\arg\min} \|\Phi^m - \widehat{\Psi}\|_2^2$ does not result in a good solver)**.** *For weighting function $w = 1$, or $W_{\boldsymbol{\lambda}} = I$, (2.8) 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 (2.8) can be computed explicitly as $\boldsymbol{\psi} = R\tilde{\boldsymbol{\phi}}^m$, 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 (2.1) because it does not adequately capture the dominant eigenvalues of Φ^m (see Section 2.4).*

Lemma 2.4. *The solution of* (2.8) *is real valued.*

Proof. The normal equations of (2.8) are

$$\big(R\mathcal{F}^* W_{\boldsymbol{\lambda}} \mathcal{F} R^\top\big)\boldsymbol{\psi} = \big(R\mathcal{F}^* W_{\boldsymbol{\lambda}} \mathcal{F}\big)\tilde{\boldsymbol{\phi}}^m. \tag{2.9}$$

Since R and $\tilde{\boldsymbol{\phi}}^m$ are real, $\boldsymbol{\psi}$ is real if the circulant matrix $\mathcal{A} := \mathcal{F}^* W_{\boldsymbol{\lambda}} \mathcal{F} = \mathcal{F}^* \operatorname{diag}(\boldsymbol{w})\mathcal{F}$ is real. Letting $\tilde{\boldsymbol{a}}$ denote the 1st column of $\mathcal{A}$, then, because $\mathcal{A}$ is circulant, $\tilde{\boldsymbol{a}} = \mathcal{F}^*\boldsymbol{w}$; that is, $\tilde{\boldsymbol{a}}$ is the inverse DFT of $\boldsymbol{w}$. Appealing to properties of the inverse DFT (see Figure 2.4), since $\boldsymbol{w}$ is real, $\tilde{\boldsymbol{a}}$ will be real if $\boldsymbol{w}$ 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 $\boldsymbol{w}$ 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 (2.8) is found to have some small imaginary components since $\mathcal{F}$ is complex and the problem is ill-conditioned. These are simply truncated from the solution, as is justified by Lemma 2.4. 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 2.3. This is also observed for some other choices of the weight matrix W leading to particularly ill-conditioned matrices in (2.8). In practice, if an imaginary component larger than 10^{-8} is detected, the result is flagged and the resulting Ψ is not accepted as a coarse-grid operator. Note, however, that such large imaginary components do not occur for the sparsity patterns of Ψ that we advocate in the following sections.

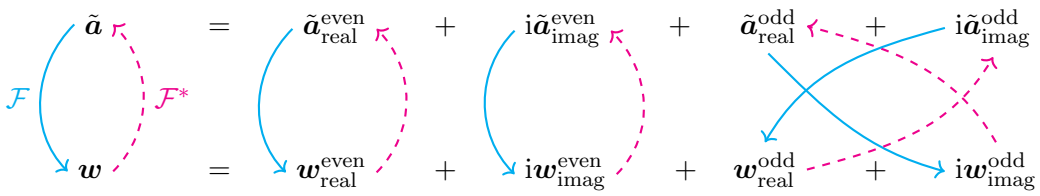

FIGURE 2.4: How the even, odd, real, and imaginary components of the vectors $\boldsymbol{a}$ and $\boldsymbol{w}$ are related through the discrete Fourier transform $\mathcal{F}$ (solid, cyan lines) and its inverse $\mathcal{F}^*$ (dashed, magenta lines) when $\mathcal{F}\tilde{\boldsymbol{a}} = \boldsymbol{w}$ or $\mathcal{F}^*\boldsymbol{w} = \tilde{\boldsymbol{a}}$.

2.5.2 Explicit schemes: Selection of Ψ's non-zero pattern

Before solving (2.8), it must first be decided how to constrain the non-zero 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 work on the finest level. The operator complexity depends on the sparsity of the coarse-grid operators Ψ. Clearly, it is required that Ψ be significantly sparser than the ideal coarse-grid operator Φ^m, so that time-stepping on the coarse grid is substantially less expensive than on the fine grid. Ideally, one 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 (note that the ideal coarse operator Φ^m is typically much denser than Φ), but the forthcoming numerical results will show that constraining Ψ to have as few non-zeros as Φ does not yield good solvers in general. Still, it is useful to consider the case where $\texttt{nnz}(\Psi) = \texttt{nnz}(\Phi)$, with $\texttt{nnz}(A)$ denoting the number of non-zeros of matrix A, as a reference case to 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 used thus far in this chapter. Now, assuming Φ_ℓ is a sparse operator, the work required to time-step with it is proportional to $\texttt{nnz}(\Phi_\ell) n_t m^{1-\ell}$, assuming a constant coarsening factor of m on all levels. Thus, the operator complexity is given by

$$\text{operator complexity} := \frac{1}{\texttt{nnz}(\Phi_1)} \sum_{\ell=1}^{L} m^{1-\ell} \texttt{nnz}(\Phi_\ell). \tag{2.10}$$

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 $\texttt{nnz}(\Psi) = \texttt{nnz}(\Phi)$, 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 $1 + 1/(m-1)$. Operator complexities of the operators derived in the following sections will be compared with these reference complexities.

Next, let us discuss how to choose the locations of the non-zeros in Ψ. Note that simply rediscretizing Φ on a temporally coarsened mesh leads to Ψ having the same non-zero pattern as Φ. To motivate a better choice of sparsity pattern, consider the effects of

temporal coarsening on the exact solution of (2.1) when it is sampled on a space-time mesh; a schematic diagram of this example is shown in Figure 2.5. The solution of a hyperbolic PDE is propagated through space-time along its characteristics, $x(t)$. Advection problem (2.1) simply has characteristics that are straight lines with slope $\mathrm{d}x/\,\mathrm{d}t = \alpha$. Now, say one has 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 $\Psi_{\text{exact}} = \Phi^4_{\text{exact}}$. By definition, Ψ_{exact} propagates the solution forward in time by a distance of $4\Delta t$; however, observe also that it 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 non-zeros of Φ^m. To assess this, let us compute Φ^m and examine its non-zeros as a function of their diagonal index i (recall entries of Φ^m are constant along its diagonals since it is circulant). 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 2.6. There is clearly a well-defined distribution in the magnitude of these non-zeros for each scheme. The distributions peak at different i essentially because the time step is chosen differently for each scheme, since $c = 0.85c_{\max}$ and $c_{\max}$ is different for each scheme (see Table 2.1). 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 the space-time point $(x, t) = (x_i, t^{n+m})$. This illustrates, not unexpectedly,

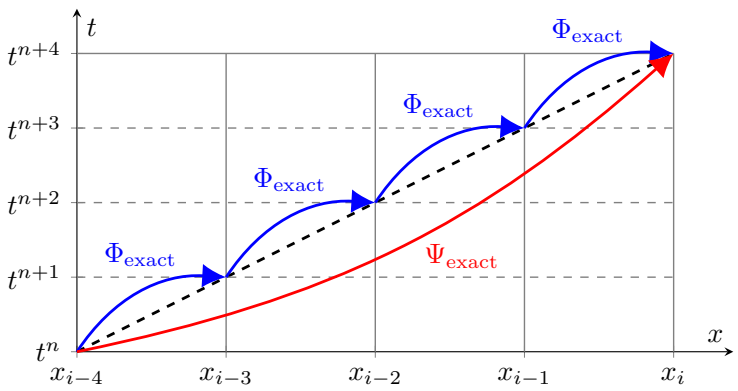

FIGURE 2.5: Exact fine- and coarse-grid time-stepping operators, Φ_{exact} and Ψ_{exact}, propagate the solution of $\frac{\partial u}{\partial t} + \alpha\frac{\partial u}{\partial x} = 0$ along one of its characteristics (thick, dashed line) on a fine and coarse grid, respectively. The fine grid has 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}.

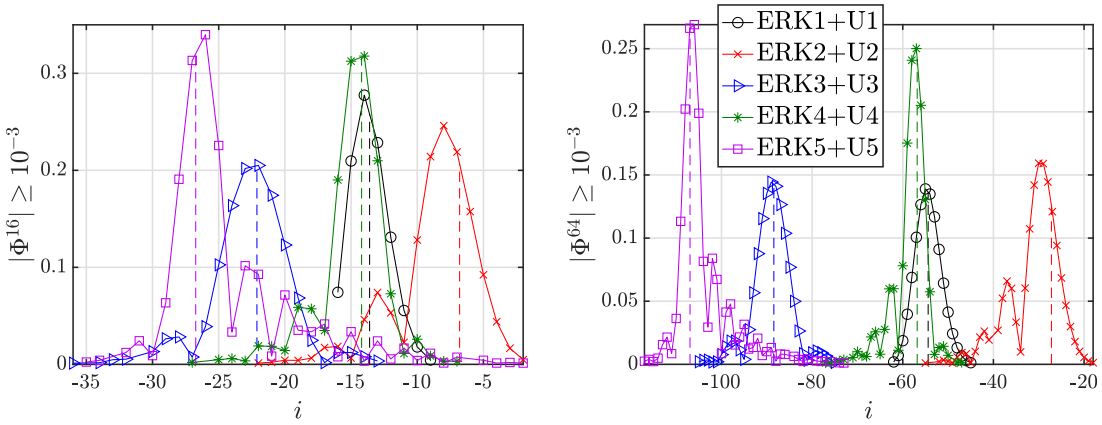

FIGURE 2.6: Magnitude of diagonal entries of $\Psi_{\text{ideal}} := \Phi^m$, $m = 16$ (**left**), and $m = 64$ (**right**), that are larger than 10^{-3}, as a function of their diagonal index, i. Fine-level discretizations, Φ, are ERKp+Up, $p \in \{1, 2, 3, 4, 5\}$. Dashed vertical lines for each discretization are included to represent a distance of $-mc\Delta x$ from $i = 0$. Note that $c = 0.85c_{\max}$ is different for each scheme.

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 2.5, it is clear that the non-zero 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 2.6). Note that in [40] it was also argued that making use of characteristic information may be important for parallel-in-time methods. The specific sparsity patterns used for the ERK+U schemes will be discussed further in Section 2.5.3.

2.5.3 Explicit schemes: Two-level results

In this section, two-level MGRIT results with Ψ as the solution of least squares problem (2.8) for ERK discretizations are considered. To demonstrate the validity of the ideas outlined in the previous section, the least squares problem is solved 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. Numerical results will first be compared for these two approaches, and then details will be given about how the sparsity pattern based on the largest nonzeros of Φ^m is chosen.

Solver iteration counts for Ψ having the same sparsity pattern as Φ are shown in the left side of Table 2.3. A convergent solver was not obtained for any m for the 1st-order scheme, convergent solvers were obtained only for $m = 2$ for the 2nd- and 3rd-order schemes, and convergent solvers with $m \in \{2, 4, 8\}$ were obtained for the 4th- and 5th-order schemes. In all cases where convergent solvers were found, the iteration counts

TABLE 2.3: Two-level iteration counts for ERK+U discretizations with Ψ as linear least squares solution (2.8). **Left:** Sparsity pattern of Ψ is equal to that of Φ. **Right:** Sparsity pattern of Ψ is based on that of Φ^m. An '✗' 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 '✗*' 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 (Φ-based sparsity)						m (Φ^m-based sparsity)					
		2	4	8	16	32	64	2	4	8	16	32	64
ERK1+U1	$2^8 \times 2^{10}$	✗	✗	✗	✗	✗	✗	11	6	6	7	6	5
	$2^{10} \times 2^{12}$	✗	✗	✗	✗	✗	✗	11	6	6	7	6	5
	$2^{12} \times 2^{14}$	✗	✗	✗	✗	✗	✗	11	6	6	7	6	5
ERK2+U2	$2^8 \times 2^{11}$	10	✗	✗	✗	✗	✗	10	7	9	8	7	7
	$2^{10} \times 2^{13}$	10	✗	✗	✗	✗	✗	10	7	9	8	7	7
	$2^{12} \times 2^{15}$	10	✗	✗	✗	✗	✗	10	7	9	8	7	7
ERK3+U3	$2^8 \times 2^9$	9	✗	✗*	✗*	✗*	✗*	7	6	5	6	5	3
	$2^{10} \times 2^{11}$	9	✗	✗*	✗*	✗*	✗*	7	6	5	6	5	4
	$2^{12} \times 2^{13}$	9	✗	✗*	✗*	✗*	✗*	7	6	5	6	5	4
ERK4+U4	$2^8 \times 2^{10}$	6	4	8	✗	✗*	✗*	5	4	4	4	5	5
	$2^{10} \times 2^{12}$	6	4	8	✗	✗*	✗*	5	4	4	4	5	6
	$2^{12} \times 2^{14}$	6	4	8	✗	✗*	✗*	5	4	4	4	5	6
ERK5+U5	$2^8 \times 2^9$	3	3	7	✗*	✗*	✗*	3	3	3	4	4	3
	$2^{10} \times 2^{11}$	3	3	7	✗*	✗*	✗*	3	3	3	4	5	4
	$2^{12} \times 2^{13}$	3	3	7	✗*	✗*	✗*	3	3	3	4	5	4

remain constant as the mesh is refined. For the cases where the solvers converge, these results are certainly an improvement on those using rediscretization, 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 2.3. 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 non-zeros in Φ correspond primarily to the largest non-zeros in Φ^m for the cases in which the Φ-based sparsity patterns yield convergent solvers. Furthermore, least squares problem (2.8) was severely ill-conditioned for many instances in which Ψ and Φ shared a sparsity pattern (see Table 2.3), 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.

Let us now explain in detail how the sparsity patterns were chosen that lead to the results in the right-hand side of Table 2.3, 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 non-zeros in Ψ_{ideal} (as in Figure 2.6, for example). As

a first approximation, we choose a contiguous subset of the locations of the largest $\texttt{nnz}(\Phi)$ non-zeros of Φ^m (even if the locations of the largest non-zeros 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 non-zero 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 non-zeros (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 non-zeros. As an example, the left panel of Figure 2.7 shows the non-zero 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 2.7 also shows (right panel), for each discretization, the operator complexities (2.10) of the resulting solvers along with the operator complexity of $1 + 1/m$ that results when $\texttt{nnz}(\Psi) = \texttt{nnz}(\Phi)$ in a two-level method (see Section 2.5.2).

We find that, in general, to obtain convergent and scalable solvers there has to be a slight increase in the number of non-zeros in Ψ as the coarsening factor is increased, as can be seen for ERK3+U3 in Figure 2.7 (left panel), for example. This behaviour appears consistent with that seen in [108] for the geometric 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 non-zeros required is smaller for higher-order discretizations, as seen in the right panel of Figure 2.7, where operator complexities tend to be smaller for higher-order methods. Notice that ERK4+U4

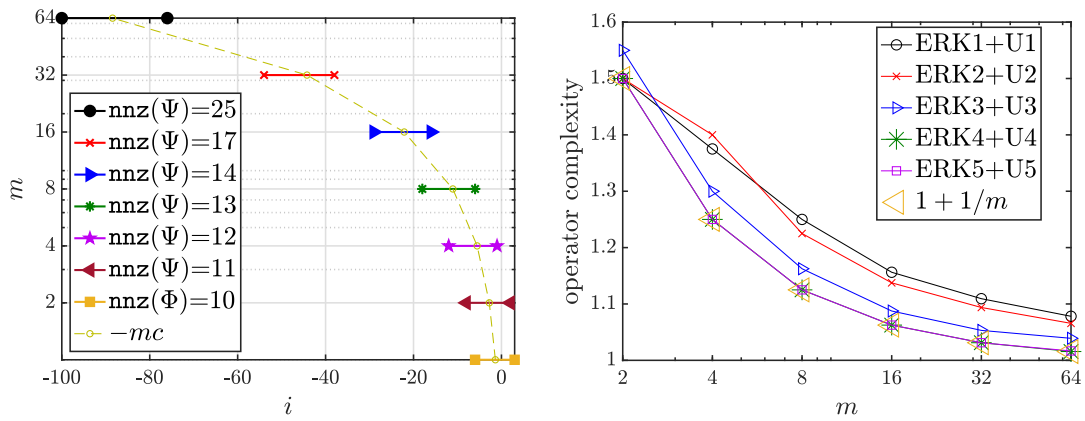

FIGURE 2.7: Two-level solvers for $m \in \{2, 4, 8, 16, 32, 64\}$ with Ψ as linear least squares solution (2.8). **Left:** Sparsity patterns chosen for Ψ for ERK3+U3, as represented by the non-zero diagonal indices i for each value of m. Also plotted is $-mc$, which represents the characteristic departure point on a grid using a time-step of $m\Delta t$. **Right:** Operator complexities (2.10) for all discretizations; shown also is the reference operator complexity of $1 + 1/m$.

and ERK5+U5 essentially have operator complexities of $1 + 1/m$ since very few (if any) additional non-zeros were needed.

The results at the right of Table 2.3 show that it is possible to overcome the CFL instability that arises from rediscretizing 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 2.3 are comparable to those for model diffusion problems using rediscretized coarse-grid operators [28, 32]. 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 order. When combined with the trend in Figure 2.7 (right panel) that operator complexities are smaller for higher-order discretizations, this suggests that higher-order discretizations of model problem (2.1) 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 2.8. 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 behaviour, it is unsurprising that the solver converges quickly, and that the associated error bounds are small (bottom right

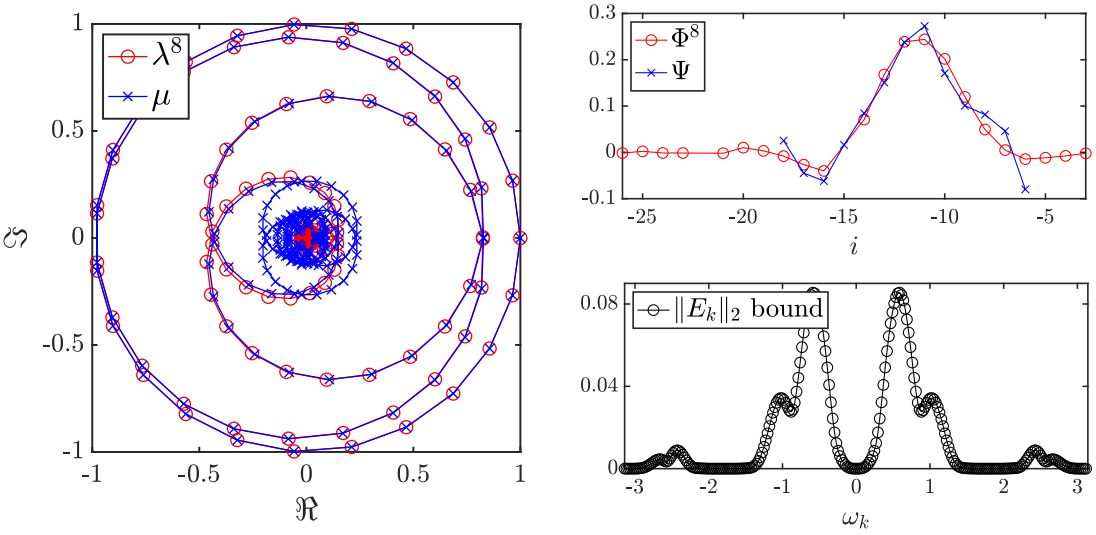

FIGURE 2.8: Linear least squares solution (2.8) for ERK3+U3 with $n_x \times n_t = 2^8 \times 2^9$, coarsening factor $m = 8$, and the sparsity pattern of the least-squares determined Ψ based on that of $\Psi_{\text{ideal}} = \Phi^8$. **Left:** Eigenvalues λ^8 of Φ^8, and μ of Ψ. **Top right:** Entries 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 (2.6) as a function of spatial Fourier frequency.

panel of Figure 2.8). The entries of the least squares Ψ (upper right panel of Figure 2.8) are clearly correlated with those of the ideal operator.

Remark 2.5 (Relation between Ψ and semi-Lagrangian discretizations)**.** *It is clear that the optimized coarse operators Ψ with Φ^m-based sparsity have a non-local 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 2.7), Ψ clearly does not just represent a particular semi-Lagrangian discretization of the PDE on coarse grids. Semi-Lagrangian coarse-grid operators will be given further consideration in Chapters 3 and 4.*

2.5.4 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.

Convergence theory of multilevel MGRIT is significantly more complicated than in the two-level setting [51] 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 (2.8). As previously, sparsity patterns of coarse-grid operators are selected by roughly choosing some subset of the locations of the largest non-zeros 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, only results for ERKp+Up, $p \in \{1, 3, 5\}$ are shown. 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 2.6). 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 2.4 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 non-zeros 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 2.7), 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, [58] 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

TABLE 2.4: Multilevel iteration counts as a function of number of grid levels for V-cycles with $m = 4$ coarsening on each level. Operator complexities (OC) (2.10) are also given for each discretization. Note the reference operator complexity for a multilevel method is bounded above by $1 + 1/(m-1) = 1.33\ldots$ for $m = 4$ (see Section 2.5.2). 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).

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	–

was slow, with on the order of 40 iterations required to reach convergence for the mesh sizes considered here.

2.5.5 Explicit schemes: Application to inflow/outflow boundaries

Discretizations of (2.1) subject to inflow/outflow spatial boundary conditions result in non-circulant 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 non-normal and likely not even diagonalizable, and, so, the rigorous optimization of the corresponding Ψ would require more sophisticated convergence theory than in [28, 51] such as that in [88], for example, and would certainly be highly nonlinear.

In the spirit of local Fourier analysis of multigrid methods (see, e.g., [99]) 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 non-zero 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. 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 [88]. 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 [18], 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 (see [52, 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 (see Table 2.1).

TABLE 2.5: Iteration counts for ERK+U discretizations of (2.1) with inflow/outflow boundaries; Ψ is given by truncating the circulant matrix resulting from linear least squares solution (2.8), 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).

Scheme	$n_x \times n_t$	Two level, m						Multilevel, ℓ					
		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	–

Iteration counts for the inflow/outflow problem discretized with ERKp+Up, $p \in \{1, 3, 5\}$, are given in Table 2.5. 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 2.3 for two level, and Table 2.4 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 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.

2.5.6 Implicit schemes

We now consider linear least squares problem (2.8) for the computation of suitable coarse-grid operators Ψ for SDIRK+U discretizations of (2.1). 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 2.9 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 non-zeros 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 non-dissipative nature of (2.1) well. Indeed, the plots in Figure 2.9 are qualitatively similar to their analogues for the ERK schemes in Figure 2.6 (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 2.9—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, as in the example shown in the right panel of Figure 2.2 in Section 2.3. 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.

Let us 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 non-zeros in Ψ. To do so, we compute the entries in the 1st column of Φ^m (this can be done using the FFT and its inverse), and then we select a non-zero pattern using thresholding. That is, recalling $\tilde{\boldsymbol{\phi}}^m$

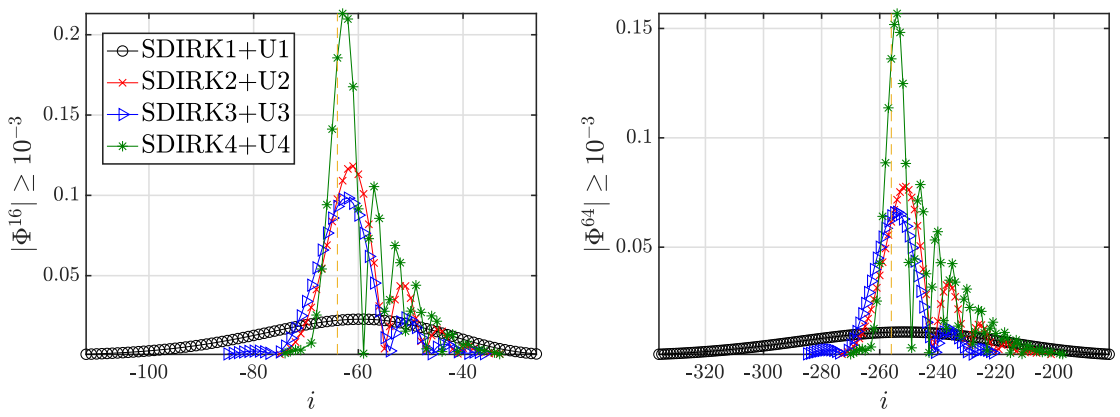

FIGURE 2.9: Magnitude of diagonal entries of the dense matrices $\Psi_{\text{ideal}} := \Phi^m$, $m = 16$ (**left**), and $m = 64$ (**right**), that are larger than 10^{-3}, as a function of their diagonal index, i. Fine-level discretizations, Φ, are SDIRKp+Up, $p \in \{1, 2, 3, 4\}$. A value of $n_x = 2^{10}$ 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 $c = 4$).

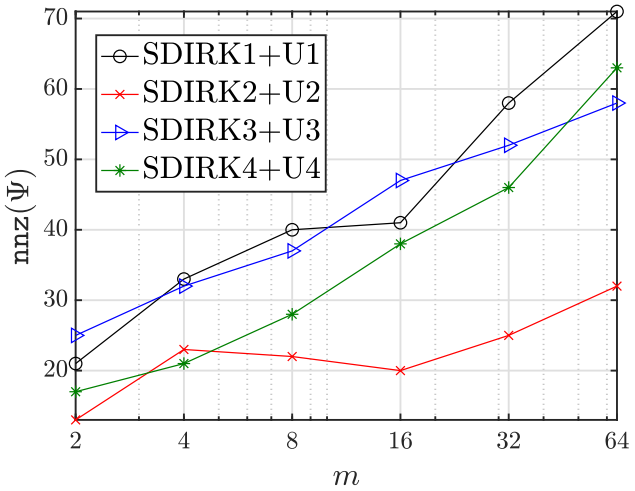

FIGURE 2.10: Number of non-zeros per row of Ψ for SDIRK+U discretizations as a function of coarsening factor, m.

is the (dense) first column of Φ^m, we take the non-zero 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 schemes, we have loosely tried to achieve some balance between the rate of convergence and the number of non-zeros 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 pth-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 2.9. Figure 2.10 shows the number of non-zeros 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 non-zeros in Φ^m whose magnitude is significant (Figure 2.9).

The iteration counts for the solvers are shown in Table 2.6. 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 2.2 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 (2.1) 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 2.6: Two-level iteration counts for SDIRK+U discretizations with Ψ given as linear least squares solution (2.8).

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

2.6 Parallel results

In this section, strong parallel scaling results for the ERKp+Up, with $p \in \{1, 3, 5\}$, are given for the solvers developed in the previous sections. Specifically, multilevel results are given in Section 2.6.1, and, for completeness, two-level results have can be found in Appendix A.3.

The implementations use the open-source package XBraid [107]. The results were generated on Quartz, a Linux cluster at Lawrence Livermore National Laboratory consisting of 2,688 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 2.3 in the two level case, and in the multilevel case it is the largest problem show in Table 2.4 and the number of levels in the solver is taken as the maximum shown in the 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 chapter, a stopping criterion based on achieving a space-time residual below 10^{-10} in the discrete ℓ^2-norm is used, but a stopping criterion based on achieving discretization error accuracy is also considered.

2.6.1 Parallel results: Multilevel solvers

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 2.11. The plots show good

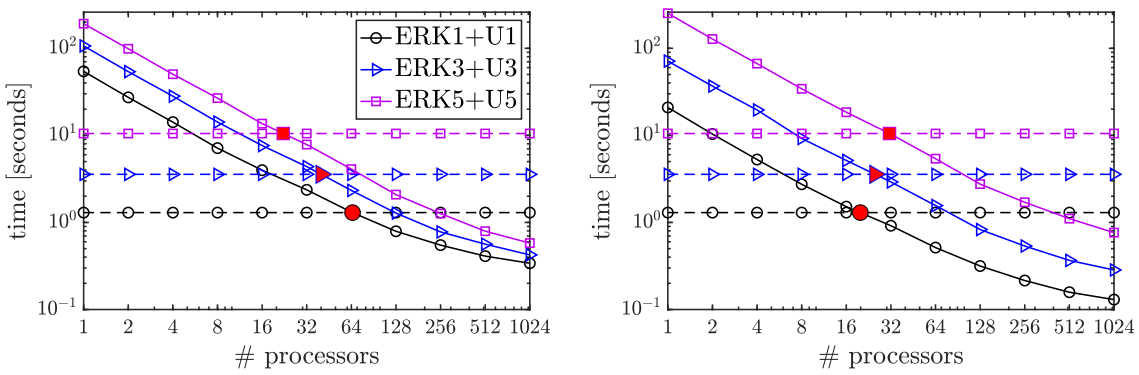

FIGURE 2.11: Strong parallel scaling: Runtimes of MGRIT V-cycles with $m = 4$ coarsening and using time-only parallelism for ERKp+Up discretizations on space-time grids of size $n_x \times n_t = 2^{12} \times (2^{14}, 2^{13}, 2^{13})$ for $p = (1, 3, 5)$. **Left:** Fixed residual stopping tolerance of 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.

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 [33]. 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 5th-order discretization run at a realistic CFL fraction is significant.

2.7 Conclusions

In this chapter, we have considered the parallel-in-time integration of the one-dimensional, constant-coefficient linear advection problem using the MGRIT algorithm. 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.

In Section 2.3, we showed several motivating examples that demonstrate the difficulty this problem poses for these parallel-in-time solvers when using a rediscretized coarse-grid operator. In Section 2.4, we used existing convergence theory to explain why this problem is so difficult, and what is required of coarse-grid operators for its efficient solution. In particular, convergence hinges on the coarse-grid operator accurately propagating spatial modes that decay slowly in time very similarly to that of the ideal coarse-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.

In Section 2.5 we develop 'near-optimal' coarse-grid operators for this PDE 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 MGRIT 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 achieved through 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, parallel results were given in Section 2.6, which showed that significant speed-ups over sequential time integration are possible when using our optimized coarse-grid operators.

The optimization approaches presented in this chapter rely on calculations that are feasible only 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 provide powerful tools to demonstrate that, for this canonical hyperbolic PDE, it is possible to obtain highly efficient MGRIT and Parareal solvers. However, the heuristics developed here regarding effective coarse-grid operators apply more generally than just to the constant-coefficient problem, and these principles will be made use of in future chapters of this thesis.

Chapter 3

Closed-form Fourier analysis of MGRIT with applications to advection-dominated problems

3.1 Introduction and outline

The previous chapter demonstrated that MGRIT diverges when applied to constant-coefficient linear advection when the coarse-grid operator is based on the standard approach of rediscretization. More broadly, poor convergence on a wide variety of advection-dominated problems has been reported throughout the MGRIT and Parareal literature [19, 23, 28, 46, 40, 50, 58, 57, 69, 81, 80, 83, 97]. There has been a large number of convergence theories produced for MGRIT and Parareal [26, 28, 46, 40, 51, 80, 88, 44, 38], with several paying particular attention to advection-related problems [26, 80, 40, 46]. Despite this, there is not yet a widely accepted explanation for what fundamentally makes the parallel-in-time solution of these problems so difficult.

Recall from Section 2.4 that we considered the MGRIT convergence theory from [28] in the context of advection-dominated problems. For fast convergence, we argued that not only should the coarse-grid operator propagate spatial modes similarly to the ideal coarse-grid operator in general, but it should do so with increased accuracy for modes which decay slowly in time. We further argued that an increase in the presence of slowly decaying spatial modes, coupled with the fact that rediscretization does not provide a particularly accurate approximation to such modes, is the reason *why* convergence for advection-dominated problems is poor compared to diffusion-dominated problems. While our arguments ultimately led to an optimization-based strategy for designing coarse-grid

operators that yielded very fast convergence, our arguments were not fully rigorous. Moreover, it is not clear to what extent they generalize to more difficult problems. One outcome of this chapter is to provide a complementary explanation for these convergence issues that is more general, and, in doing so, make connections to similar issues that plague spatial multigrid solvers when applied to steady state advection-dominated problems.

More broadly, this chapter analyses the convergence of two-level MGRIT through the lens of local Fourier analysis (LFA). LFA is the most widely used and successful tool for investigating the convergence behaviour of multigrid methods (see, e.g., [9, 10, 98, 100, 108, 13, 99, 105]). While LFA has previously been used to investigate MGRIT convergence [38, 26], and for multigrid-in-time methods more broadly [100, 39, 45], what makes our application of it to MGRIT novel is that our theory is presented completely in closed form. That is, we derive analytical, and easy to interpret, expressions for the quantities of interest in LFA, such as the spectral radius and norm of the error propagation operator, rather than arriving at them through numerical computation, which offers significantly less insight.

While LFA yields convergence information that is only approximate for initial-value problems, we show that our theory yields results that are exact in the asymptotic limit $n_t \to \infty$. Furthermore, our analysis does yield exact convergence information for a certain class of time-periodic MGRIT/Parareal solvers for finite n_t. For this reason, our analysis shares some similarities with that presented in [43, Sec. 3.1] for a time-periodic Parareal algorithm. We note that many interesting physical problems possess time-periodic structure, such as the rotation of wind turbines, for example, and for this reason there have been many time-periodic, parallel-in-time algorithms developed (see, e.g., [100, 43, 87, 20, 49]).

The remainder of this chapter is organised as follows. Some key assumptions and notation are presented in Section 3.2. Section 3.3 derives the error propagation operator for the MGRIT algorithm. The convergence behaviour of MGRIT is theoretically analysed in Section 3.4 using LFA. A brief discussion on time-periodic problems is the subject of Section 3.5. Commentary on the theory and comparisons to related literature are the subject of Section 3.6. In Section 3.7, the LFA theory is applied to advection-dominated problems. Finally, concluding remarks are given in Section 3.8.

3.2 Notation and assumptions

The analysis in this chapter relies on the following assumptions on the time-stepping operators.

Assumption 3.1 (Simultaneous diagonalizability)**.** *The fine- and coarse-grid time-stepping operators Φ and Ψ are time independent, and simultaneously diagonalizable by a unitary matrix $\mathcal{U}$, with*

$$\Phi = \mathcal{U}\,\mathrm{diag}(\boldsymbol{\lambda})\mathcal{U}^* \in \mathbb{R}^{n_x \times n_x}, \tag{3.1}$$

$$\Psi = \mathcal{U}\,\mathrm{diag}(\boldsymbol{\mu})\mathcal{U}^* \in \mathbb{R}^{n_x \times n_x}, \tag{3.2}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ denote the vectors containing their eigenvalues,

$$\boldsymbol{\lambda} := \left(\lambda_1, \ldots, \lambda_{n_x}\right)^\top \in \mathbb{C}^{n_x}, \tag{3.3}$$

$$\boldsymbol{\mu} := \left(\mu_1, \ldots, \mu_{n_x}\right)^\top \in \mathbb{C}^{n_x}. \tag{3.4}$$

The assumption of simultaneous diagonalizability allows us to decouple the spatial and temporal components of the problem, and it has also appeared in previous analyses for this reason [38, 28, 88, 51, 26].

We also place an assumption on the stability of the operators, as follows.

Assumption 3.2 (Stability)**.** *The fine- and coarse-grid time-stepping operators Φ and Ψ are ℓ^2-stable, with $\|\Phi\|_2, \|\Psi\|_2 < 1$. Or equivalently, since the operators are unitarily diagonalizable, $|\lambda_i|, |\mu_i| < 1$, $\forall i \in \{1, \ldots, n_x\}$.*

Remark 3.1 (The constant mode)**.** *While Assumption 3.2 states that $|\lambda_i|, |\mu_i| < 1$ for all $i \in \{1, \ldots, n_x\}$, we will, however, also discuss the case that there exists an $i_* \in \{1, \ldots, n_x\}$ for which $\lambda_{i_*} = \mu_{i_*} = 1$. This case often arises for periodic boundary conditions in space, where the constant vector is propagated through time unchanged by Φ and Ψ. In general the reader should assume that Assumption 3.2 holds, and we will give specific commentary where relevant to the $\lambda_{i_*} = \mu_{i_*} = 1$ case.*

Suppose that the time interval $t \in [0, T]$ is discretized with n_t equidistant points $0 = t_0 < t_1 < \ldots < t_{n_t-1} = T$, where $t_n = n\delta t$. Further, suppose that the first time point t_0 is a C-point, and that the total number of points n_t is divisible by m.[1]

[1] This is a slightly different convention than we have used elsewhere in the thesis, such as in Section 1.4, for example, where there were $n_t + 1$ points in time, and the final time point was a C-point as shown in Figure 1.2. The motivation for our new convention is that it simplifies the notation and analysis if there is always a group of $m - 1$ F-points following a C-point.

The fine- and coarse-grid space-time operators are then given by

$$A_0 = \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{bmatrix} = I_{n_t} \otimes I_{n_x} - L_{n_t} \otimes \Phi \in \mathbb{R}^{n_t n_x \times n_t n_x}, \tag{3.5}$$

$$A_1 = \begin{bmatrix} I & & & \\ -\Psi & I & & \\ & \ddots & \ddots & \\ & & -\Psi & I \end{bmatrix} = I_{n_t/m} \otimes I_{n_x} - L_{n_t/m} \otimes \Psi \in \mathbb{R}^{\frac{n_t}{m} n_x \times \frac{n_t}{m} n_x}, \tag{3.6}$$

where L_n is the lower shift matrix,

$$L_n = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{3.7}$$

The symbol '$\otimes$' used here denotes the Kronecker product, which, for two matrices $A \in \mathbb{C}^{p \times q}$ and $B \in \mathbb{C}^{r \times s}$ is defined as the block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1q}B \\ \vdots & & \vdots \\ a_{p1}B & \cdots & a_{pq}B \end{bmatrix} \in \mathbb{C}^{pr \times qs}. \tag{3.8}$$

(See, e.g., [66] for the definition and basic properties of this operator.)

Finally, based on our description in the following section of operators in MGRIT acting in a block fashion, it will be convenient to represent solution vectors in block format. Specifically, we define a CF-interval as a C-point and the $m-1$ F-points that follow it, and we denote the kth CF-block of a space-time vector $\boldsymbol{u} = \left(\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_t-1}\right)^\top \in \mathbb{R}^{n_t n_x}$ by

$$\widehat{\boldsymbol{u}}_k = \left(\boldsymbol{u}_{km}, \boldsymbol{u}_{km+1}, \ldots \boldsymbol{u}_{km+m-1}\right)^\top \in \mathbb{R}^{m n_x}, \quad k \in \{0, \ldots, n_t/m - 1\}. \tag{3.9}$$

Furthermore, the jth vector in the kth CF-block is denoted as $\widehat{\boldsymbol{u}}_{k,j} = \boldsymbol{u}_{km+j}$, $j \in \{0, \ldots, m-1\}$.

3.3 Error propagation

The purpose of this section is to derive the error propagation operator for the MGRIT algorithm. Background discussion is first given in Section 3.3.1, and then key operators are derived in Section 3.3.2. A simplified representation for the error propagator is then derived in Section 3.3.3.

3.3.1 Background

The error propagation operator (or error propagator, or iteration operator) of a linear and stationary iterative algorithm describes the transformation of an error vector under the action of the algorithm. Specifically, suppose $\boldsymbol{e}^{(k)}$ is the algebraic error after k iterations of such an algorithm, then its error propagator $\mathcal{E}$ transforms the initial error $\boldsymbol{e}^{(0)}$ as

$$\boldsymbol{e}^{(k)} = \mathcal{E}^k \boldsymbol{e}^{(0)}, \quad k \in \mathbb{N}. \tag{3.10}$$

The properties of $\mathcal{E}^k$ therefore completely characterize the convergence of the iteration. For example, the spectral radius $\rho(\mathcal{E})$—the largest absolute eigenvalue of $\mathcal{E}$—gives the asymptotic convergence rate of the method. Furthermore, the norm of $\mathcal{E}$ may be used to bound the norm of the error, $\|\boldsymbol{e}^{(k)}\| = \|\mathcal{E}^k \boldsymbol{e}^{(0)}\| \leq \|\mathcal{E}^k\| \|\boldsymbol{e}^{(0)}\| \leq \|\mathcal{E}\|^k \|\boldsymbol{e}^{(0)}\|$.

The error propagator for a classical two-grid multigrid method takes the form (see, e.g., [99, Sec. 2.2.3])

$$\mathcal{E} = S_{\text{post}} \mathcal{K} S_{\text{pre}}, \quad \text{where } \mathcal{K} := I - P A_1^{-1} R A_0, \tag{3.11}$$

where S_{pre} and S_{post} denote iteration matrices for pre- and post relaxation, respectively. The error propagator $\mathcal{K}$ is that of the coarse-grid correction, in which R and P denote the restriction and interpolation operators, respectively.

There are several ways to express the MGRIT error propagator. In the literature, most approaches have done so using the reduction inherent in MGRIT to analyse error propagation on the coarse grid only [28, 88, 26, 51]. We will instead analyse error propagation on the fine grid, and in this sense, our approach is more closely related to those presented in [38] and [26, Sec. 4.1.3]. In addition to the above, MGRIT is not typically considered as a standard multigrid algorithm, in the sense of combining pre- and post-relaxation with a coarse-grid correction, and so its error propagator is not typically presented in a form akin to (3.11). Instead, MGRIT is most often described in terms of combining pre-relaxation with a coarse-grid correction that uses a non-standard interpolation, known as ideal interpolation [32, 38, 44, 58, 88, 51]. However, ideal interpolation is equivalent to the injection

of coarse-grid error at C-points, followed by an F-relaxation to update F-point values on the fine grid (see, e.g., [58]). Therefore, MGRIT can be interpreted as a standard multigrid algorithm with injection used for interpolation, and F-relaxation used for post-relaxation. We choose to adopt this description since it results in an error propagator which is closer to being symmetric (albeit in appearance only), and one that is more reminiscent of the error propagator (3.11). Note that our MGRIT description given in Algorithm 1 also does not use the terminology of ideal interpolation.

Therefore, we may write the MGRIT error propagator as

$$\mathcal{E} = S^{\mathrm{F}}\mathcal{K}\big(S^{\mathrm{CF}}\big)^{\nu}S^{\mathrm{F}}, \quad \text{where } \mathcal{K} = I - PA_1^{-1}P^{\top}A_0, \quad \nu \in \mathbb{N}_0. \tag{3.12}$$

Here, A_0 and A_1, given in (3.5) and (3.6), are the space-time operators on levels zero and one, respectively. The operator P is simply injection interpolation, and its transpose $P^{\top}$ is injection restriction. The operator S^{F} is the error propagator for F-relaxation. Finally, S^{CF} is the error propagator for CF-relaxation, which is defined as a C-relaxation followed by an F-relaxation. The variable $\nu \in \mathbb{N}_0$ represents the number of CF-relaxation sweeps, so that pre-relaxation consists of an F-relaxation followed by ν sweeps of CF-relaxation. For example, $\nu = 0$ corresponds to F-relaxation, and $\nu = 1$ to FCF-relaxation. As discussed in Section 1.4, numerical experiments throughout this thesis use FCF-relaxation, since we typically find it yields the most robust convergence. However, it is interesting to theoretically quantify the effect of the number of CF-relaxations. This ν-generalized relaxation has also been considered in [44, 51, 88].

3.3.2 Interpolation matrices and error propagators of relaxation

The purpose of this section is to derive convenient representations for the interpolation and relaxation components of the MGRIT error propagator $\mathcal{E}$ in (3.12). Recall that the interpolation operator P in (3.12) is based on injection. That is, P maps the kth C-point variable $\boldsymbol{u}_{km} \in \mathbb{R}^{n_x}$ as

$$\boldsymbol{u}_{km} \mapsto \begin{bmatrix} \boldsymbol{u}_{km} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} = \big(\boldsymbol{e}_1 \otimes I_{n_x}\big)\boldsymbol{u}_{km} \in \mathbb{R}^{mn_x}, \tag{3.13}$$

in which $\boldsymbol{e}_1 \in \mathbb{R}^m$ is the canonical (column-oriented) basis vector in the first direction, and is unrelated to the algebraic error $\boldsymbol{e}^{(k)}$ in the following. Therefore, the global space-time interpolation operator that acts on all n_t/m C-point variables simultaneously is the block

diagonal matrix

$$P = I_{n_t/m} \otimes \left(\boldsymbol{e}_1 \otimes I_{n_x}\right). \tag{3.14}$$

Note that the transpose of injection, which acts as the restriction operator in (3.12), is simply

$$P^\top = I_{n_t/m} \otimes \left(\boldsymbol{e}_1^\top \otimes I_{n_x}\right). \tag{3.15}$$

Next we consider the more complicated cases of the iteration operators for F- and C-relaxation in (3.12). To this end, suppose we have some approximation to the true solution of the system $A_0\boldsymbol{u} = \boldsymbol{b}$ denoted by $\boldsymbol{w}^{(0)} \approx \boldsymbol{u}$. Then, by definition, F-relaxation generates a new approximation $\boldsymbol{w}^{(0)} \mapsto \boldsymbol{w}^{(1)}$ such that C-point values of $\boldsymbol{w}^{(1)}$ are unchanged from those of $\boldsymbol{w}^{(0)}$, and F-point values of $\boldsymbol{w}^{(1)}$ have zero residual. In other words, F-relaxation represents an exact solve for the F-point variables of the system $A_0\boldsymbol{w}^{(1)} = \boldsymbol{b}$, where C-point variables in $\boldsymbol{w}^{(1)}$ are equal to those in $\boldsymbol{w}^{(0)}$. Therefore, on the kth CF-block, $k \in \{0, 1, \ldots, n_t/m - 1\}$, F-relaxation can be expressed as the update (recalling the block notation from (3.9))

$$\begin{aligned} \widehat{\boldsymbol{w}}_{k,0}^{(1)} &= \widehat{\boldsymbol{w}}_{k,0}^{(0)}, \\ \widehat{\boldsymbol{w}}_{k,j}^{(1)} &= \Phi^j \widehat{\boldsymbol{w}}_{k,0}^{(0)} + \widehat{\boldsymbol{b}}_{k,j}, \quad j \in \{1, \ldots, m-1\}. \end{aligned} \tag{3.16}$$

Replacing the approximations $\boldsymbol{w}^{(q)}$ via the error equations $\boldsymbol{w}^{(q)} = \boldsymbol{u} - \boldsymbol{e}^{(q)}$ leads to

$$\begin{aligned} &\widehat{\boldsymbol{u}}_{k,0} - \widehat{\boldsymbol{e}}_{k,0}^{(1)} = \widehat{\boldsymbol{u}}_{k,0} - \widehat{\boldsymbol{e}}_{k,0}^{(0)}, \\ &\widehat{\boldsymbol{u}}_{k,j} - \widehat{\boldsymbol{e}}_{k,j}^{(1)} = \Phi^j \left(\widehat{\boldsymbol{u}}_{k,0} - \widehat{\boldsymbol{e}}_{k,0}^{(0)}\right) + \widehat{\boldsymbol{b}}_{k,j} = \left(\Phi^j \widehat{\boldsymbol{u}}_{k,0} + \widehat{\boldsymbol{b}}_{k,j}\right) - \Phi^j \widehat{\boldsymbol{e}}_{k,0}^{(0)}, \quad j \in \{1, \ldots, m-1\}, \end{aligned} \tag{3.17}$$

with the last equality following from the linearity of Φ. Note that the exact solution $\boldsymbol{u}$ is a fixed-point of the update (3.16), $\widehat{\boldsymbol{u}}_{k,j} = \Phi^j \widehat{\boldsymbol{u}}_{k,0} + \widehat{\boldsymbol{b}}_{k,j}$, $j \in \{1, \ldots, m-1\}$. Therefore, update (3.17) can be recast in terms of the error as

$$\begin{aligned} \widehat{\boldsymbol{e}}_{k,0}^{(1)} &= \widehat{\boldsymbol{e}}_{k,0}^{(0)}, \\ \widehat{\boldsymbol{e}}_{k,j}^{(1)} &= \Phi^j \widehat{\boldsymbol{e}}_{k,0}^{(0)}, \quad j \in \{1, \ldots, m-1\}. \end{aligned} \tag{3.18}$$

Expressed in CF-block form, update (3.18) is

$$\widehat{\boldsymbol{e}}_k^{(1)} = \begin{bmatrix} I & 0 & \cdots & 0 \\ \Phi & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \Phi^{m-1} & 0 & \cdots & 0 \end{bmatrix} \widehat{\boldsymbol{e}}_k^{(0)} = \left[\boldsymbol{e}_1^\top \otimes v(\Phi)\right] \widehat{\boldsymbol{e}}_k^{(0)}, \tag{3.19}$$

where the Vandermonde-style function $v\colon \mathbb{C}^{n\times n} \to \mathbb{C}^{mn\times n}$ is defined as the block column vector

$$v(X) = \begin{bmatrix} I \\ X \\ \vdots \\ X^{m-1} \end{bmatrix}. \tag{3.20}$$

Thus, based on (3.19), the error propagator for F-relaxation that acts on all CF-blocks $k = 0, \ldots, n_t/m - 1$ simultaneously is the block diagonal matrix

$$S^{\mathrm{F}} = I_{n_t/m} \otimes \left[\boldsymbol{e}_1^\top \otimes v(\Phi)\right]. \tag{3.21}$$

Now consider C-relaxation. Recall that C-relaxation leaves F-point values unchanged and updates C-points such that they have zero residuals. In other words, C-relaxation represents an exact solve for the C-point variables of the system $A_0\boldsymbol{w}^{(1)} = \boldsymbol{b}$, where F-point variables in $\boldsymbol{w}^{(1)}$ are equal to those in $\boldsymbol{w}^{(0)}$. Therefore, the C-point update on variables in the kth CF-interval, $k \in \{1, \ldots, n_t/m - 1\}$, may be written

$$\begin{aligned} \widehat{\boldsymbol{w}}_{k,0}^{(1)} &= \Phi\widehat{\boldsymbol{w}}_{k-1,m-1}^{(0)} + \widehat{\boldsymbol{b}}_{k,0}, \\ \widehat{\boldsymbol{w}}_{k,j}^{(1)} &= \widehat{\boldsymbol{w}}_{k,j}^{(0)}, \quad j \in \{1, \ldots, m-1\}. \end{aligned} \tag{3.22}$$

Using the same logic as for F-relaxation above, (3.22) can be rewritten as the following update on the error,

$$\widehat{\boldsymbol{e}}_k^{(1)} = \begin{bmatrix} 0 & 0 & \cdots & \Phi \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \widehat{\boldsymbol{e}}_{k-1}^{(0)} + \begin{bmatrix} 0 & & & \\ & I_n & & \\ & & \ddots & \\ & & & I_n \end{bmatrix} \widehat{\boldsymbol{e}}_k^{(0)}. \tag{3.23}$$

Recall that in our formulation of $\mathcal{E}$ given by (3.12), a C-relaxation is always followed by an F-relaxation to create a CF-relaxation. Combining (3.19) and (3.23), it is easy to show that the error update for a CF-relaxation is simply

$$\widehat{\boldsymbol{e}}_k^{(1)} = \begin{bmatrix} 0 & \cdots & 0 & \Phi \\ 0 & \cdots & 0 & \Phi^2 \\ \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & \Phi^m \end{bmatrix} \widehat{\boldsymbol{e}}_{k-1}^{(0)} = \left[\boldsymbol{e}_m^\top \otimes v(\Phi)\Phi\right] \widehat{\boldsymbol{e}}_{k-1}^{(0)}, \tag{3.24}$$

Therefore, the error propagator for CF-relaxation that acts on all CF-blocks $k = 0, \ldots, n_t/m-1$ simultaneously is the block lower bidiagonal matrix

$$S^{\mathrm{CF}} = L_{n_t/m} \otimes \left[\boldsymbol{e}_m^\top \otimes v(\Phi)\Phi \right]. \tag{3.25}$$

3.3.3 Time-only MGRIT error propagation

Now that we have expressions for all of the components of the MGRIT error propagator $\mathcal{E}$ in (3.12), we exploit the diagonalizability of the time-stepping operators Φ and Ψ to block diagonalize it. In particular, the block diagonalization is accomplished via the similarity transform,

$$\left[\mathcal{P}^\top \left(I_{n_t} \otimes \mathcal{U}^*\right)\right] \mathcal{E} \left[\left(I_{n_t} \otimes \mathcal{U}\right)\mathcal{P}\right] =: \breve{\mathcal{E}} = \operatorname*{diag}_{1 \le i \le n_x} (\mathcal{E}_i) \in \mathbb{C}^{n_x n_t \times n_x n_t}, \tag{3.26}$$

where $\mathcal{P}$ is a permutation matrix to be defined shortly, and recall from Assumption 3.1 that $\mathcal{U}$ is the unitary matrix with eigenvectors of Φ and Ψ as its columns. Before describing in more detail this similarity transform, note that the diagonal blocks of $\breve{\mathcal{E}}$ and their components are given by

$$\mathcal{E}_i = S_i^{\mathrm{F}} \left(I_{n_t} - P_i A_{1,i}^{-1} P_i^\top A_{0,i} \right) \left(S_i^{\mathrm{CF}}\right)^\nu S_i^{\mathrm{F}} \in \mathbb{C}^{n_t \times n_t}, \tag{3.27}$$

$$A_{0,i} = I_{n_t} - \lambda_i L_{n_t} \in \mathbb{C}^{n_t \times n_t}, \tag{3.28}$$

$$A_{1,i} = I_{n_t/m} - \mu_i L_{n_t/m} \in \mathbb{C}^{n_t/m \times n_t/m}, \tag{3.29}$$

$$P_i = I_{n_t/m} \otimes \boldsymbol{e}_1 \in \mathbb{C}^{n_t \times n_t/m}, \tag{3.30}$$

$$S_i^{\mathrm{F}} = I_{n_t/m} \otimes \left[v(\lambda_i) \boldsymbol{e}_1^\top \right] \in \mathbb{C}^{n_t \times n_t}, \tag{3.31}$$

$$S_i^{\mathrm{CF}} = L_{n_t/m} \otimes \left[\lambda_i v(\lambda_i) \boldsymbol{e}_m^\top \right] \in \mathbb{C}^{n_t \times n_t}. \tag{3.32}$$

Notice that the interpolation operators P_i are independent of i. Since similar matrices have the same eigenvalues, the eigenvalues of $\breve{\mathcal{E}}$ are equal to those of $\mathcal{E}$. Furthermore, since the similarity transform (3.26) uses only unitary matrices, it preserves the ℓ^2-norm,

$$\|\mathcal{E}\|_2 = \left\|\breve{\mathcal{E}}\right\|_2 = \max_{1 \le i \le n_x} \|\mathcal{E}_i\|_2. \tag{3.33}$$

Therefore, the task of characterizing the convergence of MGRIT has been reduced to characterizing the properties of the n_t-dimensional error propagators $\mathcal{E}_i$. The analysis of these operators via LFA is the subject of the following section.

Remark 3.2 (Non-unitary $\mathcal{U}$)**.** *If one removes the assumption that the matrix of eigenvectors $\mathcal{U}$ is unitary, then much of the forthcoming analysis in this chapter is still valid. However, rather than holding in the ℓ^2-norm, it does so in a modified norm related to the*

eigenvectors. From a practical perspective, it suffices to know that the results derived here can still provide an upper bound on the ℓ^2-norm on $\mathcal{E}$; more specifically, rather than (3.33), in the case of non-unitary $\mathcal{U}$, one has

$$\|\mathcal{E}\|_2 \leq \kappa(\mathcal{U}) \max_{1\leq i\leq n_x} \|\mathcal{E}_i\|_2, \tag{3.34}$$

in which $\kappa(\mathcal{U}) = \|\mathcal{U}\|_2\|\mathcal{U}^{-1}\|_2$, and $\mathcal{E}_i$ is still as in (3.27). See [88, 26, 51] for further details. Note also that in the event that computing eigenvalues is difficult, or that the theory cannot be applied rigorously since the operators are not diagonalizable, for example, one may derive an approximate convergence results by replacing the true eigenvalues of Φ and Ψ with their Fourier symbols (see [26, Rem. 1]).[2]

We now briefly explain how the similarity transform in (3.26) works. The transformation matrix is $(I_{n_t} \otimes \mathcal{U})\mathcal{P}$, a product of two matrices; the first one is $I_{n_t} \otimes \mathcal{U}$, and its role is to diagonalize the occurrences of Φ and Ψ in (3.12), replacing them with diagonal matrices populated with their eigenvalues. The second matrix $\mathcal{P}$ is a permutation that reorders space-time vectors from the original ordering where all spatial degrees of freedom (DOFs) at a single time point are blocked together, to one in which all temporal DOFs belonging at a single spatial point are blocked together.

Note that the Kronecker product and outer product of two column vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^n$ are related by $\boldsymbol{a}^\top \otimes \boldsymbol{b} = \boldsymbol{b}\boldsymbol{a}^\top$. For example, the ith component/diagonal block of the CF-relaxation propagator given in (3.32) can also be written as $S_i^{\mathrm{CF}} = L_{n_t/m} \otimes \left[\boldsymbol{e}_m^\top \otimes \lambda_i v(\lambda_i)\right]$. Contrasting with the full CF-relaxation operator $S^{\mathrm{CF}} = L_{n_t/m} \otimes \left[\boldsymbol{e}_m^\top \otimes v(\Phi)\Phi\right]$ from (3.25), we see that the similarity transform has block diagonalized S^{CF} by replacing occurrences of the time-stepping operator with its eigenvalues, $\Phi \mapsto \lambda_i$. Considering more broadly the components given by (3.27)–(3.32), we see that the effect of the similarity transform in (3.26) has been to block diagonalize the full space-time error propagator by replacing occurrences of the time-stepping operators with their eigenvalues $\Phi \mapsto \lambda_i$, $\Psi \mapsto \mu_i$. In other words, the global space-time problem has been decoupled into n_x scalar, time-only problems, with the ith such problem representing the error propagation of the ith spatial eigenvector.

Having made this realisation, let us briefly return to the content of Remark 3.1 in which we stated that a pair of fine- and coarse-grid eigenvalues equal to unity was permissible, but otherwise, under Assumption 3.2, the magnitude of all eigenvalues should be less than unity.

[2] In fact, we already did this in Section 2.5.5, where we applied our periodic coarse-grid optimization approach to an inflow problem. Specifically, the convergence theory from [28] that we used for the approach did not rigorously apply for inflow boundaries because the resulting Ψ was not diagonalizable, so we applied the theory from [28] using instead the Fourier symbols of Ψ (i.e., the eigenvalues of the circulant extension of Ψ).

Lemma 3.3. *Suppose there exists an index* $i_* \in \{1, \ldots, n_x\}$ *such that* $\lambda_{i_*} = \mu_{i_*} = 1$. *Then, the associated error propagator* (3.27) *is*

$$\mathcal{E}_{i_*} = 0. \tag{3.35}$$

Proof. Observe that the error propagator $\mathcal{E}_{i_*}$ in (3.27) corresponds to a scalar initial-value problem with fine-grid time-stepping operator $\lambda_{i_*} = 1$, and coarse-grid time-stepping operator $\mu_{i_*} = 1$ (see, e.g., the expressions for $A_{0,i}$ and $A_{1,i}$ in (3.28) and (3.29)). Notice that $\mu_{i_*} = 1 = \left(\lambda_{i_*}\right)^m$, and therefore that the problem uses an ideal coarse-grid operator. The result (3.35) follows immediately by recalling that MGRIT converges to the exact solution in a single iteration when using the ideal coarse-grid operator (see Section 1.4). ■

In other words, if there exists an eigenvector for which $\lambda_{i_*} = \mu_{i_*} = 1$, then MGRIT exactly eliminates error in the direction of this eigenvector in a single iteration. Thus, in our forthcoming investigations of $\mathcal{E}_i$, we assume that $|\lambda_i|, |\mu_i| < 1$, $\forall i$ as per Assumption 3.2.

3.4 Local Fourier analysis

In this section, we use LFA to analyse the scalar or time-only MGRIT error propagators $\mathcal{E}_i$ given by (3.27). The LFA framework for analysing the error propagators $\mathcal{E}_i$ is described in Section 3.4.1. The initial theoretical work of computing the eigenmatrices is the subject of Section 3.4.2. Finally, the main theoretical results are given in Section 3.4.3.

3.4.1 Introduction and preliminaries

Originally proposed in [9], LFA is a predictive tool for approximately determining the asymptotic convergence behaviour of multigrid methods, or that of their components, such as relaxation, for example. To analyse the problem with LFA, one reconsiders the two-grid problem posed on a pair of infinite grids, in which the influence and effects of boundary conditions are ignored. On these grids, one makes use of the fact that Fourier modes are formal eigenfunctions of the infinite-dimensional Toeplitz operators that are present in the multigrid error propagator (i.e., those arising from the discretization of a constant-coefficient PDE). It is then possible to show that the error propagator leaves invariant certain low-dimensional spaces of these Fourier modes. As such, the properties of the iteration operator can be characterized by the properties of certain low-dimensional operators which are feasible to compute numerically.

To this end, with ℓ the level in our multigrid hierarchy, we associate the semi-infinite temporal grid

$$\boldsymbol{G}_\ell := \left\{ t_k = km^\ell \delta t \colon k \in \mathbb{N}_0 \right\}, \quad \ell \in \{0, 1\}. \tag{3.36}$$

On the grids $\boldsymbol{G}_\ell$, $\ell \in \{0, 1\}$, we consider the infinite-dimensional extension of matrix $\mathcal{E}_i$ in (3.27) and the infinite-dimensional extensions of the matrices that define it given in (3.28)–(3.32).

In addition to considering the multigrid components on the grids (3.36), we also consider the following Fourier modes

$$\varphi_\ell(\theta, t) := \exp\left(\frac{\mathrm{i}\theta t}{m^\ell \delta t}\right), \quad t \in \boldsymbol{G}_\ell, \quad \theta \in \Theta_\ell, \tag{3.37}$$

where

$$\Theta_\ell = \begin{cases} \left[-\frac{\pi}{m}, 2\pi - \frac{\pi}{m}\right), & \ell = 0, \\ [-\pi, \pi) & \ell = 1, \end{cases} \tag{3.38}$$

and θ varying continuously in Θ_ℓ. Note that any intervals of length 2π could be used for Θ_ℓ, but these choices provide some notational simplifications (see also [26]). We adopt the shorthand $\boldsymbol{\varphi}_\ell(\theta)$ for denoting the Fourier mode (3.37) sampled at all time points $t \in \boldsymbol{G}_\ell$. The modes $\boldsymbol{\varphi}_\ell(\theta)$ lie at the heart of LFA because they are formally eigenfunctions of any infinite-dimensional Toeplitz operator that acts on the grid $\boldsymbol{G}_\ell$ [99, 105].

On $\boldsymbol{G}_\ell$ we introduce the following scaled Hermitian inner product of two grid functions $a_\ell, b_\ell \colon \boldsymbol{G}_\ell \to \mathbb{C}$ as

$$\langle \boldsymbol{a}_\ell, \boldsymbol{b}_\ell \rangle := \lim_{n_t \to \infty} \frac{1}{n_t} \sum_{k=0}^{n_t - 1} \bar{a}_\ell(t_k) b_\ell(t_k), \tag{3.39}$$

in which $\bar{a}(t_k)$ denotes the complex conjugate of the complex number $a(t_k)$. Note that the Fourier modes (3.37) are orthonormal with respect to this inner product.

As is standard in the Fourier analysis of multigrid methods, we partition the continuous frequency space Θ_0 into two disjoint sets by

$$\Theta^{\text{low}} := \left[-\frac{\pi}{m}, \frac{\pi}{m}\right), \quad \Theta^{\text{high}} := \left[\frac{\pi}{m}, 2\pi - \frac{\pi}{m}\right). \tag{3.40}$$

Observe for any $\theta \in \Theta^{\text{low}}$ that $\varphi_0\left(\theta + \frac{2\pi\alpha}{m}, t\right) = \frac{1}{\sqrt{m}} \varphi_1\left(m\theta, t\right)$ for $t \in \boldsymbol{G}_1$, $\alpha \in \{0, \ldots, m-1\}$. These m fine-grid functions $\varphi_0\left(\theta + \frac{2\pi\alpha}{m}, t\right)$ are known as *harmonics* of one another. The fact that the harmonics are indistinguishable from one another when sampled on the

coarse-grid points is the motivation for the partitioning in (3.40). See also [26, Sec. 4.1.2] for related discussion.

Since the harmonics are indistinguishable from one another when sampled on the coarse grid, they become coupled or mixed when acted on by the restriction and interpolation operators. Based on this, we define the following m-dimensional spaces of harmonics.

Definition 3.4 ($m\delta t$-harmonics)**.** *For a given $\theta \in \Theta^{\text{low}}$, the associated m-dimensional space of harmonics is*

$$\mathcal{H}_{\delta t}^{\theta} \coloneqq \operatorname*{span}_{0 \le \alpha < m} \left\{ \boldsymbol{\varphi}_0 \left(\theta + \frac{2\pi\alpha}{m} \right) \right\}. \tag{3.41}$$

Not only does interpolation intermix Fourier harmonics, but so does relaxation, as we shall show. In other words, the Fourier modes (3.37) are neither eigenfunctions of interpolation nor relaxation. More generally, for a given $\theta \in \Theta^{\text{low}}$, the action of the various MGRIT components (3.28)–(3.32) on Fourier modes (3.37) may be characterized as

$$A_{0,i} \colon \mathcal{H}_{\delta t}^{\theta} \to \mathcal{H}_{\delta t}^{\theta}, \tag{3.42}$$

$$A_{1,i} \colon \operatorname{span}\{\boldsymbol{\varphi}_1(m\theta)\} \to \operatorname{span}\{\boldsymbol{\varphi}_1(m\theta)\}, \tag{3.43}$$

$$P_i \colon \operatorname{span}\{\boldsymbol{\varphi}_1(m\theta)\} \to \mathcal{H}_{\delta t}^{\theta}, \tag{3.44}$$

$$S_i^{\text{F}} \colon \mathcal{H}_{\delta t}^{\theta} \to \mathcal{H}_{\delta t}^{\theta}, \tag{3.45}$$

$$S_i^{\text{CF}} \colon \mathcal{H}_{\delta t}^{\theta} \to \mathcal{H}_{\delta t}^{\theta}. \tag{3.46}$$

Note that the spans in (3.43) and (3.44) are over a one-dimensional set.

Since the MGRIT error propagator $\mathcal{E}_i$ is composed of the above operators (see (3.27)), it is invariant on the space of $m\delta t$-harmonics:

$$\mathcal{E}_i \colon \mathcal{H}_{\delta t}^{\theta} \to \mathcal{H}_{\delta t}^{\theta} \quad \text{for all } \theta \in \Theta^{\text{low}}. \tag{3.47}$$

Therefore, by grouping together harmonic Fourier modes, the infinite-dimensional error propagator $\mathcal{E}_i$ can be block diagonalized. Specifically, the transformed operator has one diagonal block $\widehat{\mathcal{E}}_i(\theta) \in \mathbb{C}^{m \times m}$ associated with each $\theta \in \Theta^{\text{low}}$, where, for a given $\theta \in \Theta^{\text{low}}$, $\widehat{\mathcal{E}}_i(\theta)$ is the representation of $\mathcal{E}_i$ on the harmonic space $\mathcal{H}_{\delta t}^{\theta}$. That is,

$$\mathcal{E}_i \underset{\text{similarity transform}}{\mapsto} \operatorname*{diag}_{\theta \in \Theta^{\text{low}}} \left(\widehat{\mathcal{E}}_i(\theta) \right). \tag{3.48}$$

Following the notation of [78], we call $\widehat{\mathcal{E}}_i(\theta)$ the *eigenmatrix* of $\mathcal{E}_i$ associated with the harmonic space $\mathcal{H}^{\theta}_{\delta t}$ (or just the eigenmatrix of $\mathcal{E}_i$ for short).[3] More specifically, let $V(\theta)$ be the matrix whose m columns are the Fourier modes defining the harmonic space $\mathcal{H}^{\theta}_{\delta t}$ given in (3.41),

$$V(\theta) := \left[\boldsymbol{\varphi}_0(\theta) \quad \boldsymbol{\varphi}_0\left(\theta + \tfrac{2\pi}{m}\right) \quad \dots \quad \boldsymbol{\varphi}_0\left(\theta + \tfrac{2\pi(m-1)}{m}\right)\right]. \tag{3.49}$$

Then, the eigenmatrix arising in the similarity transform (3.48) associated with $\mathcal{H}^{\theta}_{\delta t}$ can be expressed as

$$\mathcal{E}_i V(\theta) = V(\theta)\widehat{\mathcal{E}}_i(\theta) \quad \Longrightarrow \quad \widehat{\mathcal{E}}_i(\theta) = V^*(\theta)\mathcal{E}_i V(\theta) \in \mathbb{C}^{m\times m}. \tag{3.50}$$

Similar matrix operators have the same eigenvalues, and therefore, from (3.48), the spectral radius of the error propagator is

$$\rho(\mathcal{E}_i) = \rho\left(\operatorname*{diag}_{\theta\in\Theta^{\text{low}}}\left(\widehat{\mathcal{E}}_i(\theta)\right)\right) = \sup_{\theta\in\Theta^{\text{low}}} \rho\big(\widehat{\mathcal{E}}_i(\theta)\big). \tag{3.51}$$

Moreover, since the similarity transform in (3.48) is unitary, it preserves the ℓ^2-norm of $\mathcal{E}_i$,

$$\|\mathcal{E}_i\|_2 = \left\|\operatorname*{diag}_{\theta\in\Theta^{\text{low}}}\left(\widehat{\mathcal{E}}_i(\theta)\right)\right\|_2 = \sup_{\theta\in\Theta^{\text{low}}} \big\|\widehat{\mathcal{E}}_i(\theta)\big\|_2. \tag{3.52}$$

Thus, the computation of the spectral radius and norm of the infinite-dimensional $\mathcal{E}_i$ has been reduced to the computation of these quantities on an infinite number of matrices $\widehat{\mathcal{E}}_i(\theta) \in \mathbb{C}^{m\times m}$, $\theta \in \Theta^{\text{low}}$.

From the definition of $\mathcal{E}_i$ in (3.27), the m-dimensional error propagator $\widehat{\mathcal{E}}_i(\theta)$ from (3.50) can be written as

$$\widehat{\mathcal{E}}_i(\theta) = \widehat{S}_i^{\text{F}}(\theta)\,\widehat{\mathcal{K}}_i(\theta)\left(\widehat{S}_i^{\text{CF}}(\theta)\right)^{\nu}\widehat{S}_i^{\text{F}}(\theta) \in \mathbb{C}^{m\times m}, \tag{3.53}$$

where $\widehat{\mathcal{K}}_i(\theta)$ is the eigenmatrix of the coarse-grid correction,

$$\widehat{\mathcal{K}}_i(\theta) = I_m - \widehat{P}_i(\theta)\left(\widehat{A}_{1,i}(m\theta)\right)^{-1}\widehat{P}_i^{\top}(\theta)\,\widehat{A}_{0,i}(\theta) \in \mathbb{C}^{m\times m}. \tag{3.54}$$

[3] Note that in the LFA literature this is most commonly referred to as the *Fourier symbol* of $\mathcal{E}_i$, but we reserve this terminology for eigenvalues, as is discussed shortly. The term eigenmatrix is intended to express that this is a generalization of the standard eigenvector-eigenvalue relationship in which an operator maps a vector to a constant multiple of itself; now an operator maps a space of vectors to a matrix multiple of that space, or a subspace of it (see (3.50)).

The component matrices used in these expressions are the eigenmatrices of the infinite-dimensional extensions of the multigrid components defined in (3.28)–(3.32). From (3.42)–(3.46), these eigenmatrices may be expressed as

$$\widehat{A}_{0,i}(\theta) = V^*(\theta)A_{0,i}V(\theta) \in \mathbb{C}^{m\times m}, \tag{3.55}$$

$$\widehat{A}_{1,i}(m\theta) = \boldsymbol{\varphi}_1^*(m\theta)A_{1,i}\boldsymbol{\varphi}_1(m\theta) \in \mathbb{C}, \tag{3.56}$$

$$\widehat{P}_i(\theta) = V^*(\theta)P_i\boldsymbol{\varphi}_1(m\theta) \in \mathbb{C}^m, \tag{3.57}$$

$$\widehat{S}_i^{\mathrm{F}}(\theta) = V^*(\theta)S_i^{\mathrm{F}}V(\theta) \in \mathbb{C}^{m\times m}, \tag{3.58}$$

$$\widehat{S}_i^{\mathrm{CF}}(\theta) = V^*(\theta)S_i^{\mathrm{CF}}V(\theta) \in \mathbb{C}^{m\times m}. \tag{3.59}$$

The outstanding task is computing these eigenmatrices.

3.4.2 Derivations of eigenmatrices

The subject of this section is deriving all of the eigenmatrices (3.53)–(3.59). Specifically, those for the fine- and coarse-grid operators, and interpolation are presented in Section 3.4.2.1. Those for relaxation are derived in Section 3.4.2.2, and finally, that of the error propagator itself is considered in Section 3.4.2.3.

3.4.2.1 Eigenmatrices of fine- and coarse-grid operators, and interpolation

Let us begin with the simplest components, which are the fine- and coarse-grid operators (3.55) and (3.56). The Fourier modes $\boldsymbol{\varphi}_\ell(\theta)$ are eigenfunctions of the fine- and coarse-grid operators, $A_{0,i}$ and $A_{1,i}$ since they are infinite-dimensional Toeplitz operators, and, therefore, their eigenmatrices are diagonal, and are simply given by

$$\widehat{A}_{i,0}(\theta) = \operatorname*{diag}_{0\le\alpha<m}\left(\widetilde{A}_{0,i}\left(\theta + \frac{2\pi\alpha}{m}\right)\right) \in \mathbb{C}^{m\times m}, \tag{3.60}$$

$$\widehat{A}_{1,i}(m\theta) = \widetilde{A}_{1,i}(m\theta) \in \mathbb{C}. \tag{3.61}$$

Here, $\widetilde{A}_{\ell,i}(\theta)$ denotes the eigenvalue or *Fourier symbol* of $A_{\ell,i}$ associated with the eigenfunction $\boldsymbol{\varphi}_\ell(\theta)$. The lower bidiagonal operators $A_{0,i}$ and $A_{1,i}$, given by (3.28) and (3.29), respectively, have Fourier symbols

$$\widetilde{A}_{0,i}(\theta) = 1 - \lambda_i e^{-\mathrm{i}\theta} \in \mathbb{C}, \quad \widetilde{A}_{1,i}(m\theta) = 1 - \mu_i e^{-\mathrm{i}m\theta} \in \mathbb{C}. \tag{3.62}$$

Remark 3.5 (Invertability of space-time operators)**.** *Thus far, we have implicitly assumed that the infinite-dimensional operators* $\left(A_{0,i}\right)^{-1}$ *and* $\left(A_{1,i}\right)^{-1}$ *exist. This is equivalent to their Fourier symbols vanishing nowhere with respect to* θ*. Recall from Assumption 3.2,*

that $|\lambda_i|, |\mu_i| < 1 \, \forall i$*. Therefore, from* (3.62)*, it is clear that their the Fourier symbols are bounded away from zero.*

Now we consider the case of injection interpolation (3.57). This is perhaps easiest by considering its transpose, which is injection restriction. Recall that part of our motivation for defining the spaces of harmonics (3.41) was that for any $\theta \in \Theta^{\text{low}}$, $\varphi_0\left(\theta + \frac{2\pi\alpha}{m}, t\right) = \frac{1}{\sqrt{m}}\varphi_1\left(m\theta, t\right)$ for coarse time points $t \in \boldsymbol{G}_1$, where $\alpha \in \{0, \ldots, m-1\}$. Notice then that injection restriction, which takes the values of $\varphi_0\left(\theta + \frac{2\pi\alpha}{m}, t\right)$ at coarse time points $t \in \boldsymbol{G}_1$, maps a fine-grid harmonic to the associated coarse-grid Fourier mode $\varphi_1\left(m\theta, t\right)$, but with its amplitude scaled by $\frac{1}{\sqrt{m}}$. In other words, the eigenmatrix of injection restriction is $\frac{1}{\sqrt{m}}\mathbf{1}^\top$, where $\mathbf{1} \in \mathbb{R}^m$ denotes a column vector of ones. Therefore the eigenmatrix of injection interpolation—its transpose—is simply[4]

$$\widehat{P}_i(\theta) = \frac{1}{\sqrt{m}}\mathbf{1}. \tag{3.63}$$

3.4.2.2 Eigenmatrices of relaxation

Let us now move to the more difficult task of computing the eigenmatrices of the relaxation operators (3.58) and (3.59). These calculations are more difficult because the relaxation operators intermix harmonics in a non-trivial way.

Lemma 3.6 (Eigenmatrix of F-relaxation)**.** *The eigenmatrix* (3.58) *of F-relaxation may be written as the following rank-1 matrix*

$$\widehat{S}_i^{\text{F}}(\theta) = c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top, \tag{3.64}$$

in which $\widehat{A}_{0,i}(\theta)$ *given in* (3.60) *is the diagonal eigenmatrix of the fine-grid operator* $A_{0,i}$*, and* $c(\theta)$ *is the function*

$$c(\theta) := \frac{1}{m}\left[1 - \left(\lambda_i e^{-\mathrm{i}\theta}\right)^m\right]. \tag{3.65}$$

Proof. From the representation of $\widehat{S}_i^{\text{F}}(\theta)$ given by (3.58), its (p,q)th element, $p, q \in \{0, \ldots, m-1\}$, is equal to

$$\left[\widehat{S}_i^{\text{F}}(\theta)\right]_{p,q} = \left\langle \boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi p}{m}\big), S_i^{\text{F}}\boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi q}{m}\big)\right\rangle. \tag{3.66}$$

[4]Note that in general when interpolation is the transpose of restriction, its eigenmatrix is equal to the transpose of that of interpolation only up to some constant scaling (see, e.g., [99, Rem. 4.4.3]). However, injection represents a special case in which the constant scaling is one.

We begin by considering the vector in the right-hand side of this inner product. Using the definition of S_i^{F} given in (3.31), the jth element of the vector $S_i^{\mathrm{F}}\boldsymbol{\varphi}_0\big(\theta+\frac{2\pi q}{m}\big)$ is

$$\Big[S_i^{\mathrm{F}}\boldsymbol{\varphi}_0\big(\theta+\tfrac{2\pi q}{m}\big)\Big]_j = \lambda_i^{j \bmod m}\exp\Big[\tfrac{\mathrm{i}}{\delta t}\big(\theta+\tfrac{2\pi q}{m}\big)\big(j-j \bmod m\big)\delta t\Big], \tag{3.67}$$

$$= \Big[\lambda_i \exp\Big(-\mathrm{i}\big(\theta+\tfrac{2\pi q}{m}\big)\Big)\Big]^{j \bmod m}\Big[\boldsymbol{\varphi}_0\big(\theta+\tfrac{2\pi q}{m}\big)\Big]_j, \tag{3.68}$$

$$\equiv \big[\zeta_i(\theta,q)\big]^{j \bmod m}\Big[\boldsymbol{\varphi}_0\big(\theta+\tfrac{2\pi q}{m}\big)\Big]_j \tag{3.69}$$

where, in the final equation, we have introduced the shorthand function $\zeta_i(\theta,q)$.

Now, considering the inner product (3.66) and some algebra gives

$$\Big[\widehat{S}_i^{\mathrm{F}}(\theta)\Big]_{p,q} = \lim_{n_t\to\infty}\frac{1}{n_t}\sum_{k=0}^{n_t-1}\big[\zeta_i(\theta,q)\big]^{k \bmod m}\exp\Big[\tfrac{-\mathrm{i}}{\delta t}\big(\theta+\tfrac{2\pi p}{m}\big)k\delta t\Big]\exp\Big[\tfrac{\mathrm{i}}{\delta t}\big(\theta+\tfrac{2\pi q}{m}\big)k\delta t\Big], \tag{3.70}$$

$$= \lim_{n_t\to\infty}\frac{1}{n_t}\sum_{k=0}^{n_t-1}\big[\zeta_i(\theta,q)\big]^{k \bmod m}\exp\Big(\tfrac{2\pi\mathrm{i}k}{m}(q-p)\Big), \tag{3.71}$$

$$= \lim_{n_t\to\infty}\frac{1}{n_t}\sum_{r=0}^{m-1}\left(\big[\zeta_i(\theta,q)\big]^r\left[\sum_{k=0}^{\frac{n_t}{m}-1}\exp\Big(\tfrac{2\pi\mathrm{i}(km+r)}{m}(q-p)\Big)\right]\right), \tag{3.72}$$

$$= \lim_{n_t\to\infty}\frac{1}{n_t}\sum_{r=0}^{m-1}\left(\big[\zeta_i(\theta,q)\big]^r\left[\exp\Big(\tfrac{2\pi\mathrm{i}r}{m}(q-p)\Big)\sum_{k=0}^{\frac{n_t}{m}-1}\exp\Big(2\pi\mathrm{i}k(q-p)\Big)\right]\right), \tag{3.73}$$

$$= \lim_{n_t\to\infty}\frac{1}{n_t}\sum_{r=0}^{m-1}\left(\big[\zeta_i(\theta,q)\big]^r\left[\frac{n_t}{m}\exp\Big(\tfrac{2\pi\mathrm{i}r}{m}(q-p)\Big)\right]\right), \tag{3.74}$$

$$= \frac{1}{m}\sum_{r=0}^{m-1}\Big[\zeta_i(\theta,q)\exp\Big(\tfrac{2\pi\mathrm{i}}{m}(q-p)\Big)\Big]^r, \tag{3.75}$$

Substituting the value of $\zeta_i(\theta,q)$ from (3.69) into the equation above and simplifying the resulting geometric sum gives

$$\Big[\widehat{S}_i^{\mathrm{F}}(\theta)\Big]_{p,q} = \frac{1}{m}\sum_{r=0}^{m-1}\Big[\lambda_i\exp\Big(-\mathrm{i}\Big(\theta+\tfrac{2\pi p}{m}\Big)\Big)\Big]^r, \tag{3.76}$$

$$= \frac{1}{m}\frac{1-\big(\lambda_i e^{-\mathrm{i}\theta}\big)^m}{1-\exp\big(-\mathrm{i}\big(\theta+\frac{2\pi p}{m}\big)\big)}. \tag{3.77}$$

Observe from (3.77) that $\Big[\widehat{S}_i^{\mathrm{F}}(\theta)\Big]_{p,q}$ does not depend on the column index q, but only the row index p. This means that $\widehat{S}_i^{\mathrm{F}}(\theta)$ can be expressed as an outer product of two vectors in the form $\widehat{S}_i^{\mathrm{F}}(\theta)=\frac{1}{m}\big[1-\big(\lambda_i e^{-\mathrm{i}\theta}\big)^m\big]\boldsymbol{a}\boldsymbol{1}^\top$, for some vector $\boldsymbol{a}$ whose pth element is one divided by the function $1-\exp\big(-\mathrm{i}\big(\theta+\frac{2\pi p}{m}\big)\big)$. Notice that this function is simply

$\widetilde{A}_{0,i}\big(\theta + \frac{2\pi p}{m}\big)$, where $\widetilde{A}_{0,i}(\theta)$ is the Fourier symbol of $A_{0,i}$ given in (3.62). Therefore, the vector $\boldsymbol{a}$ is equal to $\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}$, in which $\widehat{A}_{0,i}(\theta)$ is the diagonal matrix given in (3.60) that contains the Fourier symbols of $A_{0,i}$ for the harmonic modes. This immediately gives the result (3.64) for $\widehat{S}_i^{\mathrm{F}}(\theta)$. ■

That fact that the F-relaxation eigenmatrix $\widehat{S}_i^{\mathrm{F}}(\theta) = c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^{\top}$, as given in (3.64), is dense reflects that F-relaxation couples together the m harmonics in the space $\mathcal{H}_{\delta t}^{\theta}$ (see (3.41)). This is in contrast to some simple relaxation methods, such as those typically used in the multigrid solution of Poisson problems, for example, for which the eigenmatrix is diagonal (or block diagonal), representing that harmonics are not intermixed (or partially intermixed) by relaxation [99].

Having identified the eigenmatrix for F-relaxation, we move to consider the eigenmatrix of CF-relaxation, with our result described in the following lemma.

Lemma 3.7 (Eigenmatrix of pre-relaxation)**.** *The eigenmatrix* (3.59) *of CF-relaxation may be expressed as*

$$\widehat{S}_i^{\mathrm{CF}}(\theta) = \widehat{S}_i^{\mathrm{F}}(\theta)\Big[I - \widehat{A}_{0,i}(\theta)\Big], \tag{3.78}$$

where the eigenmatrix of F-relaxation $\widehat{S}_i^{\mathrm{F}}(\theta)$ *is given by* (3.64)*, and* $\widehat{A}_{0,i}(\theta)$ *is given in* (3.60)*. Furthermore, the eigenmatrix for the entire pre-relaxation operator in* (3.53) *may be written as*

$$\big(\widehat{S}_i^{\mathrm{CF}}(\theta)\big)^{\nu}\,\widehat{S}_i^{\mathrm{F}}(\theta) = \big(\lambda_i e^{-\mathrm{i}\theta}\big)^{m\nu}\,\widehat{S}_i^{\mathrm{F}}(\theta), \quad \nu \in \mathbb{N}_0. \tag{3.79}$$

Proof. We begin by computing the eigenmatrix for CF-relaxation. From (3.59), the (p,q)th element of $\widehat{S}_i^{\mathrm{CF}}(\theta)$ is equal to

$$\Big[\widehat{S}_i^{\mathrm{CF}}(\theta)\Big]_{p,q} = \Big\langle \boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi p}{m}\big), S_i^{\mathrm{CF}}\boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi q}{m}\big)\Big\rangle. \tag{3.80}$$

Using the expression for S_i^{CF} given in (3.32), the jth element of the vector $S_i^{\mathrm{CF}}\boldsymbol{\varphi}_0\big(\theta + \frac{2\pi q}{m}\big)$ can be written as

$$\Big[S_i^{\mathrm{CF}}\boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi q}{m}\big)\Big]_j = \lambda_i \lambda_i^{j \bmod m} \exp\Big[\tfrac{\mathrm{i}}{\delta t}\big(\theta + \tfrac{2\pi q}{m}\big)\big(j - 1 - j \bmod m\big)\delta t\Big], \tag{3.81}$$

$$= \zeta_i(\theta,q)\big[\zeta_i(\theta,q)\big]^{j \bmod m}\Big[\boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi q}{m}\big)\Big]_j, \tag{3.82}$$

$$= \zeta_i(\theta,q)\Big[S_i^{\mathrm{F}}\boldsymbol{\varphi}_0\big(\theta + \tfrac{2\pi q}{m}\big)\Big]_j, \tag{3.83}$$

where the function $\zeta_i(\theta, q)$ is as in (3.69), and $\left[\zeta_i(\theta, q)\right]^{j \bmod m}\left[\boldsymbol{\varphi}_0\big(\theta + \frac{2\pi q}{m}\big)\right]_j$ has been replaced with $\left[S_i^{\mathrm{F}}\boldsymbol{\varphi}_0\big(\theta + \frac{2\pi q}{m}\big)\right]_j$ by using (3.69).

Therefore, from (3.80) and (3.83), the following simple relationship holds between the (p, q)th element of the eigenmatrices of CF- and F-relaxation:

$$\left[\widehat{S}_i^{\mathrm{CF}}(\theta)\right]_{p,q} = \zeta_i(\theta, q)\left[\widehat{S}_i^{\mathrm{F}}(\theta)\right]_{p,q}. \tag{3.84}$$

From its definition (see (3.69)), $\zeta_i(\theta, q) = 1 - \widetilde{A}_{0,i}\big(\theta + \frac{2\pi q}{m}\big)$, where $\widetilde{A}_{0,i}(\theta)$ defined in (3.62) is the Fourier symbol of $A_{0,i}$. Thus, from (3.84) and (3.64), the eigenmatrix of CF-relaxation may be expressed as

$$\widehat{S}_i^{\mathrm{CF}}(\theta) = \widehat{S}_i^{\mathrm{F}}(\theta)\Big[I - \widehat{A}_{0,i}(\theta)\Big] = c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big[I - \widehat{A}_{0,i}(\theta)\Big], \tag{3.85}$$

in which $\widehat{A}_{0,i}(\theta)$ is the diagonal matrix holding the Fourier symbols of the harmonics (see (3.60)). This proves (3.78), the first claim of the lemma.

Now let us consider the effect of taking powers of the eigenmatrix. Exploiting the rank-1 structure of $\widehat{S}_i^{\mathrm{CF}}(\theta)$ in (3.85), any power $\nu \in \mathbb{N}$ of the matrix can be computed as

$$\left[\widehat{S}_i^{\mathrm{CF}}(\theta)\right]^\nu = \left(c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big[I - \widehat{A}_{0,i}(\theta)\Big]\right)^\nu, \tag{3.86}$$

$$= c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\left(\mathbf{1}^\top\Big[I - \widehat{A}_{0,i}(\theta)\Big]c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\right)^{\nu-1}\mathbf{1}^\top\Big[I - \widehat{A}_{0,i}(\theta)\Big], \tag{3.87}$$

$$= \left(\mathbf{1}^\top\Big[I - \widehat{A}_{0,i}(\theta)\Big]c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\right)^{\nu-1} c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big[I - \widehat{A}_{0,i}(\theta)\Big], \tag{3.88}$$

$$= \left(\mathbf{1}^\top c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1} - c(\theta)\mathbf{1}^\top\mathbf{1}\right)^{\nu-1}\widehat{S}_i^{\mathrm{CF}}(\theta), \quad \nu \in \mathbb{N}. \tag{3.89}$$

The outstanding problem then is to compute the two functions in (3.89) that occur as inner products, $\mathbf{1}^\top c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}$ and $c(\theta)\mathbf{1}^\top\mathbf{1}$. The latter is simply $c(\theta)\mathbf{1}^\top\mathbf{1} = mc(\theta)$, however, the former requires more careful consideration. Using the expression for the eigenmatrix $\widehat{A}_{0,i}(\theta)$ given in (3.60), and the definition of $c(\theta)$ in (3.65), note that the pth element, $p \in \{0, \ldots, m-1\}$, of the column vector $c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}$ can be rewritten as a

geometric sum as follows

$$\Big[c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big]_p = \frac{1}{m}\frac{1-\big(\lambda_i e^{-\mathrm{i}\theta}\big)^m}{1-\exp\big(-\mathrm{i}\big(\theta+\frac{2\pi p}{m}\big)\big)}, \tag{3.90}$$

$$= \frac{1}{m}\sum_{r=0}^{m-1}\big[\lambda_i \exp\big(-\mathrm{i}\big(\theta+\tfrac{2\pi p}{m}\big)\big)\big]^r. \tag{3.91}$$

Using (3.91), the aforementioned inner product can be written as

$$\mathbf{1}^\top c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1} = \frac{1}{m}\sum_{p=0}^{m-1}\Bigg(\sum_{r=0}^{m-1}\big[\lambda_i \exp\big(-\mathrm{i}\big(\theta+\tfrac{2\pi p}{m}\big)\big)\big]^r\Bigg), \tag{3.92}$$

$$= \frac{1}{m}\sum_{r=0}^{m-1}\Big[\lambda_i e^{-\mathrm{i}\theta}\Big]^r\sum_{p=0}^{m-1}\Big[\exp\big(\tfrac{-2\pi \mathrm{i} r}{m}\big)\Big]^p, \tag{3.93}$$

$$= \frac{1}{m}\sum_{r=0}^{m-1}\Big[\lambda_i e^{-\mathrm{i}\theta}\Big]^r m\delta_{r,0}, \tag{3.94}$$

$$= 1. \tag{3.95}$$

In (3.94), $\delta_{r,0}$ denotes the Kronecker delta function, and it has arisen from simplifying the geometric sum over p in the previous line.

From the definition of $c(\theta)$ in (3.65), the result (3.95), and the fact that $c(\theta)\mathbf{1}^\top\mathbf{1} = mc(\theta)$, it follows immediately that the function appearing in (3.89) is

$$\mathbf{1}^\top c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1} - c(\theta)\mathbf{1}^\top\mathbf{1} = 1 - mc(\theta) = \big(\lambda_i e^{-\mathrm{i}\theta}\big)^m. \tag{3.96}$$

Substituting this result into (3.89) leads to powers of the CF-relaxation eigenmatrix being given by

$$\big[\widehat{S}_i^{\mathrm{CF}}(\theta)\big]^\nu = \big(\lambda_i e^{-\mathrm{i}\theta}\big)^{(\nu-1)m}\widehat{S}_i^{\mathrm{CF}}(\theta), \quad \nu\in\mathbb{N}. \tag{3.97}$$

Finally, let us complete the proof by considering the product of the CF- and F-relaxation eigenmatrices. Using the same rank-1 exploit as used above to compute powers and using (3.96) gives

$$\widehat{S}_i^{\mathrm{CF}}(\theta)\widehat{S}_i^{\mathrm{F}}(\theta) = \Big(c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big[I-\widehat{A}_{0,i}(\theta)\Big]\Big)\Big(c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big), \tag{3.98}$$

$$= c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big(\mathbf{1}^\top\Big[I-\widehat{A}_{0,i}(\theta)\Big]c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big)\mathbf{1}^\top, \tag{3.99}$$

$$= \Big(\mathbf{1}^\top\Big[I-\widehat{A}_{0,i}(\theta)\Big]c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big)c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top, \tag{3.100}$$

$$= \big(\lambda_i e^{-\mathrm{i}\theta}\big)^m\,\widehat{S}_i^{\mathrm{F}}(\theta). \tag{3.101}$$

Combining this result with (3.97) leads immediately to the claimed result of (3.79) that $\left[\widehat{S}_i^{\mathrm{CF}}(\theta)\right]^{\nu}\widehat{S}_i^{\mathrm{F}}(\theta) = \left(\lambda_i e^{-\mathrm{i}\theta}\right)^{m\nu}\widehat{S}_i^{\mathrm{F}}(\theta)$ for $\nu \in \mathbb{N}_0$. ■

Recall that F-relaxation updates F-point values so that they have zero residuals, and leaves C-point values unchanged. Therefore, any successive applications of F-relaxation do not alter the solution since F-point residuals are already zero (see also "*Updating the solution at the F-points with* Φ" [28, Sec. 2]). Therefore, the error propagator S^{F} of F-relaxation (see (3.21)), and that of S_i^{F}, which is the error propagator of F-relaxation on the ith spatial mode (see (3.31)), are *idempotent*. We now prove the same property holds for the eigenmatrix $\widehat{S}_i^{\mathrm{F}}(\theta)$ of F-relaxation.

Corollary 3.8. *The eigenmatrix $\widehat{S}_i^{\mathrm{F}}(\theta)$ for F-relaxation, as given by (3.64), is idempotent,*

$$\widehat{S}_i^{\mathrm{F}}(\theta)\widehat{S}_i^{\mathrm{F}}(\theta) = \widehat{S}_i^{\mathrm{F}}(\theta). \tag{3.102}$$

Furthermore, $\widehat{S}_i^{\mathrm{F}}(\theta)$ has a single eigenvalue of unity, and $m-1$ eigenvalues that are zero; that is, the spectrum of $\widehat{S}_i^{\mathrm{F}}(\theta)$ is the multiset

$$\sigma\left(\widehat{S}_i^{\mathrm{F}}\right) = \{0, \ldots, 0, 1\}. \tag{3.103}$$

Proof. Using the expression for $\widehat{S}_i^{\mathrm{F}}(\theta)$ given by (3.64), its square can be expressed as

$$\begin{aligned}
\widehat{S}_i^{\mathrm{F}}(\theta)\,\widehat{S}_i^{\mathrm{F}}(\theta) &= \left(c(\theta)\left(\widehat{A}_{0,i}(\theta)\right)^{-1}\mathbf{1}\mathbf{1}^{\top}\right)\left(c(\theta)\left(\widehat{A}_{0,i}(\theta)\right)^{-1}\mathbf{1}\mathbf{1}^{\top}\right), && (3.104)\\
&= c(\theta)\left(\widehat{A}_{0,i}(\theta)\right)^{-1}\mathbf{1}\left(\mathbf{1}^{\top}c(\theta)\left(\widehat{A}_{0,i}(\theta)\right)^{-1}\mathbf{1}\right)\mathbf{1}^{\top}, && (3.105)\\
&= \left(\mathbf{1}^{\top}c(\theta)\left(\widehat{A}_{0,i}(\theta)\right)^{-1}\mathbf{1}\right)\widehat{S}_i^{\mathrm{F}}(\theta). && (3.106)
\end{aligned}$$

From (3.95), the bracketed term in (3.106) is one.

Now consider the spectrum of the eigenmatrix. Since $\widehat{S}_i^{\mathrm{F}}(\theta) \in \mathbb{C}^{m\times m}$ has a rank equal to one (see (3.64)), it has $m-1$ eigenvalues that are zero, and one eigenvalue equal to the inner product of the two vectors whose outer product define it. That is, the single non-zero eigenvalue of $\widehat{S}_i^{\mathrm{F}}(\theta)$ is equal to the inner product

$$\left(\mathbf{1}^{\top}\right)\left(c(\theta)\left(\widehat{A}_{0,i}(\theta)\right)^{-1}\mathbf{1}\right) = 1, \tag{3.107}$$

with the equality being due to (3.95). ■

3.4.2.3 Eigenmatrix of error propagation

Having explicitly computed the eigenmatrices that compose the error propagator eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ from (3.53), we now take the next step of explicitly computing the eigenmatrix itself.

Theorem 3.9 (Error propagator eigenmatrix)**.** *The error propagator eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ given by (3.53) may be written as*

$$\widehat{\mathcal{E}}_i(\theta) = f(\theta)\widehat{S}_i^{\mathrm{F}}(\theta), \tag{3.108}$$

where the function $f(\theta)$ is defined as

$$f(\theta) := \left(\lambda_i e^{-\mathrm{i}\theta}\right)^{m\nu} \frac{\lambda_i^m - \mu_i}{e^{\mathrm{i}m\theta} - \mu_i}, \tag{3.109}$$

and $\widehat{S}_i^{\mathrm{F}}(\theta)$ is the eigenmatrix for F-relaxation given in (3.64).

Proof. We begin with the coarse-grid correction component of $\widehat{\mathcal{E}}_i(\theta)$. Invoking the eigenmatrix of interpolation given by (3.63), and exploiting that the eigenmatrix of the coarse-grid operator is simply a scalar (see (3.61)), the eigenmatrix (3.54) of the coarse-grid correction operator is

$$\widehat{\mathcal{K}}_i(\theta) = I_m - \widehat{P}_i(\theta)\left(\widehat{A}_{1,i}(m\theta)\right)^{-1}\widehat{P}_i^\top(\theta)\,\widehat{A}_{0,i}(\theta), \tag{3.110}$$

$$= I_m - \frac{1}{m}\left(\widehat{A}_{1,i}(m\theta)\right)^{-1}\mathbf{1}\mathbf{1}^\top\,\widehat{A}_{0,i}(\theta). \tag{3.111}$$

Substituting this into (3.53), and using the result from (3.79) that $\left[\widehat{S}_i^{\mathrm{CF}}(\theta)\right]^\nu \widehat{S}_i^{\mathrm{F}}(\theta) = \left(\lambda_i e^{-\mathrm{i}\theta}\right)^{m\nu}\widehat{S}_i^{\mathrm{F}}(\theta)$ gives the error propagator eigenmatrix as

$$\widehat{\mathcal{E}}_i(\theta) = \widehat{S}_i^{\mathrm{F}}(\theta)\Big(I_m - \frac{1}{m}\left(\widehat{A}_{1,i}(m\theta)\right)^{-1}\mathbf{1}\mathbf{1}^\top\,\widehat{A}_{0,i}(\theta)\Big)\left[\widehat{S}_i^{\mathrm{CF}}(\theta)\right]^\nu\widehat{S}_i^{\mathrm{F}}(\theta), \tag{3.112}$$

$$= \left(\lambda_i e^{-\mathrm{i}\theta}\right)^{m\nu}\Big(\left[\widehat{S}_i^{\mathrm{F}}(\theta)\right]^2 - \frac{1}{m}\left(\widehat{A}_{1,i}(m\theta)\right)^{-1}\left[\widehat{S}_i^{\mathrm{F}}(\theta)\mathbf{1}\mathbf{1}^\top\,\widehat{A}_{0,i}(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)\right]\Big). \tag{3.113}$$

From (3.64), recall that $\widehat{S}_i^{\mathrm{F}}(\theta) = c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top$. Using this, and the fact that $\mathbf{1}^\top\mathbf{1} = m$, the second term in closed parentheses in (3.113) can be rewritten as

$$\begin{aligned}
\widehat{S}_i^{\mathrm{F}}(\theta)\Big(\mathbf{1}\mathbf{1}^\top\,\widehat{A}_{0,i}(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)\Big) &= \widehat{S}_i^{\mathrm{F}}(\theta)\Big(\mathbf{1}\mathbf{1}^\top\,\widehat{A}_{0,i}(\theta)c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big), && (3.114)\\
&= \widehat{S}_i^{\mathrm{F}}(\theta)\Big(c(\theta)\mathbf{1}\mathbf{1}^\top\mathbf{1}\mathbf{1}^\top\Big), && (3.115)\\
&= mc(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)\mathbf{1}\mathbf{1}^\top, && (3.116)\\
&= mc(\theta)\Big(c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\mathbf{1}\mathbf{1}^\top\Big), && (3.117)\\
&= m^2c(\theta)\widehat{S}_i^{\mathrm{F}}(\theta). && (3.118)
\end{aligned}$$

Substituting this result into (3.113) gives the error propagator eigenmatrix as

$$\begin{aligned}
\widehat{\mathcal{E}}_i(\theta) &= \big(\lambda_i e^{-\mathrm{i}\theta}\big)^{m\nu}\Big(\Big[\widehat{S}_i^{\mathrm{F}}(\theta)\Big]^2 - mc(\theta)\,\big(\widehat{A}_{1,i}(m\theta)\big)^{-1}\,\widehat{S}_i^{\mathrm{F}}(\theta)\Big), && (3.119)\\
&= \big(\lambda_i e^{-\mathrm{i}\theta}\big)^{m\nu}\Big(1 - mc(\theta)\,\big(\widehat{A}_{1,i}(m\theta)\big)^{-1}\Big)\widehat{S}_i^{\mathrm{F}}(\theta), && (3.120)
\end{aligned}$$

where the second equality follows by replacing the square of $\widehat{S}_i^{\mathrm{F}}(\theta)$ with itself, since it is idempotent (see (3.102)), and then pulling out the common factor of $\widehat{S}_i^{\mathrm{F}}(\theta)$.

Finally, using the definition of $c(\theta)$ given in (3.65), and the expression for $\widehat{A}_{1,i}(m\theta)$ given by (3.61), simple algebra reveals that

$$1 - mc(\theta)\,\big(\widehat{A}_{1,i}(m\theta)\big)^{-1} = \frac{\widehat{A}_{1,i}(m\theta) - mc(\theta)}{\widehat{A}_{1,i}(m\theta)} = \frac{\lambda_i^m e^{-\mathrm{i}m\theta} - \mu_i e^{-\mathrm{i}m\theta}}{1 - \mu_i e^{-\mathrm{i}m\theta}} = \frac{\lambda_i^m - \mu_i}{e^{\mathrm{i}m\theta} - \mu_i}. \tag{3.121}$$

Substituting this into (3.120) yields the claimed form of $\widehat{\mathcal{E}}_i(\theta)$ given by (3.108). ■

3.4.3 LFA estimates for error propagation

In this section, we present our main theoretical results on the LFA estimates for error propagation, which are based on the *simple* representation of the error propagator eigenmatrix given in Theorem 3.9. We begin with the norm of this matrix, which is the subject of the following theorem.

Theorem 3.10 (Error propagator eigenmatrix norm)**.** *The ℓ^2-norm of the error propagator eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ given in (3.108) is*

$$\big\|\widehat{\mathcal{E}}_i(\theta)\big\|_2 = |\lambda_i|^{m\nu}\frac{|\lambda_i^m - \mu_i|}{|e^{\mathrm{i}m\theta} - \mu_i|}\sqrt{\frac{1 - |\lambda_i|^{2m}}{1 - |\lambda_i|^2}}. \tag{3.122}$$

Proof. Using the expression for $\widehat{\mathcal{E}}_i(\theta)$ given in (3.108), its squared norm can be expressed as

$$\|\widehat{\mathcal{E}}_i(\theta)\|_2^2 = \rho\big(\widehat{\mathcal{E}}_i^*(\theta)\widehat{\mathcal{E}}_i(\theta)\big), \tag{3.123}$$

$$= \rho\Big(f(\theta)\bar{f}(\theta)\Big(\big(\widehat{S}_i^{\mathrm{F}}(\theta)\big)^*\widehat{S}_i^{\mathrm{F}}(\theta)\Big)\Big), \tag{3.124}$$

$$= |f(\theta)|^2\rho\Big(\big(\widehat{S}_i^{\mathrm{F}}(\theta)\big)^*\widehat{S}_i^{\mathrm{F}}(\theta)\Big). \tag{3.125}$$

Using that $\widehat{S}_i^{\mathrm{F}}(\theta) = c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top$ (see (3.64)), the above can be written as

$$\|\widehat{\mathcal{E}}_i(\theta)\|_2^2 = |f(\theta)|^2\rho\Big(\Big[\mathbf{1}\mathbf{1}^\top\bar{c}(\theta)\big(\widehat{A}_{0,i}^*(\theta)\big)^{-1}\Big]\Big[c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\mathbf{1}^\top\Big]\Big), \tag{3.126}$$

$$= |f(\theta)|^2\rho\Big(\Big[\mathbf{1}^\top\bar{c}(\theta)\big(\widehat{A}_{0,i}^*(\theta)\big)^{-1}c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big]\mathbf{1}\mathbf{1}^\top\Big), \tag{3.127}$$

$$= m|f(\theta)|^2\Big|\mathbf{1}^\top\bar{c}(\theta)\big(\widehat{A}_{0,i}^*(\theta)\big)^{-1}c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big|. \tag{3.128}$$

To arrive at (3.128), we have used the fact that all of the terms appearing before $\mathbf{1}\mathbf{1}^\top$ in (3.127) are scalars, and then that $\rho(\mathbf{1}\mathbf{1}^\top) = \mathbf{1}^\top\mathbf{1} = m$ (see the proof of Corollary 3.8).

The key outstanding issue in evaluating the norm of the eigenmatrix using (3.128) is evaluating the inner product that appears inside the absolute value. To begin, recall the identity given in (3.91) for the pth element in the column vector $c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}$,

$$\Big[c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big]_p = \frac{1}{m}\sum_{r=0}^{m-1}\big[\lambda_i\exp\big(-\mathrm{i}\big(\theta+\tfrac{2\pi p}{m}\big)\big)\big]^r. \tag{3.129}$$

Therefore, it must also be the case that its conjugate transpose satisfies

$$\Big[\mathbf{1}^\top\bar{c}(\theta)\big(\widehat{A}_{0,i}^*(\theta)\big)^{-1}\Big]_p = \frac{1}{m}\sum_{s=0}^{m-1}\big[\bar{\lambda}_i\exp\big(\mathrm{i}\big(\theta+\tfrac{2\pi p}{m}\big)\big)\big]^s. \tag{3.130}$$

Using (3.129) and (3.130), and then reordering the resulting sums gives the inner product as

$$\begin{aligned}
&\mathbf{1}^\top \bar{c}(\theta)\big(\widehat{A}^*_{0,i}(\theta)\big)^{-1} c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1} \\
&= \sum_{p=0}^{m-1}\left(\Big[\mathbf{1}^\top \bar{c}(\theta)\big(\widehat{A}^*_{0,i}(\theta)\big)^{-1}\Big]_p \Big[c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1}\Big]_p\right), && (3.131)\\
&= \frac{1}{m^2}\sum_{p=0}^{m-1}\left(\sum_{r=0}^{m-1}\big[\lambda_i \exp\big(-\mathrm{i}\big(\theta+\tfrac{2\pi p}{m}\big)\big)\big]^r \sum_{s=0}^{m-1}\big[\bar{\lambda}_i \exp\big(\mathrm{i}\big(\theta+\tfrac{2\pi p}{m}\big)\big)\big]^s\right), && (3.132)\\
&= \frac{1}{m^2}\sum_{p=0}^{m-1}\sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\lambda_i^r\bar{\lambda}_i^s e^{\mathrm{i}\theta(s-r)}\exp\left(\tfrac{2\pi \mathrm{i} p(s-r)}{m}\right), && (3.133)\\
&= \frac{1}{m^2}\sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\lambda_i^r\bar{\lambda}_i^s e^{\mathrm{i}\theta(s-r)}\left(\sum_{p=0}^{m-1}\exp\left(\tfrac{2\pi \mathrm{i} p(s-r)}{m}\right)\right). && (3.134)
\end{aligned}$$

Considering the geometric sum in the open parentheses in (3.134) gives

$$\sum_{p=0}^{m-1}\left[\exp\left(\tfrac{2\pi \mathrm{i}(s-r)}{m}\right)\right]^p = \begin{cases} m, & \text{if } (s-r) \bmod m = 0,\\ 0, & \text{else.}\end{cases} \tag{3.135}$$

Notice in (3.134), that $r, s \in \{0,\dots,m-1\}$, and thus, $s-r \in \{1-m,\dots,m-1\}$. Therefore, the only time that $s-r$ can be an integer multiple of m is when $s-r=0$, or $r=s$. As such, the geometric sum of (3.135) is simply equal to $m\delta_{r,s}$, where δ is the Kronecker delta function. Substituting this result into (3.134) gives

$$\begin{aligned}
\mathbf{1}^\top \bar{c}(\theta)\big(\widehat{A}^*_{0,i}(\theta)\big)^{-1} c(\theta)\big(\widehat{A}_{0,i}(\theta)\big)^{-1}\mathbf{1} &= \frac{1}{m^2}\sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\lambda_i^r\bar{\lambda}_i^s e^{\mathrm{i}\theta(s-r)} m\delta_{r,s}, && (3.136)\\
&= \frac{1}{m}\sum_{r=0}^{m-1}\lambda_i^r\bar{\lambda}_i^r, && (3.137)\\
&= \frac{1}{m}\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}. && (3.138)
\end{aligned}$$

The final equality follows here by noting that the expression above is a geometric sum in $|\lambda_i|^2$.

The claimed result of (3.122) follows by substituting into (3.128) the value of $|f(\theta)|^2$ using the definition of $f(\theta)$ given in (3.109), and substituting the quantity given in (3.138), and then finally taking the square root of the result. ■

Recall from (3.52) that $\|\mathcal{E}_i\|_2 = \sup_{\theta\in\Theta^{\text{low}}}\big\|\widehat{\mathcal{E}}_i(\theta)\big\|_2$. That is, the maximization of the norm over $\theta \in \Theta^{\text{low}}$ provides information about the worst-case convergence possible in practice, where, typically, all Fourier modes of all frequencies are present in the data.

Theorem 3.11 (Error propagator norm)**.** *The ℓ^2-norm of the MGRIT error propagator $\mathcal{E}_i$ defined in (3.27), and associated with the ith eigenvector of the time-stepping operators Φ and Ψ, is*

$$\|\mathcal{E}_i\|_2 = \sup_{\theta\in\Theta^{\text{low}}} \left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2 = |\lambda_i|^{m\nu} \frac{|\lambda_i^m - \mu_i|}{1-|\mu_i|} \sqrt{\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}}. \tag{3.139}$$

Furthermore, under the action of $\mathcal{E}_i$, the harmonic space $\mathcal{H}^{\theta}_{\delta t} := \operatorname*{span}_{0\le\alpha<m} \{\boldsymbol{\varphi}_0\left(\theta + \frac{2\pi\alpha}{m}\right)\}$ (see (3.41)) whose error is reduced the least, as measured in the ℓ^2-norm, is the one associated with frequency $\theta = \theta_i^\dagger$, where

$$\theta_i^\dagger := \operatorname*{arg\,max}_{\theta\in\Theta^{\text{low}}} \left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2 = \frac{1}{m}\arg\mu_i, \tag{3.140}$$

in which $\arg\mu_i$ denotes the argument of the complex number μ_i.

Proof. The first equality in (3.139) was already given as (3.52) and follows from the fact that under unitary similarity transform $\mathcal{E}_i \mapsto \operatorname*{diag}_{\theta\in\Theta^{\text{low}}}\left(\widehat{\mathcal{E}}_i(\theta)\right)$. Let us first consider the slowest converging harmonic space, and then return to the second equality in (3.139). From the expression for $\left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2$ given in (3.122), the only dependence on frequency θ is via the term $\frac{1}{|e^{\mathrm{i}m\theta}-\mu_i|}$. Therefore, we have

$$\theta_i^\dagger := \operatorname*{arg\,max}_{\theta\in\Theta^{\text{low}}} \left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2 = \operatorname*{arg\,max}_{\theta\in\Theta^{\text{low}}} \frac{1}{\left|e^{\mathrm{i}m\theta}-\mu_i\right|} = \operatorname*{arg\,min}_{\theta\in\Theta^{\text{low}}} \left|e^{\mathrm{i}m\theta}-\mu_i\right|. \tag{3.141}$$

Furthermore, since Θ^{low} is the continuous frequency space spanning $\left[-\frac{\pi}{m}, \frac{\pi}{m}\right)$ (see (3.40)), introducing the new variable $\vartheta = m\theta$ gives

$$\min_{\theta\in\Theta^{\text{low}}} \left|e^{\mathrm{i}m\theta}-\mu_i\right| = \min_{\vartheta\in[-\pi,\pi)} \left|e^{\mathrm{i}\vartheta}-\mu_i\right|. \tag{3.142}$$

Notice that this quantity is simply the shortest distance from the unit circle to the complex number μ_i that lies inside it (recall that $|\mu_i| < 1$ under Assumption 3.2). By a simple geometric argument, this distance is minimized by the point on the boundary of the unit circle that has the same argument or phase as μ_i; that is, the minimum over $\vartheta \in [-\pi, \pi)$ is achieved at $\vartheta = \arg\mu_i$. Since $\theta = \vartheta/m$, the minimizing frequency over $\theta \in \Theta^{\text{low}}$ is simply $\theta_i^\dagger = \frac{1}{m}\arg\mu_i$, as stated in (3.140).

To evaluate the minimum distance, write the eigenvalue in polar form as $\mu_i = |\mu_i| e^{\mathrm{i} \arg \mu_i}$ and substitute it into the above equation to yield

$$\min_{\theta \in \Theta^{\text{low}}} \left| e^{\mathrm{i}m\theta} - \mu_i \right| = \left| e^{\mathrm{i} \arg \mu_i} - |\mu_i| e^{\mathrm{i} \arg \mu_i} \right| = |1 - |\mu_i|| \left| e^{\mathrm{i} \arg \mu_i} \right| = |1 - |\mu_i|| = 1 - |\mu_i|, \tag{3.143}$$

with the last equality following since $|\mu_i| < 1$. Finally, the result (3.139) follows by evaluating $\left\| \widehat{\mathcal{E}}_i(\theta_i^\dagger) \right\|_2$ from (3.122) using the fact that $\left| e^{\mathrm{i}m\theta_i^\dagger} - \mu_i \right| = 1 - |\mu_i|$. ■

In Section 3.7, we will return to further analyse the results of Theorem 3.11 in the context of advection-dominated problems. For symmetric coarse-grid operators Ψ, which typically arise in the context of diffusion problems, among others, the following corollary specifies the slowest converging space of harmonics.

Corollary 3.12. *Suppose that the coarse-grid time-stepping operator Ψ is symmetric. Then, for the ith eigenmode of Ψ, the harmonic space $\mathcal{H}_{\delta t}^{\theta_i^\dagger}$ that experiences the slowest error reduction over all harmonic spaces has a frequency $\theta_i^\dagger$ given by*

$$\theta_i^\dagger := \arg\max_{\theta \in \Theta^{\text{low}}} \left\| \widehat{\mathcal{E}}_i(\theta) \right\|_2 = \begin{cases} 0, & \mu_i \geq 0, \\ -\dfrac{\pi}{m}, & \mu_i < 0. \end{cases} \tag{3.144}$$

Proof. When Ψ is symmetric, its eigenvalues $(\mu_i)_{i=1}^{n_x}$ are real and the result follows immediately from (3.140) by noting that $\arg \mu_i = 0$ if $\mu_i \geq 0$, and $\arg \mu_i = -\pi$ if $\mu_i < 0$. ■

Given the structure of the eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ in Theorem 3.9, and the result from Theorem 3.11 on $\|\mathcal{E}_i\|_2$, it is straightforward to compute the spectral radius of $\mathcal{E}_i$, which we now present.

Corollary 3.13 (Spectral radius of error propagation)**.** *The spectral radius of the error propagator eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ given in (3.108) is*

$$\rho\big(\widehat{\mathcal{E}}_i(\theta)\big) = |\lambda_i|^{m\nu} \frac{|\lambda_i^m - \mu_i|}{|e^{\mathrm{i}m\theta} - \mu_i|}. \tag{3.145}$$

Furthermore, the spectral radius of the error propagator $\mathcal{E}_i$ given in (3.27) is

$$\rho(\mathcal{E}_i) = \sup_{\theta \in \Theta^{\text{low}}} \rho\big(\widehat{\mathcal{E}}_i(\theta)\big) = |\lambda_i|^{m\nu} \frac{|\lambda_i^m - \mu_i|}{1 - |\mu_i|}. \tag{3.146}$$

Proof. From the expression for $\widehat{\mathcal{E}}_i(\theta)$ given in (3.108), we have its spectral radius simply given as

$$\rho\big(\widehat{\mathcal{E}}_i(\theta)\big) = \rho\big(f(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)\big) = |f(\theta)|\rho\big(\widehat{S}_i^{\mathrm{F}}(\theta)\big) = |f(\theta)|, \tag{3.147}$$

with the final equality following from the fact that the only non-zero eigenvalue of $\widehat{S}_i^{\mathrm{F}}(\theta)$ is one, as per (3.103) of Corollary 3.8. Substituting $f(\theta)$ from its definition given in (3.109) gives the claimed result of (3.145).

Now consider the spectral radius of the error propagator in (3.146). The first equality in (3.146) was already given as (3.51). From (3.145), observe that the eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ with the largest spectral radius is that associated with the frequency $\theta = \theta_i^{\dagger} = \frac{1}{m}\arg\mu_i$, since this is the frequency that minimizes $|e^{\mathrm{i}m\theta} - \mu_i|$ (see the proof of Theorem 3.11). Furthermore, from the proof of Theorem 3.11, we have that the minimum value of $|e^{\mathrm{i}m\theta} - \mu_i|$ is $1 - |\mu_i|$. Substituting this into (3.145) gives (3.146). ■

We now present our final result of this section, which considers the convergence across multiple iterations.

Theorem 3.14 (Norm of powers of error propagator)**.** *The ℓ^2-norm of the pth power of the error propagator eigenmatrix $\widehat{\mathcal{E}}_i(\theta)$ given in (3.108) is*

$$\big\|\big[\widehat{\mathcal{E}}_i(\theta)\big]^p\big\|_2 = \left(|\lambda_i|^{m\nu}\frac{|\lambda_i^m - \mu_i|}{|e^{\mathrm{i}m\theta} - \mu_i|}\right)^p\sqrt{\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}}, \quad p \in \mathbb{N}. \tag{3.148}$$

Furthermore, the ℓ^2-norm of the pth power of the error propagator $\mathcal{E}_i$ given in (3.27) is

$$\big\|\mathcal{E}_i^p\big\|_2 = \sup_{\theta\in\Theta^{\mathrm{low}}}\big\|\big[\widehat{\mathcal{E}}_i(\theta)\big]^p\big\|_2 = \left(|\lambda_i|^{m\nu}\frac{|\lambda_i^m - \mu_i|}{1-|\mu_i|}\right)^p\sqrt{\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}}, \quad p \in \mathbb{N}. \tag{3.149}$$

Proof. Recall from (3.108) that $\widehat{\mathcal{E}}_i(\theta) = f(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)$, and therefore, one has for $p \in \mathbb{N}$

$$\big[\widehat{\mathcal{E}}_i(\theta)\big]^p = \big[f(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)\big]^p = \big[f(\theta)\big]^p\big[\widehat{S}_i^{\mathrm{F}}(\theta)\big]^p = \big[f(\theta)\big]^p\widehat{S}_i^{\mathrm{F}}(\theta), \tag{3.150}$$

with the last equality following from the idempotence of $\widehat{S}_i^{\mathrm{F}}(\theta)$, as per (3.102) of Corollary 3.8. Therefore, it follows that

$$\begin{aligned}
\big[\widehat{\mathcal{E}}_i^*(\theta)\big]^p\big[\widehat{\mathcal{E}}_i(\theta)\big]^p &= \Big(\big[\bar{f}(\theta)\big]^p\big(\widehat{S}_i^{\mathrm{F}}(\theta)\big)^*\Big)\Big(\big[f(\theta)\big]^p\widehat{S}_i^{\mathrm{F}}(\theta)\Big), && (3.151)\\
&= |f(\theta)|^{2p-2}\Big(\bar{f}(\theta)\big(\widehat{S}_i^{\mathrm{F}}(\theta)\big)^*\Big)\Big(f(\theta)\widehat{S}_i^{\mathrm{F}}(\theta)\Big), && (3.152)\\
&= |f(\theta)|^{2p-2}\Big[\widehat{\mathcal{E}}_i^*(\theta)\,\widehat{\mathcal{E}}_i(\theta)\Big]. && (3.153)
\end{aligned}$$

Now, considering the norm of p powers of the eigenmatrix and using (3.153) gives

$$\left\| \left[\widehat{\mathcal{E}}_i(\theta)\right]^p \right\|_2^2 = \rho\big(\left[\widehat{\mathcal{E}}_i^*(\theta)\right]^p \left[\widehat{\mathcal{E}}_i(\theta)\right]^p\big), \tag{3.154}$$

$$= |f(\theta)|^{2p-2} \rho\Big(\widehat{\mathcal{E}}_i^*(\theta)\widehat{\mathcal{E}}_i(\theta)\Big), \tag{3.155}$$

$$= |f(\theta)|^{2p-2} \left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2^2. \tag{3.156}$$

The claimed result for $\left\| \left[\widehat{\mathcal{E}}_i(\theta)\right]^p \right\|_2$ given in (3.148) follows from substituting $f(\theta)$ into (3.156) using (3.109), substituting $\left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2$ into (3.156) using (3.122), and then taking the square root of the result.

Now consider the result for the error propagator given by (3.149). The first equality here follows from the unitary similarity transform in (3.48) (see also (3.52)). Now observe from (3.148) that the eigenmatrix $\left[\widehat{\mathcal{E}}_i(\theta)\right]^p$ with the largest norm is that with frequency $\theta = \theta_i^\dagger = \frac{1}{m}\arg\mu_i$. This is because $\theta_i^\dagger$ maximizes the quantity $\frac{1}{|e^{\mathrm{i}m\theta}-\mu_i|^p}$, which is the only dependence that $\left\| \left[\widehat{\mathcal{E}}_i(\theta)\right]^p \right\|_2$ has on θ (see the proof of Theorem 3.11 for related discussion). Furthermore, from the proof of Theorem 3.11, it follows that the maximum value of $\frac{1}{|e^{\mathrm{i}m\theta}-\mu_i|^p}$ is $\frac{1}{(1-|\mu_i|)^p}$. Substituting this into (3.148) leads to the claimed result in (3.149). ■

3.5 Rigorous Fourier analysis for time-periodic problems

One can define a time-periodic MGRIT solver that uses either a time-periodic coarse-grid problem or an initial-value coarse-grid problem (see, e.g., [43, 49]). Our LFA theory applies *rigorously* or exactly for two-level MGRIT solvers that employ a time-periodic coarse-grid problem.

In essence, such a time-periodic MGRIT algorithm and its error propagator can be constructed in exactly the same way we constructed those for the initial-value problem previously in Sections 3.2 and 3.3. However, this is done with the key distinction that all occurrences of the lower shift matrix L_n given in (3.7) are replaced with its circulant analogue:

$$L_n^{\text{periodic}} = \begin{bmatrix} 0 & & & 1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{3.157}$$

This has the effect of replacing initial-value boundary conditions present in the space-time and CF-relaxation operators with time-periodic boundary conditions instead. For

example, rather than the initial-value space-time operators $A_0 = I_{n_t} \otimes I_{n_x} - L_{n_t} \otimes \Phi$ and $A_1 = I_{n_t/m} \otimes I_{n_x} - L_{n_t/m} \otimes \Psi$ as defined, respectively, in (3.5) and (3.6), one should use the time-periodic space-time operators $A_0^{\text{periodic}} = I_{n_t} \otimes I_{n_x} - L_{n_t}^{\text{periodic}} \otimes \Phi$ and $A_1^{\text{periodic}} = I_{n_t/m} \otimes I_{n_x} - L_{n_t/m}^{\text{periodic}} \otimes \Psi$.

Then, our LFA theory of Section 3.4 yields exact convergence results for time-periodic problems solved with the above-described algorithm, provided two key changes are made. First, we no longer need to work with the infinite-grid assumption that $n_t \to \infty$. However, none of our results explicitly depend on n_t, and, so, this change does not affect any of the resulting formulae. Second, the Fourier modes $\boldsymbol{\varphi}_\ell(\theta)$ given in (3.37) should sample the frequency θ discretely rather than continuously. More specifically, define $\Theta^{\text{low,discrete}}$ to be the set of n_t/m equidistant frequencies spanning the interval $[-\pi/m, \pi/m)$, that is, the discrete analogue of the continuous set Θ^{low} from (3.40). Then, for any $\theta \in \Theta^{\text{low,discrete}}$, the spectral radius $\rho\big(\widehat{\mathcal{E}}_i(\theta)\big)$ (see (3.145)), norm $\big\|\widehat{\mathcal{E}}_i(\theta)\big\|_2$ (see (3.122)), and the norm of powers $\big\|\big[\widehat{\mathcal{E}}_i(\theta)\big]^p\big\|_2$ (see (3.148)) hold. Moreover, the maximum of these quantities over $\theta \in \Theta^{\text{low,discrete}}$ yields exact results for $\mathcal{E}_i$ for any finite n_t:

$$\rho(\mathcal{E}_i) = \max_{\theta \in \Theta^{\text{low,discrete}}} \rho\big(\widehat{\mathcal{E}}_i(\theta)\big), \tag{3.158}$$

$$\|\mathcal{E}_i\|_2 = \max_{\theta \in \Theta^{\text{low,discrete}}} \big\|\widehat{\mathcal{E}}_i(\theta)\big\|_2, \tag{3.159}$$

$$\big\|[\mathcal{E}_i]^p\big\|_2 = \max_{\theta \in \Theta^{\text{low,discrete}}} \big\|\big[\widehat{\mathcal{E}}_i(\theta)\big]^p\big\|_2. \tag{3.160}$$

That is, the previous values of $\rho(\mathcal{E}_i)$, $\|\mathcal{E}_i\|_2$, and $\big\|[\mathcal{E}_i]^p\big\|_2$ given in (3.146), (3.139), and (3.149), respectively, do not hold for finite n_t because they were maximized over continuous θ, which corresponds to $n_t \to \infty$ in our above formulation. Note, however, that the results for $\rho\big(\mathcal{E}_i\big)$, $\big\|\mathcal{E}_i\big\|_2$, and $\big\|\big[\mathcal{E}_i\big]^p\big\|_2$ given in (3.146), (3.139), and (3.149) yield upper bounds on the finite n_t expressions in (3.158), (3.159), and (3.160), respectively.

The reason that the theory is exact for time-periodic problems is because the operators appearing in the error propagator are not just Toeplitz as they were for the initial-value problem, but they are also circulant. Therefore, the periodic Fourier modes $\boldsymbol{\varphi}_\ell(\theta)$—with θ discretely sampled at n_t equidistant frequencies—are eigenfunctions for any finite n_t, and not only formally in the limit as $n_t \to \infty$. See also [100] for another example of an LFA theory for initial-value problems that holds rigorously for time-periodic problems. More broadly, see [99, Sec. 3.4.4] for a discussion on the link between rigorous and local Fourier analysis.

As a demonstration of the above discussion, and to verify the correctness of the LFA theory, we now show a numerical experiment. Specifically, we consider the eigenvalues and singular values for the error propagator $\mathcal{E}_i^{\text{periodic}}$ corresponding to a scalar, time-periodic

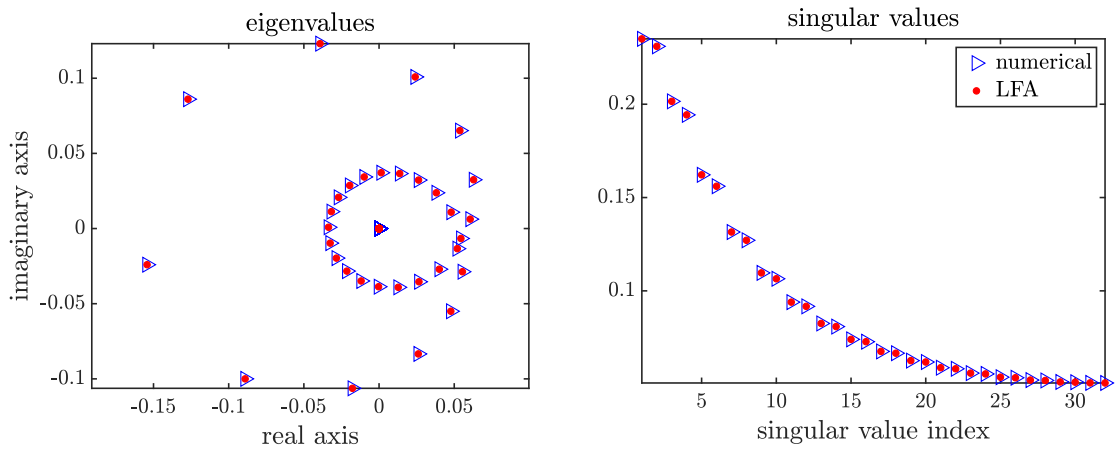

FIGURE 3.1: Quantities of interest of $\mathcal{E}_i^{\text{periodic}}$ for a scalar, time-periodic problem. Direct numerical calculations are compared with LFA predictions. The fine- and coarse-grid eigenvalues are $\lambda_i = 0.749 + 0.045\text{i}$ and $\mu_i = 0.510 + 0.404\text{i}$, respectively. The other problem parameters are $n_t = 256$, $m = 8$, and $\nu = 1$. **Left:** Eigenvalues. **Right:** Non-zero singular values sorted in descending order.

problem with arbitrarily chosen fine- and coarse-grid eigenvalues of $\lambda_i = 0.749 + 0.045\text{i}$ and $\mu_i = 0.510 + 0.404\text{i}$, respectively. That is, this test problem does not correspond to the discretization of any ODE in particular, but has just been chosen to demonstrate the accuracy of the theory.

From the discussion above, the time-periodic error propagator is the block diagonal matrix $\mathcal{E}_i^{\text{periodic}} = \underset{\theta\in\Theta^{\text{low,discrete}}}{\text{diag}} \left(\widehat{\mathcal{E}}_i(\theta)\right) \in \mathbb{C}^{n_t\times n_t}$. Therefore, the n_t eigenvalues and singular values of $\mathcal{E}_i^{\text{periodic}}$ are given by the union of the m eigenvalues and m singular values, respectively, of the n_t/m diagonal blocks $\widehat{\mathcal{E}}_i(\theta) \in \mathbb{C}^{m\times m}$ for $\theta \in \Theta^{\text{low,discrete}}$. Recall from (3.108) of Theorem 3.9 that $\widehat{\mathcal{E}}_i(\theta) = f(\theta)\widehat{\mathcal{S}}_i^{\text{F}}(\theta)$. From Corollary 3.8, $\widehat{\mathcal{S}}_i^{\text{F}}(\theta)$ has $m-1$ eigenvalues equal to zero and a single eigenvalue of one; therefore, it follows that $\widehat{\mathcal{E}}_i(\theta)$ has $m-1$ eigenvalues equal to zero and a single eigenvalue equal to $f(\theta)$. For our test problem, this set of m eigenvalues is plotted in the left panel of Figure 3.1 for the n_t/m values of $\theta \in \Theta^{\text{low,discrete}}$. Also plotted are the eigenvalues determined by direct numerical calculation (i.e., constructing numerically $\mathcal{E}_i^{\text{periodic}}$ and computing its eigenvalues). We find exact agreement between the two.

In terms of singular values, recall that $\widehat{\mathcal{E}}_i(\theta)$ has a rank equal to one (since $\widehat{\mathcal{S}}_i^{\text{F}}(\theta)$ has a rank of one), it follows that $\widehat{\mathcal{E}}_i(\theta)$ has one non-zero singular value. This singular value is therefore equal to the norm of the matrix, which is given by (3.122) of Theorem 3.10. The right panel of Figure 3.1 shows this norm for the n_t/m values of $\theta \in \Theta^{\text{low,discrete}}$. Plotted also are the non-zero singular values of $\mathcal{E}_i$ that have been determined numerically. Again, there is exact agreement between the two.

3.6 Discussion of theory

We now provide some discussion on the LFA theory derived in Section 3.4. Specifically, some important implications of the theory are described in Section 3.6.1. Section 3.6.2 presents connections to existing literature. Finally, the suitability of LFA for initial-value problems in discussed in Section 3.6.3.

3.6.1 Closed-form determination of quantities of interest

LFA convergence results for multigrid methods are typically presented in one of three ways. Commonly, numerical values are presented for quantities such as convergence factors, norms, and so on (see, e.g., [99, 39, 38, 26, 100]). Perhaps less common, approximate, asymptotic results are presented to make conclusions about the solver, such as when the mesh size goes to zero, for example (see, e.g., [108, 103, 3]). These typically offer significantly more insight than purely numerical calculations. Rarer still, convergence results are presented in closed form as they were in Section 3.4 (see, e.g., [98, Sec. 8.1] and [77]). Closed-form results are rare because most often the underlying calculations are either too complicated to carry out analytically, or the resulting expressions are so complicated that doing so would provide little insight.

Our derivation of *simple*, closed-form expressions for the convergence properties of MGRIT is novel for several reasons. First, we gain significant insight into the underlying convergence behaviour of the algorithm that is not possible from purely numerical computation. This is a benefit of our MGRIT LFA theory over the semi algebraic approach of [38] and the LFA theory described in [26], since they are purely numerical. A second reason is that, if one wants to estimate convergence rates, then evaluating our formulae is significantly less computationally intensive than a fully numerical approach. For example, the typical process is one of discretely sampling $\theta \in \Theta^{\text{low}}$, then numerically forming the eigenmatrices $\widehat{\mathcal{E}}_i(\theta) \in \mathbb{C}^{m\times m}$ for these discrete θ, computing their quantities of interest, such as their spectral radii or norms, and then selecting the maximum over the discrete sample. In contrast, we have simple, closed-from expressions for the maxima of the norm and spectral radii of $\widehat{\mathcal{E}}_i(\theta)$ over *continuous frequency space* $\theta \in \Theta^{\text{low}}$. Thus, it is clearly significantly less computationally expensive to estimate convergence rates using our expressions rather than the standard numerical approach. Another potential benefit of our closed-form maximization is that sometimes it may be difficult to numerically maximize a quantity of interest over $\theta \in \Theta^{\text{low}}$ if it is non-smooth with respect to θ. For example, we will see such examples of non-smoothness in Section 3.7, and it has been observed in the LFA study of other time-parallel multigrid solvers [100, Sec. 3.3.2].

3.6.2 Comparison to existing literature

The expressions we obtain through LFA for norms of the MGRIT error propagators bear a close resemblance to results in [28, 88, 51] that were obtained by other means. There are, however, some differences. First, [28, 88, 51] analysed error propagation on the coarse grid rather than the fine grid as we have. Second, [28, 88, 51] give equalities and/or bounds for the norm of the error propagator for finite n_t. In contrast, our results provide neither equalities nor bounds for finite n_t, but only approximations because they have been developed through the framework of LFA in the limit that $n_t \to \infty$. Thus, our results do not take into account the number of time points since they are derived under the infinite-grid assumption of LFA. Our results are exact, however, for the time-periodic case, while those of [28, 88, 51] are not.

This close resemblance is perhaps best seen by looking at [92], which serves as a companion article to [88], because it provides several results for fine-grid error propagation. Specifically, adopting our notation, and imposing Assumption 3.1 (that the eigenvectors of Φ and Ψ are unitary), [92, Cor. 3] states that the ℓ^2-norm of the MGRIT error propagator $\mathcal{E}$ from (3.12) for finite n_t is equal to[5]

$$\text{F-relaxation:} \quad \|\mathcal{E}\|_2 = \max_{1\le i\le n_x} \frac{|\lambda_i^m - \mu_i|}{1 - |\mu_i| + \mathcal{O}(1/N_c)} \sqrt{\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}}, \tag{3.161}$$

$$\text{FCF-relaxation:} \quad \|\mathcal{E}\|_2 = \max_{1\le i\le n_x} |\lambda_i|^m \frac{|\lambda_i^m - \mu_i|}{1 - |\mu_i| + \mathcal{O}(1/N_c)} \sqrt{\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}}, \tag{3.162}$$

where N_c is the number of time points on the coarse grid, which is slightly different from the quantity n_t/m that we have used. It is important to stress that the expressions (3.161) and (3.162) represent genuine equalities for finite n_t, up to the $\mathcal{O}(1/N_c)$ terms, unlike the LFA approximations considered in Section 3.4 which hold in the limit of infinite n_t. The equalities (3.161) and (3.162) were derived through the application of block Toeplitz theory. Closely related bounds for coarse-grid error propagation were derived in [28, 51] by exploiting the Toeplitz structure of the coarse-grid error propagator and applying a Hölder inequality for the ℓ^2-norm.

Recalling our LFA approximation of $\|\mathcal{E}_i\|_2$ given by (3.139), and its relation to $\|\mathcal{E}\|_2$ given in (3.33), our LFA analysis gives the following *approximation* for finite n_t (for clarity, we

[5] The formulation of $\mathcal{E}$ in [92] appears differently to our $\mathcal{E}$ given by (3.12). However, they both represent the error propagator of the MGRIT algorithm, and are therefore equivalent (after taking account of the fact that we use different conventions for the sizes of the fine and coarse grids, that is).

write the LFA approximation explicitly here),

$$\|\mathcal{E}\|_2 = \max_{1\leq i\leq n_x} \|\mathcal{E}_i\|_2 \approx \max_{1\leq i\leq n_x} \sup_{\theta\in\Theta^{\text{low}}} \left\|\widehat{\mathcal{E}}_i(\theta)\right\|_2 = \max_{1\leq i\leq n_x} |\lambda_i|^{m\nu} \frac{|\lambda_i^m - \mu_i|}{1-|\mu_i|} \sqrt{\frac{1-|\lambda_i|^{2m}}{1-|\lambda_i|^2}}. \tag{3.163}$$

Recall from Section 3.3.1 that $\nu \in \mathbb{N}_0$ represents the number of CF-relaxations included in the pre-relaxation, so that $\nu = 0$ and $\nu = 1$ correspond to F- and FCF-relaxation, respectively. Comparing our LFA approximation (3.163) with the expressions (3.161) and (3.162), it differs only by the small perturbation of $\mathcal{O}(1/N_c)$ that appear in the latter equations. As $n_t \to \infty$, and thus $N_c = \mathcal{O}(n_t/m) \to \infty$, the expressions are equivalent. In other words, since, (3.161) and (3.162) are valid for any value of n_t, they are consistent with our LFA approximation (3.163) holding exactly for $\|\mathcal{E}\|_2$ as $n_t \to \infty$.

This consistency provides independent verification that our LFA theory of Section 3.4 is correct. It is also interesting to consider that our theory, which, in a sense, is much less technical than the block Toeplitz analysis of [88, 92], yields effectively equivalent results for the type of Φ and Ψ we have considered here (see Assumptions 3.1 and 3.2). That is, the results (3.161)–(3.162) arguably offer little more insight than our theory, particularly since one is only interested in *large* values of n_t in practice. In particular, consider the subject of the poor convergence of advection-dominated problems, which is arguably the longest-standing issue in the MGRIT/Parareal community. While there are possibly additional reasons for this poor convergence that our theory cannot explain, our simpler LFA theory can offer as much insight into the fundamental difficulties for advection-dominated problems (see Section 3.7) as the more complicated theory of [88, 92]. Moreover, our theory is able to draw connections to related issues for spatial multigrid solvers, and it is not immediately clear whether that of [88, 92] can. On the other hand, the theory of [88, 92] is more practical for estimating convergence rates for finite n_t (tight bounds for the $\mathcal{O}(1/N_c)$ constants are given in [88, 92]), and furthermore, it leads to results for much more general Φ and Ψ than considered here, such as when they are not diagonalizable.

3.6.3 On the effects of non-normality and the suitability of LFA

Time-stepping is used as the relaxation scheme in MGRIT, and thus the algorithm is nilpotent for initial-value problems: In exactly $k = \mathcal{O}(n_t/m)$ iterations, the algorithm sequentially time-steps the initial condition across the time domain and thus converges to the exact solution.[6] Equivalently, the error propagator $\mathcal{E}_i$ is nilpotent, which means $\mathcal{E}_i^k = 0$, and that the only eigenvalue of $\mathcal{E}_i$ is zero. The nilpotency of $\mathcal{E}_i$ has also been

[6] The exact number is $k = \frac{1}{\nu+1}\frac{n_t}{m}$, for a pre-relaxation consisting of an F-relaxation followed by $\nu \in \mathbb{N}_0$ sweeps of CF-relaxation.

referred to as the non-normality of the iteration [38, 26], since the nilpotency arises as a consequence of the non-normality of the space-time discretization A_0.

Recalling that LFA ignores the effects of boundary conditions, it is not immediately obvious that it should offer robust or relevant predictions for problems in which ignoring boundary effects has a dramatic effect on the problem/solver. In particular, our LFA theory clearly does not recover the fact that $\mathcal{E}_i$ is nilpotent. Theorem 3.14 states that $\|\mathcal{E}_i^p\|_2 > 0$, $\forall p \in \mathbb{N}$, and Corollary 3.13 states that $\rho(\mathcal{E}_i) > 0$. These discrepancies arise due to LFA disregarding the effects of boundary conditions. In fact, not only have we disregarded the effects of boundary conditions, we have analysed the time-periodic problem in place of the initial-value problem, recalling from Section 3.5 that the analysis holds exactly for time-periodic problems. The fact that LFA does not recover the nilpotency of $\mathcal{E}_i$ is a reflection that time-stepping from an initial approximation at $t = 0$ is not an exact solver for time-periodic problems, as it is for initial-value problems. However, this result is completely consistent with the mission statement of LFA, which is describing asymptotic convergence behaviour under the assumption that boundary behaviour is not significant.

The inability of LFA to capture nilpotency and/or the effects of non-normality has been discussed several times in the literature, and often critically so. Moreover, this observation has been used, at least in part, to justify more rigorous convergence theories that can capture such effects [38, 26, 71]. Here, however, we make a contrary argument about the applicability of LFA: *The fact that LFA cannot capture the nilpotency of MGRIT for initial-value problems is of no practical significance.* The non-normality of MGRIT for initial-value problems is a direct consequence of the relaxation scheme being an exact solver in k iterations. Therefore, if one is in a regime where non-normality effects strongly influence convergence of the algorithm, the mechanism driving convergence is time-stepping itself. Given this, it is important to recall that the underlying context of MGRIT for initial-value problems is to find the solution in a faster wall-clock time than the exact solver that is time-stepping. Therefore, if a sufficient level of convergence is reached only when the number of iterations is $\sim k = \frac{1}{\nu+1}\frac{n_t}{m}$, then MGRIT cannot possibly achieve speed-up over sequential time-stepping. Moreover, in practice, it is most often the case that parallel speed-up requires very few iterations due to the low parallel efficiency of MGRIT. Thus, a practically useful convergence theory need not capture non-normality effects. In summary, for a theory to be practically useful, it need not recover that near-exact convergence is reached in $\sim k$ iterations because this is trivially known, and such a regime is of no practical interest.

We conclude our discussion by noting that not only does our LFA theory provide accurate and relevant convergence information (see Section 3.6.2), but more broadly Fourier analysis has been demonstrated to yield accurate and practically useful information about the

convergence behaviour of multigrid-in-time methods for initial-value problems [100, 45, 37, 26].

3.7 Characteristic components

In this section, we apply the LFA theory of Section 3.4 to shed further light on the poor convergence of MGRIT for advection-dominated problems. Specifically, we present theoretical arguments in Section 3.7.1, with supporting numerical results given in Section 3.7.2. Finally, a discussion on the implications of our results is given in Section 3.7.3.

3.7.1 Theoretical arguments

In this section, we concern ourselves with the solution of the constant-coefficient, one-dimensional advection-diffusion problem

$$\mathcal{A}u := \frac{\partial u}{\partial t} + \alpha\frac{\partial u}{\partial x} - \beta\frac{\partial^2 u}{\partial x^2} = 0, \quad \beta \geq 0. \tag{3.164}$$

Specifically, we provide arguments regarding the convergence of MGRIT for discretizations of this problem via making a Fourier ansatz in space and employing our LFA theory in time. To this end, suppose that the space-time discretizations A_0 and A_1 given by (3.5) and (3.6), respectively, correspond to discretizations of $\mathcal{A}$ on space-time meshes with a mesh spacing of h in the x-direction.

Now consider the space-time discretizations A_0 and A_1 on the infinite space-time meshes $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$, respectively, which are defined by

$$\boldsymbol{M}_\ell := \big\{(x,t) = (jh, km^\ell\delta t)\colon j \in \mathbb{Z}, k \in \mathbb{N}_0\big\}, \quad \ell \in \{0,1\}. \tag{3.165}$$

On $\boldsymbol{M}_\ell$ we also consider the space-time Fourier modes

$$\varrho_\ell(\omega,\theta) := \exp\left(\frac{\mathrm{i}\omega x}{h}\right)\exp\left(\frac{\mathrm{i}\theta t}{m^\ell\delta t}\right), \quad (x,t) \in \boldsymbol{M}_\ell, \quad (\theta,\omega) \in [-\pi,\pi) \times \Theta_\ell, \tag{3.166}$$

with continuous frequencies θ and ω, and Θ_ℓ defined as previously (see (3.38)). The space-time Fourier symbol $\widetilde{A}_\ell(\omega,\theta)$ of A_ℓ is defined implicitly via $A_\ell\boldsymbol{\varrho}_\ell(\omega,\theta) = \widetilde{A}_\ell(\omega,\theta)\boldsymbol{\varrho}_\ell(\omega,\theta)$. Let $\lambda(\omega)$ and $\mu(\omega)$ be the Fourier symbols of the fine- and coarse-grid time-stepping operators Φ and Ψ, respectively. Then, the space-time Fourier symbols of the fine- and coarse-grid discretizations are

$$\widetilde{A}_0(\omega,\theta) = 1 - \lambda(\omega)e^{-\mathrm{i}\theta}, \quad \widetilde{A}_1(\omega,m\theta) = 1 - \mu(\omega)e^{-\mathrm{i}m\theta}. \tag{3.167}$$

Now, by grouping together space-time Fourier modes, the MGRIT error propagator $\mathcal{E}$ may be block diagonalized, analogously to how it was first block diagonalized in space in Section 3.3.3, and then in time in Section 3.4. Specifically, under a similarity transform we have

$$\mathcal{E} \mapsto \operatorname*{diag}_{(\omega,\theta)\in[-\pi,\pi)\times\Theta^{\text{low}}} \Big(\widehat{\mathcal{E}}(\omega,\theta)\Big), \tag{3.168}$$

where $\widehat{\mathcal{E}}(\omega,\theta) \in \mathbb{C}^{m\times m}$ is the error propagation eigenmatrix associated with the m harmonic space-time modes having frequencies $\big(\omega, \theta + \frac{2\pi p}{m}\big)$, $p \in \{0,\dots,m-1\}$ (see Definition 3.4). The space of low frequencies Θ^{low} in (3.168) is as in (3.40).

Using our LFA theory from Section 3.4, we may compute the spectral radius of the diagonal blocks $\widehat{\mathcal{E}}(\omega,\theta)$ as follows. Note that alternatively we could develop an analogous expression for the norm, but considering the spectral radius is sufficient for our purposes.

Lemma 3.15 (Spectral radius of space-time error propagator)**.** *The spectral radius of the diagonal blocks in the space-time error propagation matrix* (3.168) *may be written as*

$$\rho\big(\widehat{\mathcal{E}}(\omega,\theta)\big) = \big|\lambda(\omega)\big|^{m\nu} \left| \frac{\widetilde{A}_1(\omega,m\theta) - \widetilde{A}_1^{\text{ideal}}(\omega,m\theta)}{\widetilde{A}_1(\omega,m\theta)} \right|, \quad (\omega,\theta)\in[-\pi,\pi)\times\Theta^{\text{low}}, \tag{3.169}$$

in which $\widetilde{A}_1^{\text{ideal}}(\omega,\theta) = 1 - \big[\lambda(\omega)\big]^m e^{-\mathrm{i}\theta}$ *is the Fourier symbol of the coarse-grid space-time discretization that uses the ideal coarse-grid operator* $\Psi_{\text{ideal}} = \Phi^m$.

Proof. The diagonal block of (3.168) for a fixed spatial frequency $\omega = \omega_i$ is $\widehat{\mathcal{E}}(\omega_i,\theta)$, and is associated with fine- and coarse-grid time-stepping eigenvalues $\lambda_i = \lambda(\omega_i)$ and $\mu_i = \mu(\omega_i)$, respectively. Therefore, $\widehat{\mathcal{E}}(\omega_i,\theta)$ is simply equal to $\widehat{\mathcal{E}}_i(\theta)$, which is the eigenmatrix associated with the ith spatial eigenmode that we analysed previously in Sections 3.3 and 3.4. Recall from (3.145), that the spectral radius of $\widehat{\mathcal{E}}_i(\theta)$ is simply

$$\rho\big(\widehat{\mathcal{E}}_i(\theta)\big) = |\lambda_i|^{m\nu} \left| \frac{\lambda_i^m - \mu_i}{e^{\mathrm{i}m\theta} - \mu_i} \right|. \tag{3.170}$$

The fraction inside the absolute value may be re-written in the form

$$\frac{\lambda_i^m - \mu_i}{e^{\mathrm{i}m\theta} - \mu_i} = \frac{\big(1 - \mu_i e^{-\mathrm{i}m\theta}\big) - \big(1 - \lambda_i^m e^{-\mathrm{i}m\theta}\big)}{1 - \mu_i e^{-\mathrm{i}m\theta}} = \frac{\widetilde{A}_1(\omega_i,m\theta) - \widetilde{A}_1^{\text{ideal}}(\omega_i,m\theta)}{\widetilde{A}_1(\omega_i,m\theta)}. \tag{3.171}$$

Substituting (3.171) into (3.170) yields the claimed expression of (3.169). ■

Ignore, for the moment, the damping factor of $\big|\lambda(\omega)\big|^{m\nu}$ in (3.169) due to ν sweeps of CF-relaxation, which has little effect on the MGRIT convergence of spatial modes that decay slowly under time-stepping, $\big|\lambda(\omega)\big| \approx 1$. Observe from (3.169) that MGRIT convergence

is determined by the *relative difference* between the space-time Fourier symbols of the coarse-grid discretization and the coarse-grid discretization using the ideal coarse-grid operator. This observation will provide us with significant insight into the convergence of MGRIT for (3.164), as we now discuss. A quantity closely related to (3.169) has long been known in the spatial multigrid community, particularly to those working on the solution of advection-dominated problems. More specifically, consider the steady state advection-diffusion PDE $\mathcal{L}u := \boldsymbol{\alpha} \cdot \nabla u - \beta \Delta u = 0$, with fine- and coarse-grid discretizations given by L_h and L_H, respectively. Here, $\boldsymbol{\alpha}$ is a multidimensional wave-speed, and $\beta \geq 0$ is still the diffusivity as in (3.164). Then, under some assumptions regarding the effects of relaxation and the intergrid transfer operators, a key quantity governing the convergence of the two-grid spatial method is

$$\zeta(\boldsymbol{\omega}) = \left| \frac{\widetilde{L}_H(\boldsymbol{\omega}) - \widetilde{L}_h(\boldsymbol{\omega})}{\widetilde{L}_H(\boldsymbol{\omega})} \right|, \tag{3.172}$$

where $\boldsymbol{\omega}$ is a multi-component spatial frequency. This quantity may be derived using asymptotic two-grid Fourier analysis [108], and via a more simplistic mode analysis known as 'first differential approximation analysis' [12]. Notice that (3.172) is the *relative difference* of the Fourier symbols of the fine- and coarse-grid discretizations, and is therefore closely related to the relative difference of Fourier symbols in (3.169).

The significance of (3.172) is that it can be used to explain the poor convergence experienced by spatial multigrid solvers on advection-dominated problems. Specifically, it can be shown that when $\beta \gg h$, so that the operators L_h and L_H are close to elliptic, $\zeta(\boldsymbol{\omega})$ is small for all Fourier components [10, 108]. However, in the non-elliptic, singularly perturbed case where $\beta \to 0^+$, it can be shown that $\zeta(\boldsymbol{\omega})$ is small for almost all Fourier components $\boldsymbol{\omega}$, except for so-called *characteristic components*, for which $\zeta(\boldsymbol{\omega}) \to$ a mesh-independent constant. The exact value of this constant depends on the problem at hand, but often it is not much smaller than one, thus resulting in slow convergence of the two-grid method. The conclusion is: Two-grid performance suffers because characteristic components do not receive an adequate coarse-grid correction; under certain assumptions, one can argue that of all characteristic components, it is the smoothest ones that convergence is slowest for [10, 108].

Characteristic components/modes are Fourier modes that vary extremely slowly—relative to the grid spacing—along the direction of characteristics, but that are free to vary in the direction normal to characteristics. The *smooth* characteristic components described above are those which vary slowly—relative to the grid spacing—in the direction normal to characteristics. The particular significance of characteristic components with respect to (3.172), is that in the $\beta \to 0^+$ limit, the coarse-grid symbol $\widetilde{\mathcal{L}}_H \approx \widetilde{\mathcal{L}}$ vanishes up to some

mesh-dependent terms, since $\widetilde{\mathcal{L}}$ vanishes for characteristic components. For elliptic problems, where multigrid convergence is significantly more robust than advection-dominated problems, there do not exist any directions for which the symbol vanishes [108], and, so, this is not a problem. The poor convergence of characteristic modes for the spatial multigrid solution of advection-related problems was first described in [10, Sec. 5.1], and has since been described in many other contexts [12, 108, 99, 103, 3].

Our PDE problem (3.164) has the same form as the above steady state problem, provided one interprets time as another spatial coordinate and considers anisotropic diffusion for the steady state problem. However, the MGRIT algorithm for solving our problem is very different to that of the spatial multigrid algorithm for the steady state problem. For example, MGRIT is based on reduction principles, applies multigrid methodology in one coordinate direction only, uses coarsening factors significantly larger than two, and uses a one-step discretization in one of the coordinate directions. Nonetheless, our analysis has shown that convergence of the two algorithms is, in large part, governed by closely related quantities. Therefore, it is reasonable to expect that deterioration in MGRIT convergence for (3.164) as $\beta \to 0^+$ is due to a poor coarse-grid correction of *space-time characteristic components*. That is, A_1 is a consistent space-time discretization of the PDE (3.164), and, thus, in the advection-dominated limit $\beta \to 0^+$, we expect that its space-time Fourier symbol will vanish along characteristic directions. More specifically, the Fourier symbol of A_1 can be expanded in Taylor series about that of the continuous operator:

$$\widetilde{A}_1(\omega, m\theta) = \widetilde{\mathcal{A}}(\omega, m\theta) + \text{higher order terms}, \tag{3.173}$$

$$= \mathrm{i}\left(\alpha\frac{\omega}{h} + \frac{\theta}{\delta t}\right) + \beta\frac{\omega^2}{h^2} + \text{higher order terms}. \tag{3.174}$$

For space-time characteristic components—modes with frequency $\theta \approx -\alpha\frac{\delta t}{h}\omega$—we thus have

$$\widetilde{A}_1(\omega, m\theta) \approx \beta\frac{\omega^2}{h^2} + \text{higher order terms}. \tag{3.175}$$

Therefore, as $\beta \to 0^+$, along characteristic directions the symbol $\widetilde{A}_1$ vanishes up to some mesh-dependent constant, analogously to the spatial case described above. We leave more rigorous study to future work, including whether we can determine asymptotically what the fraction in the spectral radius (3.169) limits to for some specific discretizations, as has been done for the spatial case described above. For now, however, we present in the following section numerical evidence to support our claim that overall MGRIT convergence for advection-dominated problems is hampered by its lack of convergence on characteristic components.

3.7.2 Numerical results

In this section, we consider plots of the spectral radius (3.169) discretely sampled over (ω, θ)-space for several discretizations of the constant-coefficient linear advection problem

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0, \quad \alpha > 0, \tag{3.176}$$

subject to periodic boundary conditions in space. Recall from Section 3.6.3 that the spectral radius derived via our LFA theory is inconsistent with the initial-value problem in the sense that it is not zero, despite MGRIT being nilpotent for the initial-value problem. However, as per our discussion in Section 3.6.3, the spectral radius does provide a reasonable measure of medium-term convergence for the initial-value problem before the effects of nilpotency begin to set in.

Recall from Section 2.3 that MGRIT convergence for SDIRK+U discretizations of (3.176) was poor when using rediscretization. Specifically, the solver converged very slowly for SDIRK1+U1, and diverged for all of the higher-order discretizations we considered. Here, we show in Figure 3.2 the spectral radius associated with two of these tests from Table 2.2. From the maximum spectral radius listed in the titles of the plots, we see that analysis

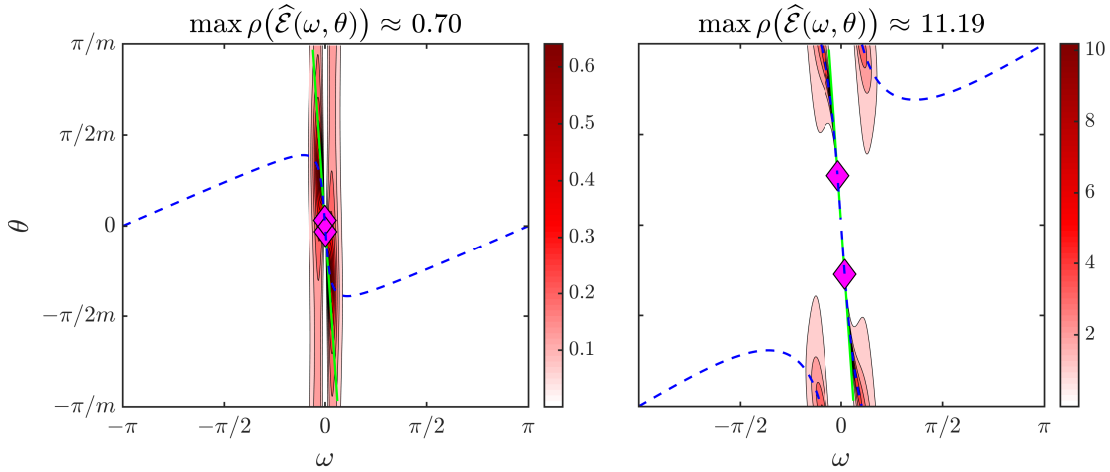

FIGURE 3.2: Spectral radius from (3.169) for the advection problem (3.176) shown over discretely sampled (ω, θ)-space using $2^{10} \times \frac{1}{m} 2^{10}$ equidistant points. The value of the function is indicated by the colour map. The maximum over the space is listed in the title, and the two locations at which this occurs are marked on the plot with magenta diamonds. Note that the spectral radius is non-smooth in certain regions where $|\mu(\omega)| \approx 1$, and, so, the contouring algorithm fails to show in which regions the spectral radius is the largest, particularly in the right panel. A dashed blue line represents for each ω, the value of θ at which the maximum spectral radius is obtained; that is, the curve $\omega(\theta^\dagger) = \frac{1}{m} \arg \mu(\omega)$ (see (3.140)). The solid green line represents the frequency relationship of characteristic components $\theta = -\alpha \delta t / h\omega$ for $|\theta| < \pi/m$. **Left:** SDIRK1+U1. **Right:** SDIRK2+U2. Both schemes use $m = 4$ and a fine-grid CFL number of $c = \alpha \delta t / h = 4$.

is consistent with the numerical results in that the maximum spectral radius is not much less than one for SDIRK1+U1, and it is much larger than one for SDIRK2+U2.

Observe that the specific modes that diverge the fastest—the magenta diamonds—are indeed characteristic components since the green lines represent (a subset of all) characteristic components. Moreover, for any fixed frequency $|\omega| \approx 0$, of all modes, it is the characteristic component that converges most slowly, since for $|\omega| \approx 0$ the green line overlays the dashed blue line, which represents the slowest converging mode $\theta^\dagger$ for a given ω. We also see that, for the most part, non-characteristic components converge quickly, in the sense that their spectral radii are much smaller than unity. This reinforces our theoretical arguments made in the previous section, and is consistent with what has been described in the spatial multigrid case regarding characteristic components [10, 108, 99].

We now consider a second class of discretization, namely those of Semi-Lagrangian type. We briefly introduced semi-Lagrangian discretizations in Chapter 1, and they were discussed in Remark 2.5, where we stated that they were closely related to the optimized coarse-grid operators that we presented in Chapter 2. Semi-Lagrangian discretizations are fundamentally different from method-of-lines discretizations, such as those of ERK+U and SDIRK+U that we considered in Chapter 2. Semi-Lagrangian methods are unconditionally stable with respect to the time-step size because they explicitly track characteristics. For this reason, they are appealing to us in the MGRIT context when considering time-only coarsening, as we do in this thesis.

Consider now the MGRIT solution of semi-Lagrangian discretizations of the advection problem (3.176). Note that we will consider semi-Lagrangian discretizations in much

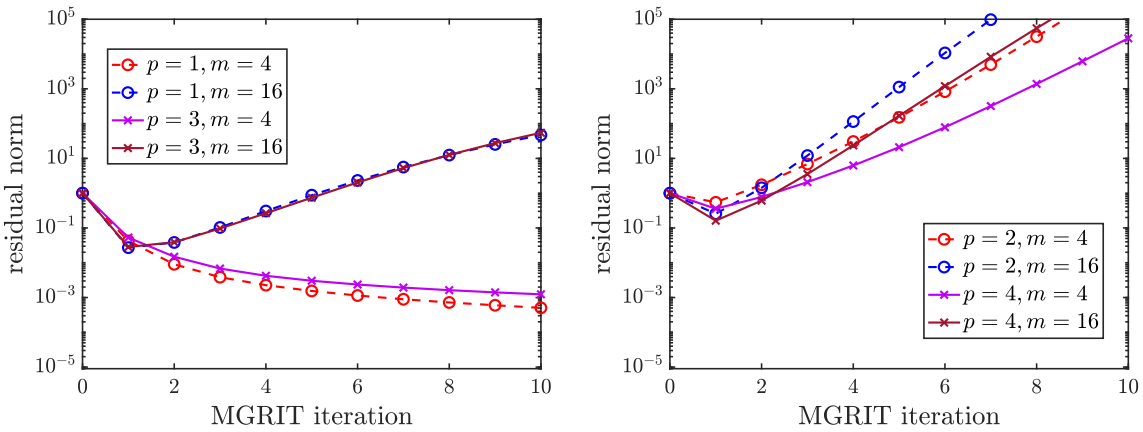

FIGURE 3.3: Two-norm of space-time residual (relative to its initial value) as a function of MGRIT iteration for solving advection problem (3.176) discretized with a pth-order semi-Lagrangian method. A two-level MGRIT solver with coarsening factor m is used, and the coarse-grid operator is given by rediscretization. The fine-grid CFL number is $c = \alpha\delta t/h = 0.85$, and the problem is discretized on an $n_x \times n_t = 2^8 \times 2^{11}$ space-time mesh. **Left:** Odd p. **Right:** Even p.

greater detail in Chapter 4, and, so, we leave further discussion about the mechanics of the discretization until then. The MGRIT residual histories for our tests are shown in Figure 3.3. Perhaps unsurprisingly given our arguments in this section and our findings in Chapter 2, MGRIT convergence is poor despite the coarse-grid semi-Lagrangian discretization being stable. This is yet another example that stable coarse-grid time integration is not sufficient for MGRIT to converge, let alone converge quickly.

We now test whether this poor MGRIT convergence for the semi-Lagrangian methods is the result of poor convergence of characteristic modes, as for the SDIRK+U schemes we considered previously. In Figure 3.4 we show analogous plots as in Figure 3.2, but now for the 1st-order semi-Lagrangian problem (left panel); we also contrast this with SDIRK1+U1 (right panel) using the same CFL as the semi-Lagrangian method, $c = 0.85$. Considering the spectral radius for the semi-Lagrangian method, the slowest converging modes are indeed characteristic components with frequency $|\omega| \approx 0$. However, in contrast to SDIRK1+U1, we see that there now exist other components—the additional green lines in the plot—with larger $|\omega|$ for which convergence is also slow. Note that the additional green lines in the left plot are also characteristic components, they have just simply had their temporal frequency θ periodically wrapped into the interval $[-\pi/m, \pi/m)$. That is, characteristic components have frequency $\theta = -\alpha\frac{\delta t}{h}\omega$, and, since $\omega \in [-\pi, \pi)$, it will certainly be the case that there are characteristic modes with $|\theta| > \pi/m$, despite the spectral radius (3.169) holding for $\theta \in \Theta^{\text{low}} = [-\pi/m, \pi/m)$. Since the spectral radius is periodic in θ with period $2\pi/m$ (this is easiest seen by looking at (3.171)), any components with frequency $|\theta| > \pi/m$ can be periodically wrapped into the interval $[-\pi/m, \pi/m)$. So, in contrast to SDIRK1+U1, for any fixed ω, the slowest converging mode is (approximately) the characteristic component, even for $|\omega| \approx \pi$.

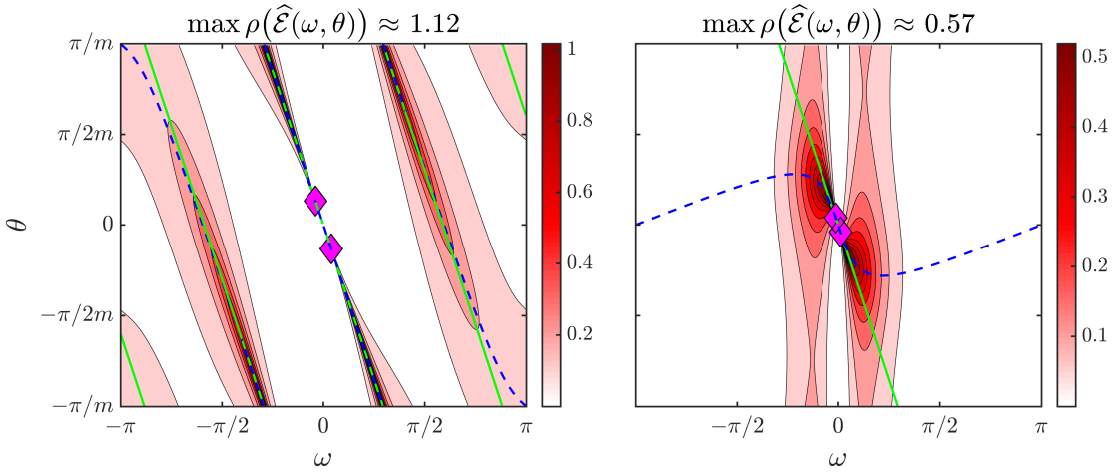

FIGURE 3.4: Analogous plots as in Figure 3.2, except (ω, θ)-space is sampled using $2^8 \times \frac{1}{m}2^{11}$ equidistant points, and the fine-grid CFL number is $c = 0.85$. **Left:** 1st-order semi-Lagrangian, $p = 1$, with $m = 4$. The additional green lines are the periodically wrapped characteristic components. **Right:** SDIRK1+U1 with $m = 4$.

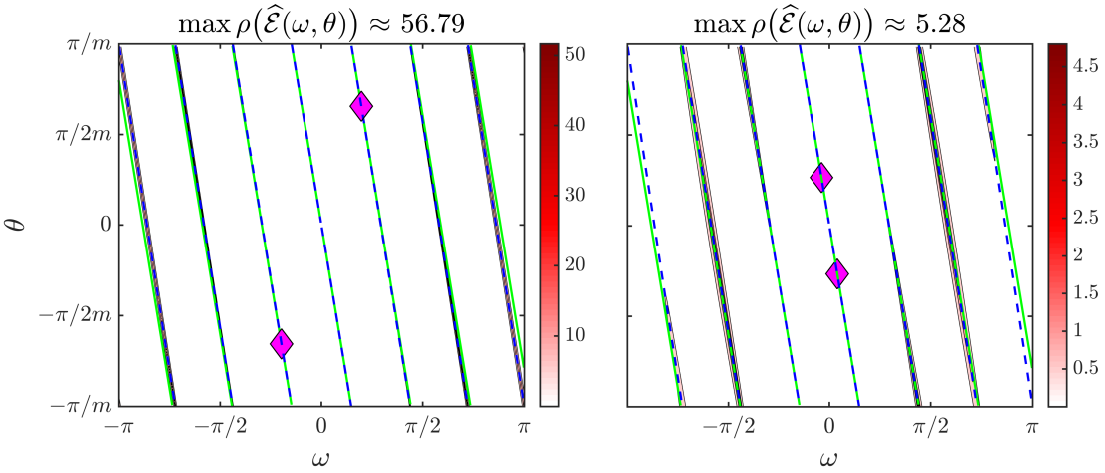

FIGURE 3.5: Identical plots as in Figure 3.4. **Left:** 2nd-order semi-Lagrangian, $p = 2$, with $m = 8$. **Right:** 3rd-order semi-Lagrangian $m = 8$.

Nonetheless, the semi-Lagrangian method still has in common with SDIRK1+U1 that most non-characteristic components are damped quickly by MGRIT. In Figure 3.5 we show analogous plots, except for 2nd- and 3rd-order semi-Lagrangian methods using a coarsening factor of $m = 8$. Again, we see the same trends as for the 1st-order semi-Lagrangian method in the left panel of Figure 3.2; however, in this case, the subset of modes for which convergence is slowest is much more tightly clustered around the characteristic components.

3.7.3 Discussion

Our numerical results support our theoretical-based arguments that poor MGRIT convergence for advection-dominated problems is due to poor convergence on characteristic components. Since poor convergence on characteristic components has been so widely studied in the spatial multigrid case, it is instructive to consider the solutions that have been proposed to remedy the issue, to see if they can be used to remedy the problem for MGRIT. While several potential fixes have been proposed in the spatial case, it should be noted that none of them appear to lead to multigrid convergence that is as fast and robust as it is for elliptic problems in general.

An idea often proposed for improving the robustness of spatial multigrid solvers on advection-dominated problems is to use so-called downstream relaxation [11, 108, 109, 99]. This essentially amounts to carrying out the relaxation in an order that propagates errors/residuals along characteristics. This is typically done in a global fashion by beginning at an inflow boundary and concluding at an outflow boundary; therefore, this represents an exact solver in the limit that only advection is present in the discrete problem. Furthermore, the process is inherently sequential. In the time-dependent context, downstream relaxation is

equivalent to time-stepping, which is exactly the sequential procedure that we are trying to move away from.

Another approach, which proved fairly successful in [108] and [3], is to use a Petrov–Galerkin coarse-grid operator, in which careful attention is paid to the properties of the intergrid transfer operators for smooth modes. Note that Galerkin coarse-grid operators are formed by the triple product of a restriction operator, fine-grid operator, and an interpolation operator, respectively. However, such an approach is again unfortunately not applicable in the MGRIT setting, recalling that reduction underpinnings of the algorithm mean that one has no control over restriction and interpolation (they are fixed as injection), and, while one has the freedom to choose the coarse-grid time-stepping operator as $\Psi \approx \Phi^m$, such an approximation does not allow for the use of Galerkin principles.

Thus, because we desire a highly parallel solver, and because the reduction nature of MGRIT fixes all but the coarse-grid time-stepping operator, there are fewer options available to us for addressing the poor coarse-grid correction of characteristic components. So, in this sense, the only feasible choice we have is to use a coarse-grid time-stepping operator that better approximates the action of the ideal coarse-grid time-stepping operator on characteristic components than basic rediscretization does. One possibility for achieving this is to design a coarse-grid operator whose truncation error matches, at least to lowest order, that of the ideal coarse-grid operator. In [108], this idea was proposed for the spatial multigrid case, where a coarse-grid operator whose truncation error better approximates that of the fine-grid operator was proposed (compare (3.172) to (3.169)). This idea forms the basis for the next chapter of the thesis.

3.8 Conclusions

In this chapter we have analysed the convergence behaviour of two-level MGRIT using the tool of LFA. Our analysis relied on assumptions of the time-stepping operators being time-independent and simultaneously diagonalizable with unitary eigenvectors, which are common assumptions among related analyses in the literature. The majority of the theoretical analysis was conducted in Section 3.4. In particular, we derived closed-form expressions for the predictions offered by LFA, meaning that we developed simple and easy to interpret formulae for quantities such as the norm and spectral radius of the error propagator.

Our LFA theory is not rigorous for initial-value problems due to it neglecting boundary effects; however, it does apply rigorously to a class of time-periodic MGRIT solvers, as described in Section 3.5. Moreover, by comparison with existing literature, we showed

that the approximations offered by our theory are accurate for initial-value problems, and we verified that they are exact in the asymptotic limit $n_t \to \infty$ (see Section 3.6.2). Due to our closed-form derivations, our theory offers significantly more insight compared to existing LFA theories for MGRIT, which have been based on numerical computation, as described in Section 3.6.1.

In Section 3.7, we applied our LFA theory to describe the poor convergence of MGRIT for advection-dominated problems. We showed that these convergence issues are closely related to the well-known convergence issues that plague spatial multigrid solvers when applied to steady state advection-dominated problems. In particular, we showed that the slowest converging Fourier modes are the so-called characteristic components, which are the modes that oscillate in the direction orthogonal to characteristics. The insight gained for the convergence of advection-dominated problems in this chapter is significant, as is the direct link to well-documented issues for spatial multigrid solvers. This knowledge will undoubtedly be instrumental in developing improved parallel-in-time solvers for advection-dominated problems.

Chapter 4

Fast MGRIT for advection via dissipatively corrected coarse-grid operators

4.1 Previous work and outline

In Chapter 2 we developed fast solvers for advection problems, however, our approach for doing so was limited only to constant-coefficient problems. Nonetheless, we developed important heuristics for coarse-grid operators, including that they should track characteristics if they are to be consistent with the underlying hyperbolic PDE. A natural choice of coarse-grid operator is therefore a semi-Lagrangian discretization, since, by design, it tracks characteristics, independent of the size of the time step. However, in Section 3.7 we showed that coarse-grid semi-Lagrangian operators are ineffective in general, since they provide an insufficient coarse-grid correction to characteristic components. To overcome this, here we develop a semi-Lagrangian-like coarse-grid operator whose truncation error better approximates that of the ideal coarse-grid operator than a standalone semi-Lagrangian operator's does. This idea is based loosely on [108, Sec. 5.1], in which a coarse-grid operator was developed to address related issues that occur for the spatial multigrid solution of steady state advection-dominated PDEs.

The remainder of this chapter is organized as follows. Section 4.2 develops the coarse-grid operator for the constant-wave-speed (or constant-coefficient) advection problem. Section 4.3 extends the coarse-grid operator to variable-wave-speed problems. Section 4.4 extends the coarse-grid operator to the multilevel setting, and shows how it may be used to solve advection-diffusion problems. Section 4.5 generalizes the coarse-grid operator to problems in two spatial dimensions. Finally, concluding remarks are given in Section 4.6.

4.2 Constant-wave-speed advection

We begin this chapter by considering the constant-wave-speed advection problem

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0, \quad (x,t) \in \Omega \times (0,T], \quad u(x,0) = u_0(x), \quad \alpha > 0 \text{ a constant}, \tag{4.1}$$

for spatial domain $\Omega \subset \mathbb{R}$, and solution u subject to periodic boundary conditions on $\partial\Omega$. Our goal is the efficient parallel-in-time solution of semi-Lagrangian discretizations of this problem.

In Section 3.7 we saw that simply employing a semi-Lagrangian operator on the coarse grid (i.e., rediscretizing) generally leads to a divergent MGRIT solver for this problem, despite the operator being stable. To help recall the behaviour of the solver, some of these tests have been reproduced in Figure 4.1. Convergence of MGRIT is poor in all of the cases shown; the residual increases in all cases except when $m = 4$ and $p \in \{1, 3\}$, and even then, it stagnates after the first few iterations. In all other cases, the residual increases as the solver iterates. By Definition 1.1, MGRIT diverges in all of these tests. Note for the MGRIT solves shown in Figure 4.1, the initial guess at the space-time solution is taken to be uniformly random, as it is for all MGRIT runs throughout this thesis.

4.2.1 Semi-Lagrangian discretization

The coarse-grid operators we will propose as an alternative to rediscretization are intimately linked to the structure of the semi-Lagrangian discretizations we consider. So,

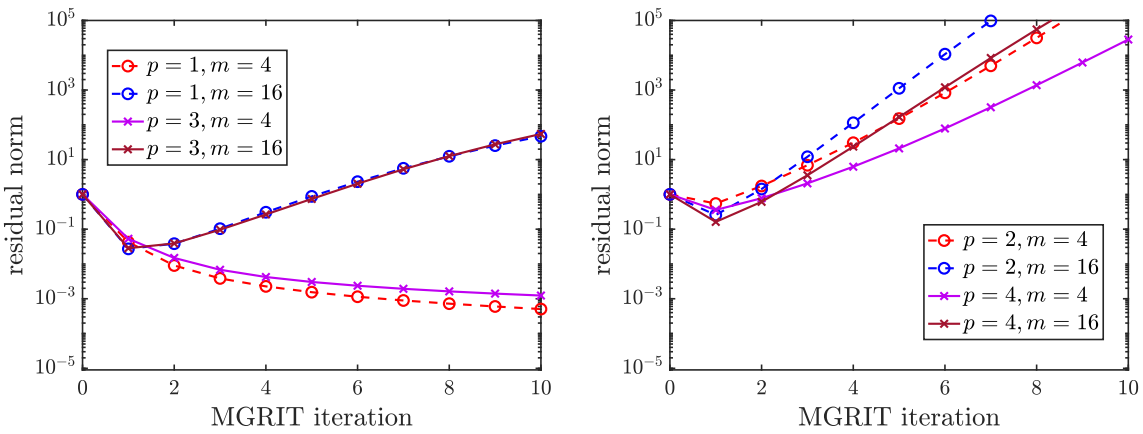

FIGURE 4.1: Two-norm of space-time residual (relative to its initial value) as a function of MGRIT iteration for solving $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$, $x \in (-1, 1)$ discretized with a pth-order semi-Lagrangian method. A two-level MGRIT solver with coarsening factor m is used. The fine-grid CFL number is $\delta t/h = 0.85$, and the problem is discretized on an $n_x \times n_t = 256 \times 2048$ space-time mesh. **Left:** Odd p. **Right:** Even p.

while we have previously used semi-Lagrangian discretization in Chapter 3, we now provide a more detailed description of how they work. The semi-Lagrangian discretization of (4.1) is based on its Lagrangian formulation, which reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = \alpha. \tag{4.2}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t),t) = 0, \quad (\xi(t),t) \in \Omega \times (0,T], \quad u(x,0) = u_0(x). \tag{4.3}$$

Equation (4.2) is the trajectory equation. Its solution describes the characteristic curves $(x,t) = (\xi(t),t)$ of the PDE. Since α is constant, the characteristics are straight lines. Equation (4.3) is the evolution equation, and it describes how the PDE solution changes along a given characteristic curve. In this case, it states that the solution is constant along characteristics.

Semi-Lagrangian discretizations use a grid- or mesh-based discretization of (4.2) and (4.3), as opposed to pure Lagrangian methods, which use particles. Since the problem is linear (i.e., the characteristics are not coupled to the PDE solution), the coupled equations may be solved by first solving (4.2) followed by that of (4.3). To this end, we discretize the spatial domain $\Omega \subset \mathbb{R}$ with a set of n_x nodes $\boldsymbol{x} = (x_1, x_2, \ldots, x_{n_x})^\top$ that are equally separated by a distance h, $x_{i+1} = x_i + h$. We discretize the time interval $t \in [0,T]$ with an equidistant mesh of points, $0 = t_0 < t_1 < \cdots < t_{n_t} = T$ with $t_{n+1} = t_n + \delta t$. Given the vector $\boldsymbol{u}_n \approx u(\boldsymbol{x}, t_n)$, which represents the approximate solution of (4.1) at time t_n at all the spatial mesh points $\boldsymbol{x}$, the semi-Lagrangian method will advance this to a new approximation $\boldsymbol{u}_{n+1}$ as we now describe.

On some characteristic $\xi_i(t)$, the evolution equation (4.3) states that the PDE solution at $(x,t) = (\xi_i(t_{n+1}), t_{n+1})$ is equal to the solution at $(x,t) = (\xi_i(t_n), t_n)$. Since we desire the solution at the mesh point $(x,t) = (x_i, t_{n+1})$, if we force the characteristic $\xi_i(t)$ to pass through this point, then the solution there is simply the solution at the foot of the characteristic (see Figure 4.2 for an example). To this end, define the local characteristic $\xi_i^{(t_n,\delta t)}(t)$ to be that which passes through the *arrival point* $(x,t) = (x_i, t_n + \delta t)$. Then, we locate the associated *departure point* $(x,t) = \left(\xi_i^{(t_n,\delta t)}(t_n), t_n\right)$ by solving the final-value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_i^{(t_n,\delta t)}(t) = \alpha, \quad t \in [t_n, t_n + \delta t), \quad \xi_i^{(t_n,\delta t)}(t_n + \delta t) = x_i. \tag{4.4}$$

Since α is constant, we can exactly calculate the departure point to be

$$\xi_i^{(t_n,\delta t)}(t_n) = x_i - \alpha\delta t. \tag{4.5}$$

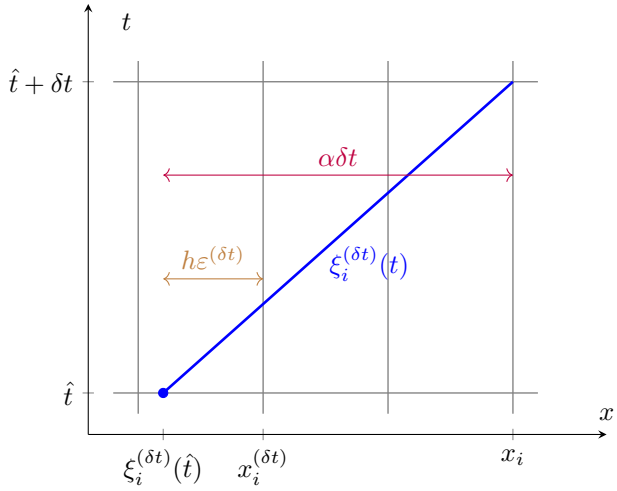

FIGURE 4.2: The characteristic $\xi_i^{(\delta t)}(t)$ for $t \in [\hat{t}, \hat{t}+\delta t]$ for the constant-wave-speed problem (4.1). The characteristic passes through the arrival point $(x,t) = (x_i, \hat{t}+\delta t)$. The departure point is the location on the x-axis of the characteristic at $t = \hat{t}$. The departure point is decomposed into the sum of its east-neighbouring mesh point and its distance from this point, as in (4.6).

For this special case of constant α, the departure point (4.5) clearly does not depend on t_n, and the distance the characteristic travels in the x-direction (i.e., $x_i - \xi_i^{(t_n,\delta t)}(t_n)$) is independent of the spatial location i. Therefore, we first simplify our notation and note that the more general notation of (4.4) will reappear later when considering variable-wave-speed problems.

Let $\xi_i^{(\delta t)}(t)$ be a local characteristic that passes though an arrival point $(x,t) = (x_i, \hat{t}+\delta t)$, where $\hat{t}$ is some time on the mesh. Then, $\xi_i^{(\delta t)}(\hat{t})$ is the departure point of this characteristic and is given by

$$\xi_i^{(\delta t)}(\hat{t}) = x_i - \alpha\delta t \equiv x_i^{(\delta t)} - h\varepsilon^{(\delta t)}, \quad \varepsilon^{(\delta t)} \in [0,1). \tag{4.6}$$

In the second expression, the departure point is decomposed into the sum of the mesh point to its east, $x_i^{(\delta t)}$, and its mesh-normalized distance from this point, $\varepsilon^{(\delta t)}$. See Figure 4.2 for a schematic example.

Upon locating the departure point (4.6), we are faced with the problem that it does not, in general, coincide with a mesh point, yet the PDE approximation $\boldsymbol{u}_n$ is only available at mesh points. To resolve this problem, a degree (at most) p interpolating polynomial is fit through the entries of $\boldsymbol{u}_n$ at the contiguous $p+1$ mesh nodes $\{x_i^{(\delta t)} + hj\}_{j=-\ell(p)}^{r(p)}$.[1] The left and right extents of the stencil, $\ell(p)$ and $r(p)$, respectively, are chosen so that the

[1]For interpolation polynomials of degree $p \geq 1$, both the west and east neighbouring mesh points appear in the interpolation stencil of the departure point. Therefore, the choice made in (4.6) to write the departure point in terms of its east neighbour $x_i^{(\delta t)}$, and hence the nodes $x_i^{(\delta t)} + hj$ in the interpolation stencil, is arbitrary in the sense that it could also have been written in terms of its west neighbour (or any other mesh point that is in the interpolation stencil, for that matter).

set of interpolation nodes represent the $p+1$ nearest neighbours of the departure point. When p is odd, which is what we primarily focus on in this chapter, the stencil uses a symmetric number of nodes on either side of the departure point. Equivalently, the stencil includes one additional node to the left of $x_i^{(\delta t)}$ (i.e., the root node of the stencil) than to its right, and, thus, $\ell(p) = \frac{p+1}{2}$ and $r(p) = \frac{p-1}{2} = \ell(p) - 1$. When p is even, the stencil has a one-point bias, such that $\ell(p)$ and $r(p)$ depend on whether $\varepsilon^{(\delta t)}$ is larger than one half, but for simplicity we ignore this dependence in our notation.

Locating the departure points for all arrival points $\boldsymbol{x}$ at time $t = t_n + \delta t$ and then carrying out this piecewise polynomial interpolation constitutes a single time-step of the semi-Lagrangian discretization. We denote the time-stepping operator for this discretization by $\mathcal{S}_p^{(\delta t)} \in \mathbb{R}^{n_x \times n_x}$. Note that in this special case of a constant wave-speed and periodic boundaries in space, $\mathcal{S}_p^{(\delta t)}$ is a circulant matrix.

4.2.2 The coarse-grid operators

From the results in Section 3.7, and those repeated in Figure 4.1, it is clear that, in general, coarse-grid semi-Lagrangian operators do not adequately approximate the ideal coarse-grid operator. Despite this, the idea of applying a coarse-grid semi-Lagrangian discretization remains appealing because the discretization is stable for all time-step sizes, unlike its explicit Eulerian counterparts. For this reason, we seek a coarse-grid operator that is based on a semi-Lagrangian discretization, but that provides a better approximation to the ideal coarse-grid operator. There are many different metrics one could use to characterize the difference between a coarse-grid operator and the ideal coarse-grid operator. Here we use the concept of local truncation error, which we define as the amount by which the exact PDE solution fails to satisfy the discrete scheme after one time step. We now present our first result, which is on the truncation error of the semi-Lagrangian method; note that error estimates closely related to the following exist elsewhere in the literature (see, e.g., [31, p. 170]).

Lemma 4.1 (Constant-wave-speed semi-Lagrangian truncation error)**.** *Suppose that at any time t, the solution $u(x,t)$ of (4.1) is at least $p+1$ times continuously differentiable with respect to x. Then, the local truncation error of the semi-Lagrangian scheme $\mathcal{S}_p^{(\delta t)}$ can be expressed by*

$$u(\boldsymbol{x}, t_{n+1}) - \mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_n) = (-h)^{p+1} f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+1}) + \mathcal{O}(h^{p+2}), \tag{4.7}$$

where $u(\boldsymbol{x}, t) \in \mathbb{R}^{n_x}$ denotes the vector composed of the PDE solution sampled at the mesh points $\boldsymbol{x}$ at time t. In (4.7), f_{p+1} is the degree $p+1$ polynomial defined by

$$f_{p+1}(z) := \frac{1}{(p+1)!} \prod_{q=-\ell(p)}^{r(p)} (q+z). \tag{4.8}$$

Furthermore, the associated ideal coarse-grid operator defined by stepping m times with $\mathcal{S}_p^{(\delta t)}$ across the interval $t \in [t_n, t_n + m\delta t]$ has a local truncation error expressible as

$$u(\boldsymbol{x}, t_{n+m}) - \left[\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}\right] u(\boldsymbol{x}, t_n) = (-h)^{p+1} m f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+m}) + \mathcal{O}(h^{p+2}). \tag{4.9}$$

Proof. Since $\mathcal{S}_p^{(\delta t)}$ exactly locates departure points of (4.1), the only truncation error resulting from applying the ith row of $\mathcal{S}_p^{(\delta t)}$ to $u(\boldsymbol{x}, t)$ is the error from the polynomial interpolation at the departure point $\xi_i^{(\delta t)}(t_n) = x_i^{(\delta t)} - h\varepsilon^{(\delta t)}$. Since $u(x, t)$ is assumed $p+1$ times continuously differentiable with respect to x, the standard error estimate from polynomial interpolation theory can be applied (see, e.g., [24, Th. 3.1.1]). Applying the fact that the interpolation nodes are equally separated by distance h, this results in an interpolation error at the departure point of

$$\begin{aligned} & u\big(\xi_i^{(\delta t)}(t_n), t_n\big) - \Big(\mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_n)\Big)_i \\ &= \frac{1}{(p+1)!} \prod_{q=-\ell(p)}^{r(p)} \Big[\big(x_i^{(\delta t)} - h\varepsilon^{(\delta t)}\big) - \big(x_i^{(\delta t)} + hq\big)\Big] \left.\frac{\partial^{p+1} u}{\partial x^{p+1}}\right|_{(\zeta_i, t_n)}, \qquad (4.10) \\ &= (-h)^{p+1} f_{p+1}\big(\varepsilon^{(\delta t)}\big) \left.\frac{\partial^{p+1} u}{\partial x^{p+1}}\right|_{(\zeta_i, t_n)}, \qquad (4.11) \end{aligned}$$

for some unknown point $\zeta_i \in \big(x_i^{(\delta t)} - h\ell(p), x_i^{(\delta t)} + hr(p)\big)$. Since ζ_i and $\xi_i^{(\delta t)}(t_n)$ are a distance of $\mathcal{O}(h)$ apart (they are both in the interval containing all the interpolation nodes), write $\zeta_i = \xi_i^{(\delta t)}(t_n) + h\hat{\zeta}_i$ for some other unknown point $\hat{\zeta}_i$. Then, by Taylor expansion, the derivative in (4.11) can be written as

$$\left.\frac{\partial^{p+1} u}{\partial x^{p+1}}\right|_{(\zeta_i, t_n)} = \left.\frac{\partial^{p+1} u}{\partial x^{p+1}}\right|_{(\xi_i^{(\delta t)}(t_n) + h\hat{\zeta}_i, t_n)} = \left.\frac{\partial^{p+1} u}{\partial x^{p+1}}\right|_{(\xi_i^{(\delta t)}(t_n), t_n)} + \mathcal{O}(h). \tag{4.12}$$

The truncation error result (4.7) follows from substituting (4.12) into (4.11), and then applying that the solution u of (4.1) at any arrival point is equal to the solution at the associated departure point, $u(x_i, t_{n+1}) = u\big(\xi_i^{(\delta t)}(t_n), t_n\big)$.

Now consider the truncation error (4.9) for the ideal coarse-grid operator. Applying $\mathcal{S}_p^{(\delta t)}$ to both sides of (4.7) gives

$$\mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_{n+1}) - \mathcal{S}_p^{(\delta t)} \mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_n) = (-h)^{p+1} f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} \mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_{n+1}) + \mathcal{O}(h^{p+2}). \tag{4.13}$$

However, from the truncation error of $\mathcal{S}_p^{(\delta t)}$ given by (4.7) we have

$$\mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_{n+1}) = u(\boldsymbol{x}, t_{n+2}) - (-h)^{p+1} f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+2}) + \mathcal{O}(h^{p+2}). \tag{4.14}$$

Substituting this result into both sides of (4.13), keeping only terms up to size $\mathcal{O}(h^{p+1})$, and rearranging gives

$$u(\boldsymbol{x}, t_{n+2}) - \mathcal{S}_p^{(\delta t)} \mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_n) = 2(-h)^{p+1} f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+2}) + \mathcal{O}(h^{p+2}). \tag{4.15}$$

By an inductive argument, repeating these steps a further $m-1$ times gives (4.9). ■

Having now developed an asymptotic expansion for the ideal coarse-grid operator, we relate this to the coarse-grid semi-Lagrangian operator in the following lemma.

Lemma 4.2 (Perturbed coarse-grid semi-Lagrangian operators)**.** *Let $\mathcal{S}_p^{(m\delta t)}$ be a coarse-grid semi-Lagrangian discretization of (4.1). Then, this operator can be expressed as an $\mathcal{O}(h^{p+1})$ perturbation of the ideal coarse-grid operator $\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}$ in the following three ways:*

$$\begin{aligned}
&\left[\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}\right] u(\boldsymbol{x}, t_n) \\
&\quad = \mathcal{S}_p^{(m\delta t)} u(\boldsymbol{x}, t_n) + \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+m}) + \mathcal{O}(h^{p+2}), && (4.16)\\
&\quad = \left(1 + \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1}}{\partial x^{p+1}}\right) \mathcal{S}_p^{(m\delta t)} u(\boldsymbol{x}, t_n) + \mathcal{O}(h^{p+2}), && (4.17)\\
&\quad = \left(1 - \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1}}{\partial x^{p+1}}\right)^{-1} \mathcal{S}_p^{(m\delta t)} u(\boldsymbol{x}, t_n) + \mathcal{O}(h^{p+2}), && (4.18)
\end{aligned}$$

where we have defined the constant

$$\phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) := (-1)^{p+1} \left(f_{p+1}\big(\varepsilon^{(m\delta t)}\big) - m f_{p+1}\big(\varepsilon^{(\delta t)}\big)\right). \tag{4.19}$$

The operator defined by the inverse in (4.18) is to be interpreted in the geometric series sense.

Proof. From the truncation error of $\mathcal{S}_p^{(\delta t)}$ in (4.7), it can immediately be seen that the truncation of $\mathcal{S}_p^{(m\delta t)}$ is

$$u(\boldsymbol{x}, t_{n+m}) - \mathcal{S}_p^{(m\delta t)} u(\boldsymbol{x}, t_n) = (-h)^{p+1} f_{p+1}\big(\varepsilon^{(m\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+m}) + \mathcal{O}(h^{p+2}). \quad (4.20)$$

The first equality (4.16) follows by subtracting the truncation error of the coarse-grid operator (4.20) from (4.9) and rearranging the resulting equation. The second equality (4.17) follows by substituting $u(\boldsymbol{x}, t_{n+m}) = \mathcal{S}_p^{(m\delta t)} u(\boldsymbol{x}, t_n) + \mathcal{O}(h^{p+1})$, as is given by (4.20), into (4.16). Finally, the third equality (4.18) follows from the fact that the coefficient in (4.17) is equal to the geometric expansion of the coefficient in (4.18) up to terms of $\mathcal{O}\big(h^{2(p+1)}\big)$. ■

The significance of Lemma 4.2 is that we now have several asymptotic relationships between the ideal coarse-grid operator, and the coarse-grid semi-Lagrangian operator. Specifically, using (4.17) and (4.18) we can devise coarse-grid operators that serve as $\mathcal{O}(h^{p+2})$ approximations to the ideal coarse-grid operator, which is an improvement over the $\mathcal{O}(h^{p+1})$ approximation offered by the coarse-grid semi-Lagrangian operator $\mathcal{S}_p^{(m\delta t)}$. Based on the relationship (4.17), we propose the following *explicit* coarse-grid operator with matrix $\mathcal{D}_{p+1} \in \mathbb{R}^{n_x \times n_x}$ to be defined later

$$\Phi^{(m\delta t)} = \mathcal{F}_{p+1} \mathcal{S}_p^{(m\delta t)}, \quad \text{where } \mathcal{F}_{p+1} := I + \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) \mathcal{D}_{p+1}. \quad (4.21)$$

Furthermore, based on relationship (4.18) we propose the following *implicit-explicit* coarse-grid operator

$$\Phi^{(m\delta t)} = \mathcal{B}_{p+1} \mathcal{S}_p^{(m\delta t)}, \quad \text{where } \mathcal{B}_{p+1} := \Big[I - \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) \mathcal{D}_{p+1}\Big]^{-1}. \quad (4.22)$$

The $\mathcal{F}_{p+1}$ and $\mathcal{B}_{p+1}$ notation is used to represent *forward* and *backward* Euler steps, respectively, for which the reasoning will become clear in Section 4.2.3. Moving forward, we will typically refer to (4.21) and (4.22) as the 'forward Euler' and 'backward Euler' coarse-grid operators, respectively. In (4.21) and (4.22), the matrix $\mathcal{D}_{p+1} \in \mathbb{R}^{n_x \times n_x}$ is an approximation of the $p+1$st-degree differential operator $h^{p+1} \operatorname{diag}\big(\frac{\partial^{p+1}}{\partial x^{p+1}}\big)$. In all of our numerical and theoretical results, $\mathcal{D}_{p+1}$ will be a periodic, finite-difference approximation. So long as $\mathcal{D}_{p+1}$ is a consistent discretization, its order of accuracy does not matter in the sense that the truncation errors of (4.21) and (4.22) will be equal to those in (4.17) and (4.18), respectively, up to terms of $\mathcal{O}(h^{p+2})$. While we will not specify the order of approximation provided by $\mathcal{D}_{p+1}$ in much of our theoretical analysis, in our numerical experiments (which use odd p, as discussed below) we will always take it to be a 2nd-order accurate discretization since this leads to the best conditioned operator (further details are

given in Section 4.2.4). Finally, note that since entries in finite-difference approximations of diag $\left(\frac{\partial^{p+1}}{\partial x^{p+1}}\right)$ are proportional to $h^{-(p+1)}$, the entries in $\mathcal{D}_{p+1}$ are independent of h.

We now present some preliminary numerical results to help motivate the direction of the remainder of this chapter. For the odd polynomial degrees $p \in \{1, 3\}$, Figure 4.3 shows MGRIT convergence plots for several coarsening factors m for the forward Euler and backward Euler coarse-grid operators (4.21) and (4.22), respectively. Notice that in almost all cases, convergence is very fast, standing in stark contrast to the left panel of Figure 4.1, where simply using $\mathcal{S}_p^{(m\delta t)}$ led to MGRIT diverging. Notice that both the $m = 64$ residual curves in the left panel have blown up initially, and are only converging as the iteration number approaches $\sim n_t/(2m)$. That is, MGRIT has diverged for these two problems. However, the $m = 64$ solves for the backward Euler operator (right panel) have converged very quickly.

We do not present here any numerical results for the case of even polynomial degrees p. In many of our numerical tests, we have found that MGRIT diverges for even p when using either coarse-grid operator (4.21) or (4.22). Currently, we do not have a proper understanding of why the coarse-grid operators seem ineffective for even p. Note the leading-order truncation error of the ideal coarse-grid operator is dispersive when p is even versus dissipative when p is odd (see Lemma 4.1). As such, we suspect the worse solver convergence for even p could be due to MGRIT being able to correct dissipative errors more easily than dispersive errors (see the analysis of [80]). We also remark that for even p the correction in some sense destabilises the coarse-grid semi-Lagrangian scheme by making it more dispersive (contrary to dissipation, increased dispersion tends to be

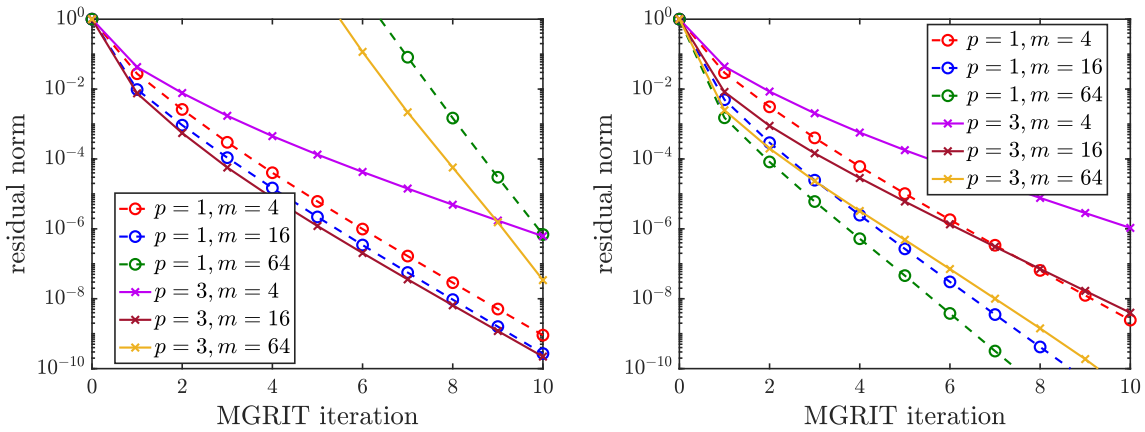

FIGURE 4.3: Two-norm of space-time residual (relative to its initial value) as a function of MGRIT iteration for solving $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$, $x \in (-1, 1)$ discretized with the semi-Lagrangian method $\mathcal{S}_p^{(\delta t)}$. A two-level MGRIT method is used with coarsening factor m. The fine-grid CFL number is $\delta t/h = 0.85$, and the problem is discretized on an $n_x \times n_t = 256 \times 2048$ space-time mesh. **Left:** Coarse-grid operator is the explicit, Forward Euler operator (4.21). **Right:** Coarse-grid operator is implicit-explicit, Backward Euler operator (4.22).

a destabilising phenomenon). Nonetheless, since we are unable to get a robust MGRIT solver for even polynomial degrees p, throughout the rest of the chapter we consider only odd p. Developing a better understanding of the inadequacies of (4.21) and (4.22) for dispersive semi-Lagrangian discretizations remains a topic for future research.

While the proposed backward Euler coarse-grid operator (4.22) appears to yield fast MGRIT convergence in our initial tests (for odd p), one over-arching concern is that of its cost. Specifically, the fine-grid operator is explicit, requiring no linear solves, yet applying the coarse-grid operator (4.22) requires a linear solve. However, we will show later in the chapter that this linear solve can be carried out approximately, with very low cost.

Remark 4.3 (An alternative coarse-grid operator)**.** *The above strategy for approximating the truncation error of the ideal coarse-grid is not the only one possible. We have also developed another coarse-grid operator which is more closely based on the coarse-grid operator proposed in [108, Sec. 5.1] for the spatial multigrid solution of steady state advection-dominated problems. Our alternative coarse-grid operator is a linear combination of order p and order $q > p$ semi-Lagrangian discretizations. The linear weights are chosen so that the resulting operator has a truncation error that matches to lowest order the ideal coarse-grid operator, and thus better approximates it than the standalone order p operator does. While this operator leads to fast MGRIT convergence in certain situations, it has two fundamental flaws which mean it cannot be used in practice. Firstly, it has poor stability properties in the sense that it becomes unstable for sufficiently large coarsening factors m. Secondly, it fails to provide an adequate approximation in the event that fine-grid characteristics are not mesh aligned, but coarse-grid characteristics are. For completeness, a more detailed discussion of this operator is provided in Appendix B.1.*

4.2.3 Interpretations of the proposed coarse-grid operators

We now provide interpretations of the proposed coarse-grid operators (4.21) and (4.22). These interpretations do not necessarily generalize to the variable-coefficient problems considered later in the chapter, but nonetheless they provide an interesting point of view, and potentially offer insight for designing coarse-grid operators for more difficult problems.

4.2.3.1 Interpretation one: Solving an augmented coarse-grid equation

The forward and backward Euler-based coarse-grid operators (4.21) and (4.22) correspond to particular coarse-grid discretizations of the PDE

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = \frac{1}{m\delta t}\phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1} u}{\partial x^{p+1}}. \tag{4.23}$$

Specifically, they use a mixed discretization, in which the coarse-grid semi-Lagrangian method $\mathcal{S}_p^{(m\delta t)}$ deals with the advection term, and then the method of lines is applied to solve the rest of the equation. In doing so, the right-hand side of (4.23) discretized in space using the matrix $\mathcal{D}_{p+1}$, and the time derivative on the left-hand side is discretized using forward and backward Euler steps in (4.21) and (4.22), respectively.

To understand why this is the case, consider the Lagrangian formulation of (4.23),

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = \alpha, \tag{4.24}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t), t) = \frac{1}{m\delta t}\phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1}}{\partial x^{p+1}} u(\xi(t), t). \tag{4.25}$$

Now, suppose the evolution equation (4.25) holds on the characteristic $\xi_i^{(m\delta t)}(t)$, and is discretized on this characteristic over $t \in [t_n, t_{n+m}]$ using a single forward Euler step resulting in the approximation $v(x,t) \approx u(x,t)$, with v satisfying

$$\frac{v\big(\xi_i^{(m\delta t)}(t_{n+m}), t_{n+m}\big) - v\big(\xi_i^{(m\delta t)}(t_n), t_n\big)}{m\delta t} = \frac{1}{m\delta t}\phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1}}{\partial x^{p+1}} v\big(\xi_i^{(m\delta t)}(t_n), t_n\big). \tag{4.26}$$

Rearranging and substituting the value of the characteristic at the arrival point gives

$$v(x_i, t_{n+m}) = \left(1 + \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) h^{p+1} \frac{\partial^{p+1}}{\partial x^{p+1}}\right) v\big(\xi_i^{(m\delta t)}(t_n), t_n\big). \tag{4.27}$$

Now, further suppose that the derivative on the right-hand side at the departure point is estimated using the finite-difference rule of $\mathcal{D}_{p+1}$ applied to the values of v at neighbouring departure points,

$$\frac{\partial^{p+1}}{\partial x^{p+1}} v\big(\xi_i^{(m\delta t)}(t_n), t_n\big) \approx h^{-(p+1)} \sum_{j \in \mathcal{J}} \big(\mathcal{D}_{p+1}\big)_{ij} v\big(\xi_{i+j}^{(m\delta t)}(t_n), t_n\big), \tag{4.28}$$

in which $\mathcal{J}$ is the index set of the non-zero weights in the finite-difference rule of $\mathcal{D}_{p+1}$. Finally, suppose that in the resulting formula values of v at departure points are interpolated using degree (at most) p polynomials. Using the fact that $\mathcal{D}_{p+1}$ and $\mathcal{S}_p^{(m\delta t)}$ commute

(they are both circulant), we arrive at the fully discrete update

$$\boldsymbol{u}_{n+m} = \mathcal{S}_p^{(m\delta t)}\boldsymbol{u}_n + \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big)\mathcal{D}_{p+1}\mathcal{S}_p^{(m\delta t)}\boldsymbol{u}_n = \mathcal{F}_{p+1}\mathcal{S}_p^{(m\delta t)}\boldsymbol{u}_n, \tag{4.29}$$

where $\boldsymbol{u}_n = (v(x_1, t_n), \ldots, v(x_{n_x}, t_n))^\top$. This is precisely the forward Euler coarse-grid operator defined in (4.21). Carrying out an analogous set of approximations but starting with a backward Euler step instead of the forward Euler step in (4.26) results in the update given in (4.22).

Formulating the coarse-grid operators as discretizations of the PDE (4.23) allows us to interpret them in the following way. The coarse-grid operators first treat the advective component of (4.23) with $\mathcal{S}_p^{(m\delta t)}$, that is, they interpolate the solution from the mesh to departure points. Then they apply an Euler step to approximately evolve the interpolated solution along the characteristic according to (4.25), which has the effect of post-processing (or correcting) the solution by removing the leading-order truncation error introduced from $\mathcal{S}_p^{(m\delta t)}$ and replacing it with the leading-order truncation error of the ideal coarse-grid operator $\prod_{k=0}^{m-1}\mathcal{S}_p^{(\delta t)}$.

4.2.3.2 Interpretation two: A dissipative correction to rediscretization

For the case of odd polynomial degrees p, we now consider a second interpretation of the proposed coarse-grid operators. Since both the ideal coarse-grid operator and the

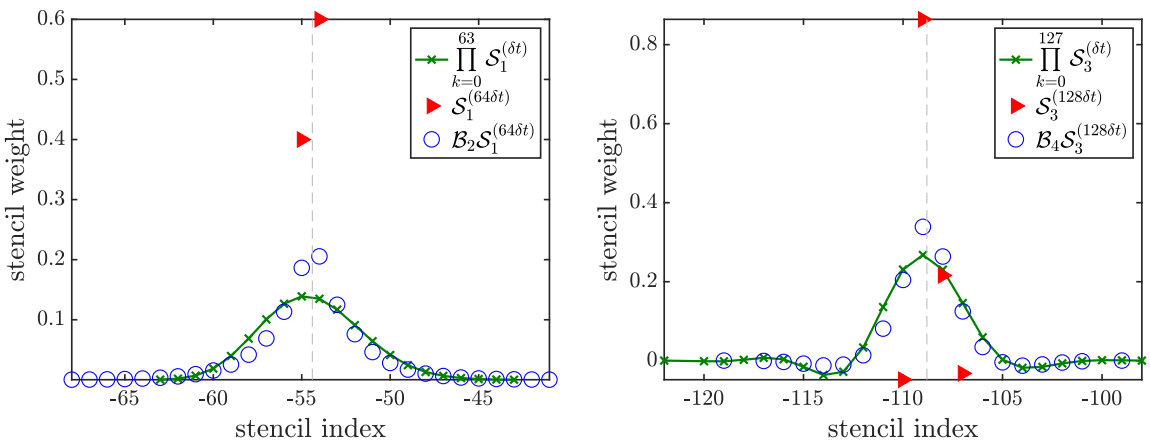

FIGURE 4.4: Stencil weights as a function of their index for three coarse-grid operators associated with $p = 1$ semi-Lagrangian using $m = 64$ (**left**), and $p = 3$ semi-Lagrangian using $m = 128$ (**right**) of (4.1). In both cases, the fine-grid CFL number is 0.85, $\delta t = 0.85 \times h$. Weights are shown for ideal coarse-grid operators $\prod_{k=0}^{m-1}\mathcal{S}_p^{(\delta t)}$, coarse-grid semi-Lagrangian operators $\mathcal{S}_p^{(m\delta t)}$, and the proposed backward Euler coarse-grid operators $\mathcal{B}_{p+1}\mathcal{S}_p^{(m\delta t)} = (I - \phi_{p+1}\mathcal{D}_{p+1})^{-1}\mathcal{S}_p^{(m\delta t)}$ of (4.22). For $\prod_{k=0}^{m-1}\mathcal{S}_p^{(\delta t)}$ and $\mathcal{B}_{p+1}\mathcal{S}_p^{(m\delta t)}$, only weights having a magnitude larger than that of 10^{-3} times the largest weight in their respective stencils are shown. The faint, dashed vertical line in each panel represents the location of the coarse-grid departure point.

coarse-grid semi-Lagrangian operator mimic the underlying behaviour of the advection problem, which is to transport data along characteristics, their stencils should reflect this (see Section 2.5.2 for more detailed arguments). Indeed, looking at the examples shown in Figure 4.4, it can be seen that this is the case, since weights are peaked around the departure point. The examples in Figure 4.4 also illustrate why $\mathcal{S}_p^{(m\delta t)}$ often serves as a poor approximation to the ideal operator $\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}$, as it does here. The stencil of $\mathcal{S}_p^{(m\delta t)}$ has only $p+1$ non-zero weights, which are centred about the departure point. In contrast, the stencil of $\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}$ has many more non-zero weights and they appear as a much smoother distribution about the departure point. This difference is a manifestation of the fact that the ideal operator is more dissipative than the semi-Lagrangian operator. Consider the proposed coarse-grid operators (4.21) and (4.22), which are $(I+\phi_{p+1}\mathcal{D}_{p+1})\mathcal{S}_p^{(m\delta t)}$ and $(I-\phi_{p+1}\mathcal{D}_{p+1})^{-1}\mathcal{S}_p^{(m\delta t)}$, respectively. Recall that $\mathcal{D}_{p+1} \approx h^{p+1}\,\mathrm{diag}\left(\frac{\partial^{p+1}}{\partial x^{p+1}}\right)$, such that when p is odd, the operators $(I + \phi_{p+1}\mathcal{D}_{p+1})$ and $(I - \phi_{p+1}\mathcal{D}_{p+1})^{-1}$ act in a dissipative fashion.[2] Therefore, one interpretation of the role of these operators in (4.21) and (4.22) is that they dissipate (i.e., loosely speaking, spread and smooth out) the weights in the stencil of the semi-Lagrangian operator $\mathcal{S}_p^{(m\delta t)}$. This effect can be seen most clearly in Figure 4.4 for the backward Euler operator $(I - \phi_{p+1}\mathcal{D}_{p+1})^{-1}\mathcal{S}_p^{(m\delta t)}$, where it has spread and smoothed the stencil of $\mathcal{S}_p^{(m\delta t)}$ so that it is a small perturbation of the ideal coarse grid operator's stencil.

4.2.4 Stability analysis

The purpose of this section is to assess the stability of the proposed forward and backward Euler coarse-grid operators (4.21) and (4.22). As we have seen, stability of a coarse-grid operator is by no means a sufficient condition for fast MGRIT convergence, but it is effectively a necessary one. Our analysis will show that for odd p the forward Euler operator is conditionally stable, with its stability limit characterized by the coarsening factor. Furthermore, for odd p, we show the backward Euler coarse-grid operator is unconditionally stable with respect to all problem parameters. We will also gain further insight into the structure of the operator that is to be inverted during the backward Euler step, and how this matrix depends on key parameters such as the coarsening factor and polynomial degree.

Our analysis relies on several assumptions, which we state here once at the beginning of the section for brevity. Given the divergence of MGRIT for even polynomial degrees p when using either of the forward or backward Euler coarse-grid operators (see Section 4.2.2), the analysis here considers only the case of odd p.

[2] At this stage, this statement should be interpreted loosely, since whether the operators act dissipately or anti-dissipatively (i.e., they exponentially blow up vectors rather than smooth them), depends on many problem parameters. This will be stated more rigorously in Section 4.2.4.

Assumption 4.1 (Coarse-grid semi-Lagrangian)**.** *Suppose that the circulant, coarse-grid semi-Lagrangian operator $\mathcal{S}_p^{(m\delta t)}$ uses interpolating polynomials of odd degree p. Furthermore, suppose that $\mathcal{S}_p^{(m\delta t)}$ is unconditionally stable in the ℓ^2-norm with $\left\|\mathcal{S}_p^{(m\delta t)}\right\|_2 = 1$.*

See [7] for a proof of this stability, and also the discussion in [31, Sec. 6.1.3] for further details.[3]

We will also fix the form of the approximation $\mathcal{D}_{p+1}$ as follows.

Assumption 4.2 (Spectrum of $\mathcal{D}_{p+1}$)**.** *Given that $p+1$ is even, assume that the approximation $\mathcal{D}_{p+1}$ is a circulant and symmetric finite-difference discretization of $h^{p+1}\operatorname{diag}\left(\frac{\partial^{p+1}}{\partial x^{p+1}}\right)$. Furthermore, assume that $\mathcal{D}_{p+1}$ is semi negative definite when $\frac{p+1}{2}$ is odd, and $\mathcal{D}_{p+1}$ is semi positive definite when $\frac{p+1}{2}$ is even. In other words, the spectrum σ of $\mathcal{D}_{p+1}$ satisfies*

$$\sigma(\mathcal{D}_{p+1}) \subseteq \begin{cases} \left[-\|\mathcal{D}_{p+1}\|_2, 0\right], & \frac{p+1}{2} \text{ is odd,} \\ \left[0, \|\mathcal{D}_{p+1}\|_2\right], & \frac{p+1}{2} \text{ is even.} \end{cases} \tag{4.30}$$

We do not know of a proof for the definiteness assumptions on $\mathcal{D}_{p+1}$. However, it is important to note that these assumptions hold numerically for all of the operators we have tested, and they certainly hold for the operators used in the numerical experiments we present in this chapter. It should also be noted that the differential operator that $\mathcal{D}_{p+1}$ approximates satisfies the continuous analogue of this assumption, as can be confirmed by Fourier analysis. That is, given the Fourier mode $e^{\mathrm{i}\beta x}$ with frequency $\beta \in \mathbb{R}$, one has $\frac{\partial^{p+1}}{\partial x^{p+1}} e^{\mathrm{i}\beta x} = -\beta^{p+1} e^{\mathrm{i}\beta x}$ if $\frac{p+1}{2}$ is odd, and $\frac{\partial^{p+1}}{\partial x^{p+1}} e^{\mathrm{i}\beta x} = \beta^{p+1} e^{\mathrm{i}\beta x}$ if $\frac{p+1}{2}$ is even; note that $\beta^{p+1} \geq 0$ since $p+1$ is even.

Having presented our assumptions, let us begin the analysis by simplifying our previous expressions for the coarse-grid operators at hand. Recall from (4.6) that the fine-grid departure point for the constant-wave-speed advection problem (4.1) is

$$\xi_i^{(t_n,\delta t)}(t_n) = x_i - \alpha\delta t \equiv x_i^{(\delta t)} - h\varepsilon^{(\delta t)}, \tag{4.31}$$

where $\varepsilon^{(\delta t)} \in [0,1)$ represents the mesh-normalized distance from the departure point $\xi_i^{(t_n,\delta t)}(t_n)$ to its east neighbouring mesh point $x_i^{(\delta t)}$. From this, notice that $x_i - \xi_i^{(t_n,\delta t)}(t_n) = \alpha\frac{\delta t}{h}h = \left(\frac{x_i - x_i^{(\delta t)}}{h} + \varepsilon^{(\delta t)}\right)h$, and therefore $\alpha\frac{\delta t}{h} = \frac{x_i - x_i^{(\delta t)}}{h} + \varepsilon^{(\delta t)}$. Since $\frac{x_i - x_i^{(\delta t)}}{h}$ is an integer (the mesh points in the numerator are separated by an integer multiple of h), and $x_i^{(\delta t)}$ is

[3] To see that the norm is equal to unity (as opposed to being bounded above by it, which would be sufficient for stability), note that the operator maps the constant vector to itself; in other words, the constant vector is an eigenvector and is associated with an eigenvalue of unity. This is because polynomial interpolation of any degree is exact on the constant function.

chosen so that $\varepsilon^{(\delta t)} \in [0,1)$, it follows that $\varepsilon^{(\delta t)}$ is the fractional part of the CFL number,

$$\varepsilon^{(\delta t)} = \alpha\frac{\delta t}{h} - \left\lfloor \alpha\frac{\delta t}{h} \right\rfloor \in [0,1). \tag{4.32}$$

Using this relationship, $\varepsilon^{(m\delta t)}$, which is the analogous quantity on the coarse grid, can be expressed in terms of $\varepsilon^{(\delta t)}$ as follows

$$\begin{aligned}
\varepsilon^{(m\delta t)} &= m\alpha\frac{\delta t}{h} - \left\lfloor m\alpha\frac{\delta t}{h} \right\rfloor, && (4.33)\\
&= m\varepsilon^{(\delta t)} + m\left\lfloor \alpha\frac{\delta t}{h} \right\rfloor - \left\lfloor m\varepsilon^{(\delta t)} + m\left\lfloor \alpha\frac{\delta t}{h} \right\rfloor \right\rfloor, && (4.34)\\
&= m\varepsilon^{(\delta t)} - \left\lfloor m\varepsilon^{(\delta t)} \right\rfloor \in [0,1). && (4.35)
\end{aligned}$$

Thus, the constant $\phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big)$ from (4.19) appearing in the coarse-grid operators depends only on the fine-grid quantity $\varepsilon^{(\delta t)}$. To simplify our notation, we define a new function that depends only on the fine-grid parameter as follows[4]

$$\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big) := \phi_{p+1}\big(\varepsilon^{(\delta t)}, \varepsilon^{(m\delta t)}\big) = f_{p+1}\big(m\varepsilon^{(\delta t)} - \lfloor m\varepsilon^{(\delta t)} \rfloor\big) - m f_{p+1}\big(\varepsilon^{(\delta t)}\big), \quad \varepsilon^{(\delta t)} \in [0,1). \tag{4.36}$$

Note that we have dropped the $(-1)^{p+1}$ factor from (4.19) since p is assumed to be odd under Assumption 4.1. The forward and backward Euler coarse-grid operators (4.21) and (4.22) can therefore be written respectively as

$$\begin{aligned}
\Phi^{(m\delta t)} &= \mathcal{F}_{p+1}\mathcal{S}_p^{(m\delta t)}, \quad \text{where } \mathcal{F}_{p+1} = I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}, && (4.37)\\
\Phi^{(m\delta t)} &= \mathcal{B}_{p+1}\mathcal{S}_p^{(m\delta t)}, \quad \text{where } \mathcal{B}_{p+1} = \Big[I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}\Big]^{-1}. && (4.38)
\end{aligned}$$

Now the stability analysis proceeds via a sequence of lemmas, beginning with the following which characterizes the stability of the operators in terms of γ_{p+1} and the spectrum of $\mathcal{D}_{p+1}$.

Lemma 4.4 (Conditions for stability)**.** *Suppose that Assumptions 4.1 and 4.2 hold. Then, the forward Euler coarse-grid operator* (4.37) *is ℓ^2-stable if*

$$\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1}) \subseteq [-2,0]. \tag{4.39}$$

Furthermore, the backward Euler coarse-grid operator (4.38) *is ℓ^2-stable if*

$$\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1}) \subseteq (-\infty,0] \cup [2,\infty). \tag{4.40}$$

[4]We choose not to redefine ϕ_{p+1} as depending on a single parameter because for variable-wave-speed problems it depends on multiple parameters.

Proof. The 2-norms of both coarse-grid operators may be written as

$$\left\|\left(I+\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\mathcal{D}_{p+1}\right)\mathcal{S}_p^{(m\delta t)}\right\|_2 = \max_{\lambda\in\sigma(\mathcal{D}_{p+1})}\left|1+\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\lambda\right|, \tag{4.41}$$

$$\left\|\left(I-\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\mathcal{D}_{p+1}\right)^{-1}\mathcal{S}_p^{(m\delta t)}\right\|_2 = \frac{1}{\min\limits_{\lambda\in\sigma(\mathcal{D}_{p+1})}\left|1-\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\lambda\right|}. \tag{4.42}$$

These expressions hold because $\mathcal{S}_p^{(m\delta t)}$ and $\mathcal{D}_{p+1}$ are simultaneously unitarily diagonalized by the discrete Fourier transform (they are both circulant), and because $\left\|\mathcal{S}_p^{(m\delta t)}\right\|_2 = 1$ under Assumption 4.1. Since $\mathcal{D}_{p+1}$ is symmetric (under Assumption 4.2), its eigenvalues are real. Using this result while enforcing that (4.41) and (4.42) are not greater than one, the results (4.39) and (4.40) respectively follow. ■

To understand the stability of the operators, it is therefore important to understand the behaviour of the constant $\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)$ given by (4.36). The polynomial f_{p+1} appearing in (4.36) is defined by (4.8), and for odd p, which uses symmetric interpolation nodes (see Section 4.2.1), it takes the form

$$f_{p+1}(z) = \frac{1}{(p+1)!}\prod_{q=-\frac{p+1}{2}}^{\frac{p-1}{2}}(q+z). \tag{4.43}$$

Key to understanding $\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)$ is first understanding $f_{p+1}(z)$. Several important properties of this function are summarized in Lemma B.1, which has been placed in Appendix B.2. Relying on Lemma B.1, the following lemma describes several important properties of γ_{p+1}.

Lemma 4.5 (Important properties of γ_{p+1})**.** *Suppose that Assumption 4.1 holds, that is, that p is odd, and let the constant $\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)$ be as in (4.36). Then, $\gamma_{p+1}(0) = 0$, and otherwise, the sign of this constant is*

$$\operatorname{sign}\left(\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\right) = \begin{cases} 1, & \varepsilon^{(\delta t)} \in (0,1), \quad \frac{p+1}{2} \text{ is odd,} \\ -1, & \varepsilon^{(\delta t)} \in (0,1), \quad \frac{p+1}{2} \text{ is even.} \end{cases} \tag{4.44}$$

Furthermore, suppose that the coarsening factor m is even, then the magnitude of $\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)$ may be bounded as[5]

$$\left|\gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\right| \le \left|\gamma_{p+1}\left(\tfrac{1}{2}\right)\right| = m\left|f_{p+1}\left(\tfrac{1}{2}\right)\right| \le m\sqrt{\frac{3}{p+2}}\left(\frac{1}{2}\right)^{p+2}, \quad \forall\varepsilon^{(\delta t)} \in [0,1), \quad \forall\frac{m}{2} \in \mathbb{N}. \tag{4.45}$$

[5]Bounds for odd m are not presented because the calculations are more complicated than for even m, and even coarsening factors are of greater interest to us since we typically coarsen by powers of two.

Proof. See Appendix B.3. ■

Using the previous lemmas, we are now able to make definitive statements on the stability of the coarse-grid operators, which we do in the following lemma.

Lemma 4.6 (Stability revisited)**.** *Suppose Assumptions 4.1 and 4.2 hold, then, the forward Euler coarse-grid operator* $\mathcal{F}_{p+1}\mathcal{S}_p^{(m\delta t)} = I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}\mathcal{S}_p^{(m\delta t)}$ *is conditionally* ℓ^2*-stable. When the coarsening factor* m *is even, a sufficient condition for stability of the operator that holds for any value of* $\varepsilon^{(\delta t)} \in [0,1)$ *is*

$$m \leq \left\lfloor \sqrt{\frac{p+2}{3}} \frac{2^{p+3}}{\|\mathcal{D}_{p+1}\|_2} \right\rfloor. \tag{4.46}$$

The backward Euler coarse-grid operator $\mathcal{B}_{p+1}\mathcal{S}_p^{(m\delta t)} = \Big[I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}\Big]^{-1}\mathcal{S}_p^{(m\delta t)}$ *is unconditionally* ℓ^2*-stable with respect to all problem parameters (i.e.,* $\varepsilon^{(\delta t)}$*,* m*,* p*, and* $\mathcal{D}_{p+1}$*).*

Proof. Let us begin with the backward Euler claim. Combining (4.30) on the spectrum of $\mathcal{D}_{p+1}$ with the sign result on $\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)$ of (4.44), we see that elements in the set $\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1})$ always lie on the non-positive real line,

$$\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1}) \subseteq (-\infty, 0], \quad \forall \varepsilon^{(\delta t)} \in [0,1). \tag{4.47}$$

From (4.40), this is a sufficient condition for the ℓ^2-stability of the backward Euler coarse-grid operator, and thus the operator is unconditionally stable.

For the forward Euler operator, the situation is more complicated. From (4.39), it is necessary for stability that the elements of $\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1})$ are non-positive, which is clearly the case given (4.47). However, it is also necessary that the minimum element of $\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1})$ be larger than -2. When m is even, we can use the bound of (4.45) to tighten the interval that these elements are in from (4.47) to

$$\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\sigma(\mathcal{D}_{p+1}) \subseteq \left[-m\sqrt{\frac{3}{p+2}}\left(\frac{1}{2}\right)^{p+2}\|\mathcal{D}_{p+1}\|_2, 0\right], \quad \forall \varepsilon^{(\delta t)} \in [0,1), \quad \forall \frac{m}{2} \in \mathbb{N}. \tag{4.48}$$

Ensuring that that the minimum element in this interval is larger than or equal to -2 therefore provides a sufficient condition for stability for the forward Euler operator. Imposing this inequality are rearranging for m yields

$$m \leq \sqrt{\frac{p+2}{3}} \frac{2^{p+3}}{\|\mathcal{D}_{p+1}\|_2}, \quad \forall\, \varepsilon^{(\delta t)} \in [0,1), \quad \forall \frac{m}{2} \in \mathbb{N}. \tag{4.49}$$

Finally, recognising that m is an integer, and therefore that the right-hand side of this bound only makes sense when it is an integer, leads to result (4.46). ■

Corollary 4.7 (Conditioning of backward Euler matrix)**.** *Suppose that Assumptions 4.1 and 4.2 hold, and that the coarsening factor m is even. Then, the matrix $\mathcal{B}_{p+1}^{-1}\big(\varepsilon^{(\delta t)}\big) = \Big[I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}\Big]$ to be inverted within the backward Euler coarse-grid operator (4.38) is symmetric positive definite, and its ℓ^2 condition number may be bounded as*

$$\kappa\left(\mathcal{B}_{p+1}^{-1}\big(\varepsilon^{(\delta t)}\big)\right) \leq 1 + m\sqrt{\frac{3}{p+2}}\left(\frac{1}{2}\right)^{p+2}\|\mathcal{D}_{p+1}\|_2, \quad \forall \varepsilon^{(\delta t)} \in [0,1), \quad \forall \frac{m}{2} \in \mathbb{N}. \tag{4.50}$$

Proof. From the proof of Lemma 4.6, we saw that $\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}$ is semi negative definite (see (4.47)), and since $\mathcal{D}_{p+1}$ is symmetric (under Assumption 4.2), it follows that $I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}$ is symmetric positive definite.

Since the matrix is symmetric positive definite, its ℓ^2-condition number can be expressed as

$$\kappa\left(\mathcal{B}_{p+1}^{-1}\big(\varepsilon^{(\delta t)}\big)\right) = \frac{\max\limits_{\lambda\in\sigma(\mathcal{D}_{p+1})} 1 - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\lambda}{\min\limits_{\lambda\in\sigma(\mathcal{D}_{p+1})} 1 - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\lambda} = 1 + \left|\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\right|\|\mathcal{D}_{p+1}\|_2. \tag{4.51}$$

The second equality occurs here because the denominator is equal to unity given that zero is the smallest eigenvalue of $-\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}$. The bound (4.50) then follows from bounding $\left|\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\right|$ using (4.45). ■

4.2.4.1 Discussion

The conditional stability of the forward Euler coarse-grid operator, as laid out by Lemma 4.6, is consistent with our earlier numerical results shown in Figure 4.3. Specifically, MGRIT convergence was fast for $m \in \{4, 16\}$, but then the solver diverged for $m = 64$. Indeed, considering numerically the eigenvalues in those cases (not shown here for brevity) shows that the coarse-grid operator is stable when $m \in \{4, 16\}$, and unstable for $m = 64$. This behaviour is perhaps unsurprising in a qualitative sense, given the interpretation in Section 4.2.3.1 of this operator as discretizing explicitly in time a high-order PDE.[6]

[6]The classical example of this phenomenon is that a forward Euler time discretization of the diffusion equation $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ is stable only under the CFL constraint $\delta t \lesssim \mathcal{O}(h^2)$ when using finite-differences in space.

Unfortunately, the conditional stability of this operator renders it effectively useless in practice.[7] Using a 2nd-order finite-difference approximation for $\mathcal{D}_{p+1}$, bound (4.46) reduces to (to two decimal places) the forward Euler operator being stable for even coarsening factors $m \leq (\lfloor 4.00 \rfloor, \lfloor 5.16 \rfloor, \lfloor 6.11 \rfloor) = (4, 4, 6)$ for $p = (1, 3, 5)$. Being restricted to such small coarsening factors is impractical, because the resulting coarse-grid problem is not significantly smaller than the fine-grid problem, and therefore could not lead to any speed-up in a parallel MGRIT solve, no matter how fast the convergence.

An interesting question then is whether employing an analogous operator in a multilevel context with slow coarsening (i.e., with sufficiently small m between each level) would be stable. Unfortunately, this is not the case. For the special case of constant-wave-speed advection, the multilevel operator to be introduced in Section 4.4.1 is closely related to the two-level operator presented thus far, and in fact, a generalized stability condition based on (4.46) applies. More specifically, a sufficient condition for stability of a level $\ell \in \mathbb{N}$ forward Euler coarse-grid operator (arising from generalizing our current two-level operator) using a time-step $m^\ell \delta t$ is (4.46) with $m \mapsto m^\ell$. Introducing any kind of slow multilevel coarsening therefore does not improve the poor stability properties of the operator. Thus, the operator is as impractical in the multilevel setting as it is in the two-level setting.

For the above reasons, we do not consider the forward Euler coarse-grid operator throughout the rest of this chapter. Before moving on, however, it is interesting to think about the underlying reasons for this instability. Recall that the operator explicitly discretizes the lowest-order difference between the truncation error of the ideal coarse-grid operator and the coarse-grid semi-Lagrangian operator. Given that both operators are stable and explicit-in-time (i.e., they are sparse), it seems somewhat counter-intuitive that an explicit-in-time discretization of the (lowest-order) difference between them would be unstable. In Chapter 2, we identified sparse coarse-grid operators through an optimization process. While these coarse-grid operators were for Eulerian discretizations of advection rather than semi-Lagrangian ones, we identified empirically the heuristic that if one is to obtain a convergent MGRIT solver, the width of the coarse-grid stencil needs to increase with the coarsening factor. Notice, however, that the stencil of the forward Euler coarse-grid operator (4.37) is constant with respect to m. Our findings here therefore are consistent with the results of Chapter 2.

We now move on to the backward Euler coarse-grid operator. This operator is clearly more suitable as a coarse-grid operator than the forward Euler operator, in the sense that it is unconditionally stable. As discussed previously, this is not a sufficient condition for

[7] This statement presupposes that the statement (4.46) is close to being a necessary and sufficient condition rather than just a sufficient one; that is, the bound is tight. However, this is essentially the case since (4.46) becomes necessary and sufficient when both $\varepsilon^{(\delta t)} = 0.5$ and $p = 1$.

fast MGRIT convergence, but it is effectively a necessary one. It is difficult to analytically understand more about how the convergence of MGRIT depends on our coarse-grid operator. Nonetheless, our previous numerical results (see Figure 4.3) seem to indicate the operator leads to fast MGRIT convergence. Furthermore, we find that the operator satisfies several *empirical rules* that we identified for effective coarse-grid operators in our previous work of Chapter 2.

The condition number bound of Corollary 4.7 on $\mathcal{B}_{p+1}^{-1} = I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}$ provides us with some insight into the difficulty of inverting this matrix, which is required to compute the action of the coarse-grid operator. In particular, the matrix is fairly well-conditioned, with a condition number scaling only as $\sim \mathcal{O}(m)$. For $m = 128$, Figure 4.5 shows a plot of the condition number $\mathcal{B}_{p+1}^{-1}$ as a function of $\varepsilon^{(\delta t)}$ for several p, along with the associated bound from Corollary 4.7. There are a few points to note. Firstly, the bound is tight in that it achieves equality for $p = 1$ when $\varepsilon^{(\delta t)} = 0.5$. For $p > 1$ the bound provides only a small overestimation of the true condition number at $\varepsilon^{(\delta t)} = 0.5$ (from the proof of Lemma 4.5 on the bound of $\big|\gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\big|$, the careful reader may have realised that this would be the case). Secondly, when using a 2nd-order accurate finite-difference operator for $\mathcal{D}_{p+1}$, conditioning improves with polynomial order p. This is driven by the fact that $|\gamma_{p+1}|$ decreases with increasing p, as per the bound (4.45) of Lemma 4.5.

At first, it seems somewhat counter-intuitive that the ideal coarse-grid operator is a sparse matrix, yet our backward Euler coarse-grid operator is dense, since it involves the dense

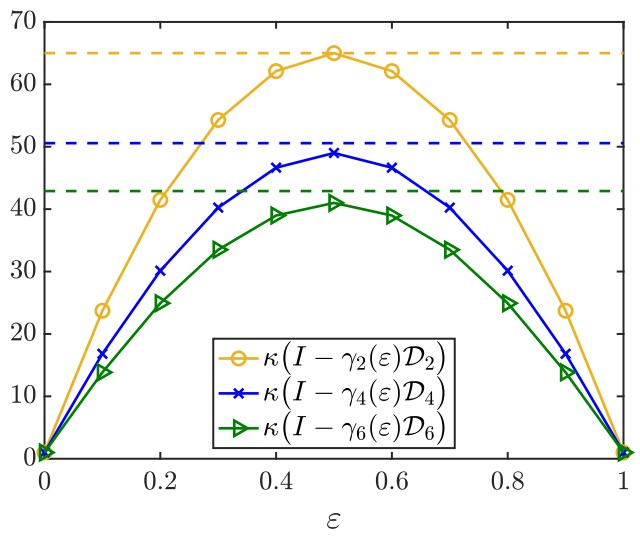

FIGURE 4.5: The ℓ^2-condition number κ of the backward Euler matrix $\mathcal{B}_{p+1}^{-1}(\varepsilon) = I - \gamma_{p+1}(\varepsilon)\mathcal{D}_{p+1}$ that corresponds to a coarsening factor of $m = 128$ is shown for polynomial degrees $p \in \{1, 3, 5\}$. Exact condition numbers (markers) are shown for several values of $\varepsilon \in [0, 1)$, as is the upper bound (dashed lines) of (4.50) from Corollary 4.7 which holds $\forall \varepsilon \in [0, 1)$. In these examples, $\mathcal{D}_{p+1}$ is a 2nd-order accurate finite-difference operator.

matrix $\mathcal{B}_{p+1} = \left[I - \gamma_{p+1}\big(\varepsilon^{(\delta t)}\big)\mathcal{D}_{p+1}\right]^{-1}$. However, as we now discuss, this operator effectively acts as though it is sparse, and can be well-approximated by a sparse operator. Aside from describing the difficulty of inverting $\mathcal{B}_{p+1}^{-1}$ with an iterative method, the condition number bound (4.50) provides us with important information about the structure of the inverse $\mathcal{B}_{p+1}$ itself. It is well known that under certain conditions, entries in the inverse of a banded matrix decay exponentially (in magnitude) away from the main diagonal. In particular, suppose that a matrix A is symmetric positive definite and k-banded, then the entries in its inverse may be bounded as [27, Th. 2.4]

$$\left|A_{ij}^{-1}\right| \leq c\big[\varrho(A)\big]^{|i-j|}, \quad \varrho(A) = \left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^{2/k}, \tag{4.52}$$

in which c is a constant depending on $\kappa(A)$ and $\|A^{-1}\|_2$. In (4.52), the bandwidth k of the symmetric matrix A is defined such that $A_{ij} = 0$ if $|i-j| > k/2$. The matrix $\mathcal{B}_{p+1}^{-1}$ is symmetric positive definite (see Corollary 4.7), and is banded, since $\mathcal{D}_{p+1}$ is banded. Thus, the entries in its inverse decay exponentially, with a rate at least as fast as $\big[\varrho(\mathcal{B}_{p+1}^{-1})\big]^{|i-j|}$. If this decay is fast enough, the matrix should be well-approximated by a sparse one. Our numerical tests in Section 4.4.1, in which we approximate the action of this inverse through GMRES, do indeed indicate that this matrix can be well-approximated by a sparse matrix.

Notice that ϱ in (4.52) is an increasing function of κ, and thus generally speaking, larger condition numbers yield slower decay rates. Furthermore, since $\varrho(\mathcal{B}_{p+1}^{-1})$ is an increasing function of $\kappa(\mathcal{B}_{p+1}^{-1})$, it can be bounded for all $\varepsilon^{(\delta t)} \in [0,1)$ by using the bound (4.50) that holds for all $\varepsilon^{(\delta t)} \in [0,1)$. Recalling that $\kappa(\mathcal{B}_{p+1}^{-1}) \lesssim \mathcal{O}(m)$, and supposing that (4.52) is tight with respect to $\kappa(\mathcal{B}_{p+1}^{-1})$, we can argue that the rate of exponential decay of entries in $\mathcal{B}_{p+1}$ is slower for larger coarsening factors. Numerically computing entries in the dense matrix $\mathcal{B}_{p+1}$ (see Figure 4.6 for a specific example), we find that they do indeed exponentially decay with a fast rate, and that the decay is slower for increasing m. This justifies our claim that while the operator is dense, it effectively acts as if it is sparse. Furthermore, this behaviour is consistent with the findings in Chapter 2 and our previous discussion on the forward Euler operator, regarding the need for the number of non-zeros in the stencil to increase with m. In other words, if we were to approximate $\mathcal{B}_{p+1}$ with a sparse matrix, the sparsity of the approximation would need to decrease as m increases.

Interestingly, when numerically computing entries in $\mathcal{B}_{p+1}$, the decay rate upper bound of (4.52) generally only seems to be tight in the $p = 1$ case. As shown in Figure 4.6, for example, when $p = 5$ the entries decay much faster than the upper bound predicts. In fact, for a given m, we typically find that the rate of decay rate increases with polynomial

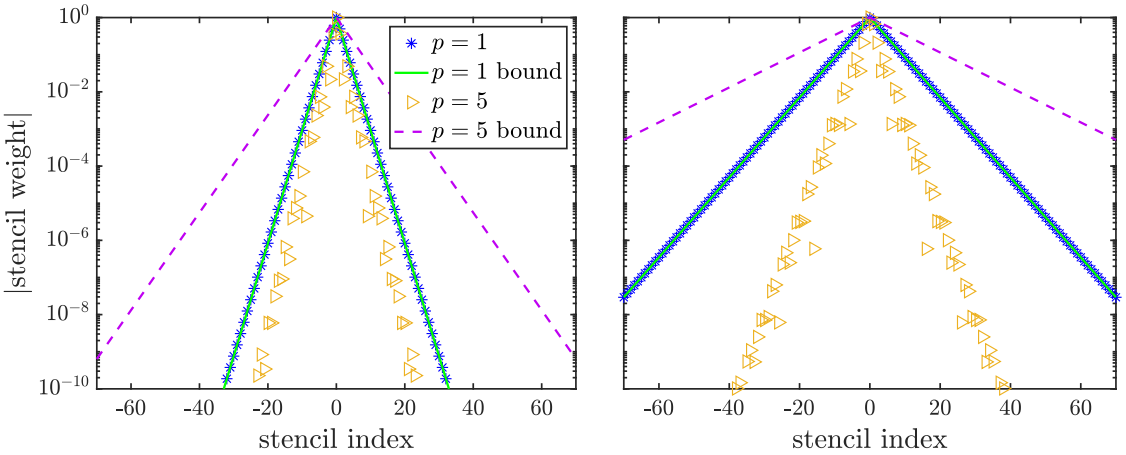

FIGURE 4.6: The absolute value of weights in the stencil of the dense matrix $\mathcal{B}_{p+1} = \left[I - \gamma_{p+1}\left(\varepsilon^{(\delta t)}\right)\mathcal{D}_{p+1}\right]^{-1} \in \mathbb{R}^{1024\times 1024}$ with $\varepsilon^{(\delta t)} = 0.85$, and $p \in \{1, 5\}$. Note that only weights with magnitude greater than 10^{-10} are shown, and that only those associated with indices in the interval $[-70, 70]$ are shown. The markers represent the weights themselves, and the dashed lines represent the upper bound of them given by (4.52). Note that the weights and their bounds have been normalized to have a maximum value of one. Note that $\mathcal{D}_{p+1}$ is taken to be a 2nd-order accurate finite-difference operator. **Left:** $m = 32$. **Right:** $m = 256$.

degree p, yet the bound (4.52) predicts the opposite.[8] There is a large amount of literature on decay rates for banded matrices, so it is likely that tighter bounds than those of (4.52) exist for our problems, however, we leave this to future investigation. Finally, note that the trend we observe numerically of the decay rate increasing with p is consistent with yet another empirical finding we made about the coarse-grid stencil in Chapter 2, which was that the required increase in its width was smaller for higher-order discretizations.

4.3 Variable-wave-speed advection

In this section, we move to consider variable-wave-speed problems of the form

$$\frac{\partial u}{\partial t} + \alpha(x,t)\frac{\partial u}{\partial x} = 0, \quad (x,t) \in \Omega \times (0,T], \quad u(x,0) = u_0(x), \tag{4.53}$$

for spatial domain $\Omega \subset \mathbb{R}$, and solution u subject to periodic boundary conditions on $\partial\Omega$. The semi-Lagrangian discretization of (4.53) shares many similarities with that described in Section 4.2.1 for the case of constant α. However, there are now more possibilities for generalization, and, so, there are many different types of semi-Lagrangian methods. We will consider, in a sense, the simplest possible class of semi-Lagrangian discretizations that assume a high-degree of smoothness of the solution of (4.53), and that the wave-speed

[8] To understand why (4.52) predicts an increase with p, note that while $\kappa(\mathcal{B}_{p+1}^{-1})$ does indeed decrease with increasing p (see Figure 4.5), the bandwidth k of $\mathcal{B}_{p+1}^{-1}$ is $p+1$ for 2nd-order accurate approximations $\mathcal{D}_{p+1}$, since the finite-difference stencil of $\mathcal{D}_{p+1}$ uses $p+2$ points for such cases.

$\alpha(x,t)$ is readily available for any x and t. The reader is directed to [31] and [29, Sec. 7] (and references therein) for discussions on more sophisticated semi-Lagrangian methods.

We now briefly highlight the differences of the semi-Lagrangian discretization of (4.53) from that described in Section 4.2.1 for constant α. The Lagrangian formulation of (4.53) is

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = \alpha(\xi(t),t). \tag{4.54}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t),t) = 0, \quad t \in (0,T], \quad u(x,0) = u_0(x). \tag{4.55}$$

From (4.54), the characteristics are now curved in general rather than straight lines. From (4.55), the PDE solution still remains constant along a given characteristic.

Recall that we define the local characteristic $\xi_i^{(t_n,\delta t)}(t)$ to be that which passes through the *arrival point* $(x,t) = (x_i, t_n + \delta t)$. Then, we locate the associated *departure point* $(x,t) = \big(\xi_i^{(t_n,\delta t)}(t_n), t_n\big)$ by solving the final-value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_i^{(t_n,\delta t)}(t) = \alpha\big(\xi_i^{(t_n,\delta t)}(t), t\big), \quad t \in [t_n, t_n + \delta t), \quad \xi_i^{(t_n,\delta t)}(t_n + \delta t) = x_i. \tag{4.56}$$

This solution of this problem can no longer be found exactly, and instead needs to be approximated with a numerical integration method. We will do so by integrating backwards along the characteristic using a single explicit Runge-Kutta (ERK) step of size δt, although there are many other possible ways to do this.[9] The Butcher tableaux for the ERK schemes we use are given as ERK1, ERK3, and ERK5 in Appendix A.1. Furthermore, suppose that the ERK method has a global accuracy of order r, such that each departure point is located with an accuracy of $\mathcal{O}(\delta t^{r+1})$ (e.g., forward Euler has $r = 1$).

Upon (approximately) locating a departure point, we again fit a degree (at most) p polynomial through its nearest neighbouring mesh points. To perform this interpolation at a given departure point, we decompose it as

$$\xi_i^{(t_n,\delta t)}(t_n) \equiv x_i^{(t_n,\delta t)} - h\varepsilon_i^{(t_n,\delta t)}, \quad \varepsilon_i^{(t_n,\delta t)} \in [0,1), \tag{4.57}$$

in which $x_i^{(t_n,\delta t)}$ is the mesh node immediately east of $\xi_i^{(t_n,\delta t)}(t_n)$, and $\varepsilon_i^{(t_n,\delta t)}$ is its (mesh-normalized) distance from this point. See Figure 4.7 for a schematic example. Notice that the quantities $x_i^{(t_n,\delta t)}$ and $\varepsilon_i^{(t_n,\delta t)}$ now depend on the departure time t_n and the location i in space.

[9] On coarse levels in our multigrid hierarchy, we will explore several alternative possibilities rather than simply rediscretizing the ERK scheme.

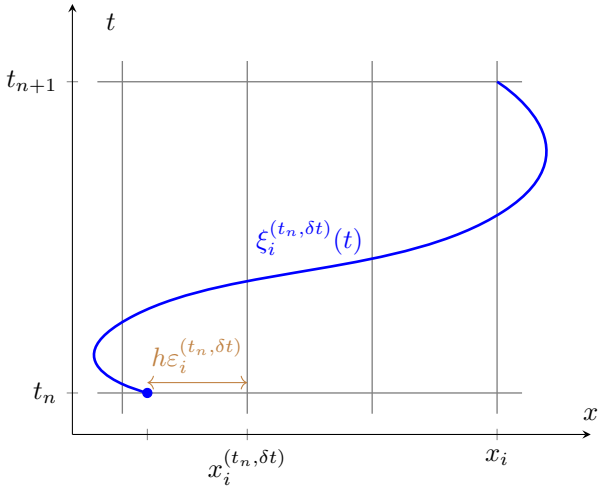

FIGURE 4.7: A characteristic $\xi_i^{(t_n,\delta t)}(t)$ for $t \in [t_n, t_n + \delta t]$ of (4.53). By definition, the characteristic passes through the arrival point $(x, t) = (x_i, t_{n+1})$. The departure point is the location on the x-axis of the characteristic at $t = t_n$. The departure point is decomposed into the sum of its east-neighbouring mesh point and its distance from this point, as in (4.57).

We denote the time-stepping operator corresponding to the above described semi-Lagrangian method as $\mathcal{S}_{p,r}^{(t_n,\delta t)} \in \mathbb{R}^{n_x \times n_x}$. By tracking characteristics (with a sufficient level of accuracy), this discretization ensures the physical domain of dependence lies within the numerical domain of dependence. Generally speaking, this is why semi-Lagrangian discretizations are typically free of a CFL constraint. However, ensuring that characteristics are tracked with sufficient accuracy can lead to the imposition of a CFL-like constraint, although it is typically looser than that imposed by Eulerian schemes [59, 86]. For sufficiently smooth solutions of (4.53), it can be argued that $\mathcal{S}_{p,r}^{(t_n,\delta t)}$ has a convergence rate of the form $\mathcal{O}\big(\delta t^r + \frac{h^{p+1}}{\delta t}\big)$, where the first term is associated with the locating of departure points, and the second with the polynomial interpolation at them [31, Sec. 6.1.2]. Thus, while one can certainly maintain stability with large time steps, there can be a trade-off with accuracy in doing so. In many applications, this accuracy trade-off is not realized until the time-step is much larger than what would be allowed by a CFL-constrained Eulerian scheme, which is why semi-Lagrangian methods are often preferable for such applications (see Chapter 1). Nonetheless, we attempt to balance temporal and spatial errors by using $\delta t \sim h$ and $r = p$, which is common in the literature when developing semi-Lagrangian discretizations [74, 59, 17, 16].

Before moving to discuss our coarse-grid operator for this problem, we describe our test problems. We will consider the advection equation (4.53) with wave-speeds given by

$$\alpha(x,t) = \cos(2\pi x)\cos(2\pi t), \tag{4.58}$$

$$\alpha(x,t) = \left[1 + \frac{1}{4}\cos(2\pi x)\right]^2 \cos\left(\frac{\pi t}{8}\right). \tag{4.59}$$

We will use the initial condition $u_0(x) = \sin^4(\pi x)$, spatial domain $\Omega = (-1, 1)$, a (fine-grid) time-step of $\delta t = 0.85h$, and a final integration time $T \approx 13.6$. These problems are challenging from a discretization point of view because the wave-speeds are highly variable over the space-time domain, resulting in highly variable characteristics. Moreover, the time domain is of significant length, so that accurate discretizations will be needed to achieve even modest errors at time T. For the above reasons, we also believe these will serve as difficult problems for MGRIT. In particular, note that a wave-speed passing through zero somewhere in the domain was shown to slow MGRIT convergence considerably in [58], albeit in the context of employing coarsening in space as well as time.

Figure 4.8 shows the PDE solution for wave-speed (4.58) and the associated semi-Lagrangian discretization errors. The discretization error has been computed using the exact solution of the PDE, which is easily obtained from the class of exact solutions derived in Appendix B.4. Figure 4.9 shows the PDE solution for wave-speed (4.59), as well as some characteristics and the velocity field over the space-time domain.

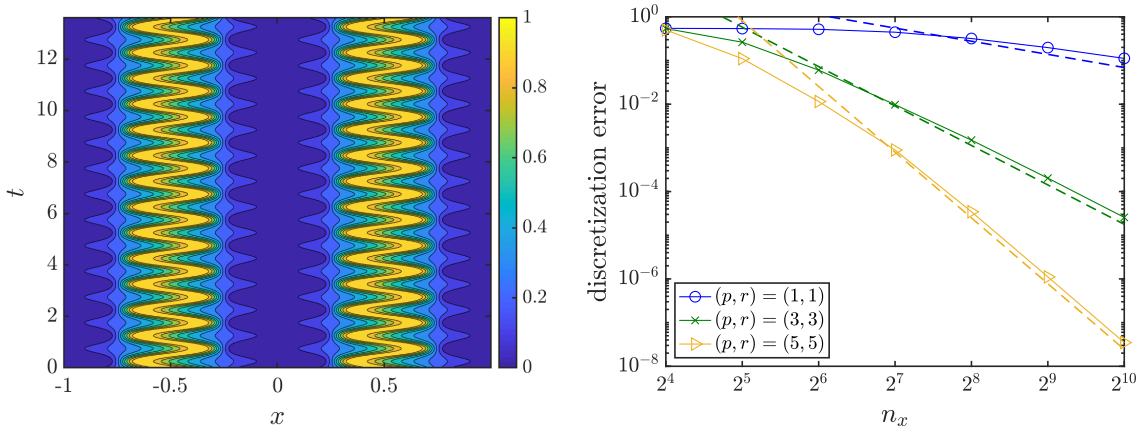

FIGURE 4.8: Solution of advection problem (4.53) with wave-speed (4.58). **Left:** Solution over the space-time domain. **Right:** Discretization error of the semi-Lagrangian discretization $\mathcal{S}_{p,r}^{(t_n,\delta t)}$ in the discrete ℓ^2-norm measured at the final time. The spatial mesh size is $h = 2/n_x$. Dashed lines indicate the expected asymptotic convergence rate of $\mathcal{O}(h^p)$.

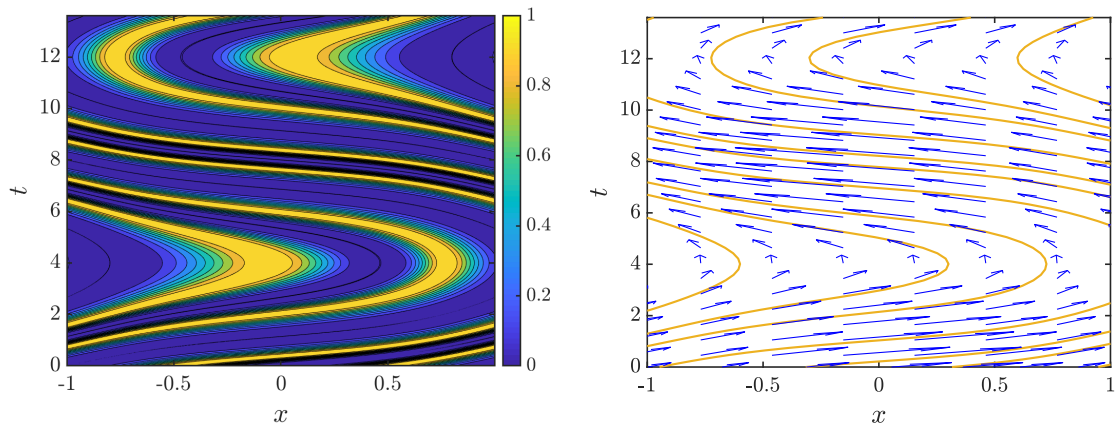

FIGURE 4.9: Solution of advection problem (4.53) with wave-speed (4.59). **Left:** Solution over the space-time domain. **Right:** Some characteristics (solid gold lines), and the space-time velocity field (blue arrows) of (4.53), which is $(\alpha(x,t),1)$. The length and direction of a given arrow represents the magnitude and direction of the velocity field at that point in space-time.

4.3.1 Exact departure points

The question now is how should we generalize the coarse-grid operator from Section 4.2 for constant wave-speeds to variable wave-speeds. There are two complicating factors that make this difficult. The first is the fact that a spatially varying wave-speed leads to non-equidistant departure points, which gives rise to a more complicated truncation error associated with the polynomial interpolation aspect of the method. The second issue is that departure points are no longer located exactly, but only approximately via some numerical integrator. Let us deal first with the non-equidistant departure points here, then in Sections 4.3.2 and 4.3.3 we will deal with the inexact locating of departure points.

We begin with the following description of the truncation error for an idealized semi-Lagrangian discretization that locates departure points exactly.

Lemma 4.8 (Semi-Lagrangian truncation error for $r=\infty$)**.** *Suppose that the solution $u(x,t)$ of (4.53) is at least $p+1$ times continuously differentiable with respect to x. Let $\mathcal{S}_{p,\infty}^{(t_n,\delta t)}$ be the previously described semi-Lagrangian discretization of (4.53) that locates departure points exactly at time t_n. Then, this operator has a local truncation error given by*

$$\begin{aligned} &u(\boldsymbol{x},t_{n+1}) - \mathcal{S}_{p,\infty}^{(t_n,\delta t)} u(\boldsymbol{x},t_n) \\ &\quad = (-h)^{p+1}\,\mathrm{diag}\left(f_{p+1}\left(\boldsymbol{\varepsilon}^{(t_n,\delta t)}\right)\right)\frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x},t_{n+1}) + \mathcal{O}(h^{p+2}), \end{aligned} \tag{4.60}$$

in which $\boldsymbol{\varepsilon}^{(t_n,\delta t)} = \left(\varepsilon_1^{(t_n,\delta t)},\ldots,\varepsilon_{n_x}^{(t_n,\delta t)}\right)^\top \in \mathbb{R}^{n_x}$ is the vector whose ith entry is the mesh-normalized distance from the ith departure point to its east-neighbouring mesh point at

time t_n, $\xi_i^{(t_n,\delta t)}(t_n) = x_i^{(t_n,\delta t)} - h\varepsilon_i^{(t_n,\delta t)}$. The polynomial f_{p+1} (as defined in (4.8)) is applied to $\boldsymbol{\varepsilon}^{(t_n,\delta t)}$ in the element-wise sense.

Furthermore, the associated ideal coarse-grid operator defined by time-stepping m times across the interval $t \in [t_n, t_n + m\delta t]$ using the above operator has a local truncation error that may be approximated by

$$\begin{aligned} u(\boldsymbol{x}, t_{n+m}) &- \left[\prod_{k=0}^{m-1} \mathcal{S}_{p,\infty}^{(t_n+k\delta t,\delta t)}\right] u(\boldsymbol{x}, t_n) \\ &\approx (-h)^{p+1} \operatorname{diag}\left(\sum_{k=0}^{m-1} f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n+k\delta t,\delta t)}\big)\right) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+m}) + \mathcal{O}(h^{p+2}), \end{aligned} \tag{4.61}$$

with the approximation becoming an equality for spatially independent wave-speeds $\alpha(x,t) \equiv \alpha(t)$.

Proof. Since there is no truncation error component associated with locating departure points, the proof of the first result (4.60) essentially follows along the same lines as that for the constant-coefficient result in Lemma 4.1. The only modification now is that vector $\boldsymbol{\varepsilon}^{(t_n,\delta t)}$ is not constant. The effect of this is that the standard error estimate for polynomial interpolation (see [24, Th. 3.1.1]) at the ith departure point at time t_n gives

$$\begin{aligned} u\big(\xi_i^{(t_n,\delta t)}(t_n), t_n\big) &- \Big(\mathcal{S}_{p,\infty}^{(t_n,\delta t)} u(\boldsymbol{x}, t_n)\Big)_i \\ &= \frac{1}{(p+1)!} \prod_{q=-\ell(p)}^{r(p)} \left[\big(x_i^{(t_n,\delta t)} - h\varepsilon_i^{(t_n,\delta t)}\big) - \big(x_i^{(t_n,\delta t)} + hq\big)\right] \frac{\partial^{p+1} u}{\partial x^{p+1}}\bigg|_{(\zeta_i, t_n)}, \end{aligned} \tag{4.62}$$

$$= (-h)^{p+1} f_{p+1}\big(\varepsilon_i^{(t_n,\delta t)}\big) \frac{\partial^{p+1} u}{\partial x^{p+1}}\bigg|_{(\zeta_i, t_n)}, \tag{4.63}$$

for some unknown point ζ_i in the interval containing all the interpolation nodes, $\big(x_i^{(t_n,\delta t)} - h\ell(p), x_i^{(t_n,\delta t)} + hr(p)\big)$. The rest of the proof for (4.60) follows the same steps as in first part of the proof in Lemma 4.1.

Now let us move on to proving (4.61). Applying $\mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)}$ to both sides of (4.60) gives

$$\begin{aligned} \mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)} u(\boldsymbol{x}, t_{n+1}) &- \mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)} \mathcal{S}_{p,\infty}^{(t_n,\delta t)} u(\boldsymbol{x}, t_n) \\ &= (-h)^{p+1} \mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)} \operatorname{diag}\big(f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,\delta t)}\big)\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+1}) + \mathcal{O}(h^{p+2}), \end{aligned} \tag{4.64}$$

$$\approx (-h)^{p+1} \operatorname{diag}\big(f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,\delta t)}\big)\big) \frac{\partial^{p+1}}{\partial x^{p+1}} \Big(\mathcal{S}_{p,\infty}^{(t_n+1,\delta t)} u(\boldsymbol{x}, t_{n+1})\Big) + \mathcal{O}(h^{p+2}). \tag{4.65}$$

The approximation introduced in arriving at (4.65) is that the matrices $\mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)}$ and $\operatorname{diag}\big(f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,\delta t)}\big)\big)$ commute. The latter is a diagonal matrix, while the former is not, and thus they will only commute in the case that the vector $\boldsymbol{\varepsilon}^{(t_n,\delta t)}$ is constant. For spatially independent wave-speeds $\alpha(x,t) \equiv \alpha(t)$, all characteristics have the same slope

as one another at a given time t, and thus neighbouring departure points are equispaced by a distance h, meaning that $\boldsymbol{\varepsilon}^{(t_n,\delta t)}$ will be constant. For spatially variable wave-speeds, however, departure points are offset by a non-constant amount, and thus $\boldsymbol{\varepsilon}^{(t_n,\delta t)}$ is not constant. We can expect this commutation error not to be too large for a problem that is sufficiently resolved in space and whose wave-speed does not vary too quickly with respect to x, since such a situation would result in a slowly varying $\boldsymbol{\varepsilon}^{(t_n,\delta t)}$.

Now substitute into (4.65) the expression for $\mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)} u(\boldsymbol{x}, t_{n+1})$ from (4.60) to give

$$\begin{aligned} u(\boldsymbol{x}, t_{n+2}) &- \left[\mathcal{S}_{p,\infty}^{(t_{n+1},\delta t)} \mathcal{S}_{p,\infty}^{(t_n,\delta t)}\right] u(\boldsymbol{x}, t_n) \\ &\approx (-h)^{p+1} \operatorname{diag}\left(f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,\delta t)}\big) + f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_{n+1},\delta t)}\big)\right) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+2}) + \mathcal{O}(h^{p+2}). \end{aligned} \tag{4.66}$$

Inductively repeating the above process on (4.66) with the remaining $m-1$ fine-grid operators $\mathcal{S}_{p,\infty}^{(t_{n+j},\delta t)}$, $j \in \{2, \ldots, m-1\}$, one arrives at the result (4.61). ■

Having now developed an approximate expression for the truncation error of the ideal coarse-grid operator, we propose a coarse-grid operator that generalizes what we did previously for the constant-wave-speed problem in Section 4.2. That is, we create a coarse-grid operator expressed as a perturbed semi-Lagrangian operator, with the perturbation capturing approximately the lowest-order difference between the truncation error of the semi-Lagrangian operator and the ideal coarse-grid operator. Recall that in the constant-wave-speed case, the explicit Euler coarse-grid operator (4.21) was of no practical use because of its poor stability properties (see Section 4.2.4). Conversely, the implicit Euler operator (4.22) was shown to be unconditionally stable (see Section 4.2.4), and it led to fast MGRIT convergence in the numerical tests. For these reasons, we propose to use the following backward Euler coarse-grid operator for semi-Lagrangian discretizations $\mathcal{S}_{p,\infty}^{(t_n,\delta t)}$ of advection problem (4.53)

$$\Phi^{(t_n,m\delta t)} = \mathcal{B}_{p+1}^{(t_n,m\delta t)} \mathcal{S}_{p,\infty}^{(t_n,m\delta t)}, \tag{4.67}$$

where $\mathcal{B}_{p+1}^{(t_n,m\delta t)}$ is the following matrix

$$\mathcal{B}_{p+1}^{(t_n,m\delta t)} := \left[I - \operatorname{diag}\left(\varphi_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,\delta t)}, \ldots, \boldsymbol{\varepsilon}^{(t_{n+m-1},\delta t)}, \boldsymbol{\varepsilon}^{(t_n,m\delta t)}\big)\right)\mathcal{D}_{p+1}\right]^{-1}, \tag{4.68}$$

$$= \left[I - \operatorname{diag}\left(f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,m\delta t)}\big) - \sum_{k=0}^{m-1} f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_{n+k},\delta t)}\big)\right)\mathcal{D}_{p+1}\right]^{-1}, \tag{4.69}$$

where, as previously, $\mathcal{D}_{p+1} \approx h^{p+1} \operatorname{diag}\left(\frac{\partial^{p+1}}{\partial x^{p+1}}\right) \in \mathbb{R}^{n_x \times n_x}$ is a finite-difference approximation. The function φ_{p+1} is defined in terms of the degree $p+1$ polynomial f_{p+1} (see (4.8))

as follows

$$\varphi_{p+1}(y_0, \ldots, y_{m-1}, y_m) := f_{p+1}(y_m) - \sum_{j=0}^{m-1} f_{p+1}(y_j). \tag{4.70}$$

In (4.68), φ_{p+1} is applied element-wise to the vectors $\boldsymbol{\varepsilon}^{(t_n,\delta t)}, \ldots, \boldsymbol{\varepsilon}^{(t_{n+m-1},\delta t)}, \boldsymbol{\varepsilon}^{(t_n,m\delta t)}$. Notice that φ_{p+1} generalizes the function ϕ_{p+1} in (4.19); that is, for constant wave-speed, and assuming $p+1$ is even, (4.70) reduces to

$$\varphi_{p+1}(y_0, \ldots, y_0, y_m) = f_{p+1}(y_m) - m f_{p+1}(y_0) = \phi_{p+1}(y_0, y_m). \tag{4.71}$$

Through the use of highly accurate numerical integration to locate departure points (e.g., MATLAB's adaptive step-size `ode45`), we have run numerical experiments using coarse-grid operator (4.67) for the variable-wave-speed problems (4.58) and (4.59). Generally speaking, the resulting MGRIT convergence has been fast for all our test problems. Since we are ultimately interested in the more practical use case of inexactly locating departure points, for the sake of brevity, we do not show the results of these intermediate numerical tests.

Remark 4.9 (Obtaining $f_{p+1}(\varepsilon)$ for free)**.** *The proposed coarse-grid operators in this work require evaluating the $p+1$st-degree polynomial f_{p+1} from (4.8) at the points ε (recall that for each departure point, ε is the mesh-normalized distance to its east-neighbouring mesh point). Recall that we approximate the value of $u(x)$ at some departure point of the form $x = \xi = x_k - h\varepsilon$ by interpolating u with a degree (at most) p polynomial at the $p+1$ mesh points that are nearest to the departure point. Denote such a polynomial by $\hat{u}_p(x)$, and for simplicity of notation, let the departure point be $\xi = -h\varepsilon$. Then, $\hat{u}_p(x)$ will interpolate $u(x)$ at the $p+1$ values $\{u_{-\ell(p)}, \ldots u_{r(p)}\}$ at the equidistant nodes $\{x_{-\ell(p)}, \ldots x_{r(p)}\}$. In the current context, where the interpolation is associated with a stencil/mesh-based discretization, it is most natural to consider this interpolation in the Lagrange basis. In doing so, the interpolating polynomial can be written as*

$$\hat{u}_p(x) = \sum_{j=-\ell(p)}^{r(p)} \left(\prod_{\substack{q=-\ell(p) \\ q\neq j}}^{r(p)} \frac{x - x_q}{x_j - x_q} \right) u_j, \tag{4.72}$$

$$= \left[\prod_{s=-\ell(p)}^{r(p)} (x - x_s) \right] \left[\sum_{j=-\ell(p)}^{r(p)} \left(\frac{1}{x - x_j} \prod_{\substack{q=-\ell(p) \\ q\neq j}}^{r(p)} \frac{1}{x_j - x_q} \right) u_j \right], \tag{4.73}$$

with the second expression holding so long as x is not an interpolation node. Evaluating this second expression at $x = -h\varepsilon$ gives the approximation

$$\hat{u}_p(-h\varepsilon) = \underbrace{\left[\prod_{s=-\ell(p)}^{r(p)} (\varepsilon + s)\right]}_{= (p+1)! f_{p+1}(\varepsilon)} \left[\sum_{j=-\ell(p)}^{r(p)} \left(\frac{1}{\varepsilon + j} \prod_{\substack{q=-\ell(p) \\ q \neq j}}^{r(p)} \frac{1}{q - j}\right) u_j\right]. \tag{4.74}$$

That is, if we compute the Lagrange interpolation weights using (4.74), then we need to calculate $f_{p+1}(\varepsilon)$. It should be noted that computing the interpolation weights using (4.74) is more efficient than the naive way resulting from evaluating (4.72) at $x = -h\varepsilon$; see [6, Sec. 3] for further details and related discussion.

4.3.2 Inexact departure points: An expensive strategy

We now move to the more practical use case in which the semi-Lagrangian methods do not exactly locate departure points. Incorporating this into the truncation estimates from Lemma 4.8 is not straightforward, so we consider a heuristic strategy instead.

It is instructive to think about the mechanics of the semi-Lagrangian discretization, recalling that it performs two operations in sequence. The first is to approximately locate departure points, and the second is to estimate the solution at the approximate departure points via polynomial interpolation. While the ideal coarse-grid operator is not a semi-Lagrangian discretization, it does represent *a* discretization of the PDE, and we can interpret it in a similar way to a semi-Lagrangian discretization. Considering the nonzeros in the stencil of the ideal coarse-grid operator for the constant-wave-speed problem in Figure 4.4, they are clearly centred about the departure point. See also Section 2.5.2 for similar arguments, albeit in the context of Eulerian discretizations. Numerical tests (not shown here for brevity) also confirm analogous behaviour occurs for variable-wave-speed problems. Therefore, we can interpret the ideal coarse-grid operator as roughly locating departure points and then carrying out an interpolation-like procedure at them. The coarse-grid operator (4.67) for $r = \infty$ mimics, to lowest order, the interpolation-like procedure of the ideal operator at coarse-grid departure points, but what can we say about how the ideal operator approximates departure points?

Suppose now that the fine-grid semi-Lagrangian operator locates departure points with a

numerical integration scheme having local accuracy $\mathcal{O}(\delta t^{r+1})$, and that it *exactly* interpolates to them. Then,

$$u(x_i, t_n + (k+1)\delta t) - \left(\mathcal{S}_{\infty,r}^{(t_n+k\delta t,\delta t)} u(\boldsymbol{x}, t_n + k\delta t)\right)_i = \mathcal{O}(\delta t^{r+1}), \quad k \in \{0, 1, \ldots, m-1\}. \tag{4.75}$$

Along the lines of reasoning used in the proof of Lemma 4.8 (i.e., repeatedly applying this approximation in sequence for $k = 0, \ldots, m-1$), it follows that

$$u(x_i, t_n + m\delta t) - \left(\prod_{k=0}^{m-1} \mathcal{S}_{\infty,r}^{(t_n+k\delta t,\delta t)} u(\boldsymbol{x}, t_n)\right)_i = \mathcal{O}(m\delta t^{r+1}). \tag{4.76}$$

That is, the ideal coarse-grid operator effectively locates departure points with an accuracy similar to that with which the fine-grid operator $\mathcal{S}_{\infty,r}^{(t_n+k\delta t,\delta t)}$ does, since it is only $\mathcal{O}(m)$ times larger. In contrast, a coarse-grid semi-Lagrangian operator using the same numerical integration scheme gives

$$u(x_i, t_n + m\delta t) - \left(\mathcal{S}_{\infty,r}^{(t_n,m\delta t)} u(\boldsymbol{x}, t_n)\right)_i = \mathcal{O}((m\delta t)^{r+1}). \tag{4.77}$$

The potential issue now is that it may be the case that $m\delta t^{r+1} \ll (m\delta t)^{r+1} = \mathcal{O}(1)$, particularly for $m \gg 1$. In other words, a coarse-grid semi-Lagrangian operator using the same numerical integrator as the fine-level operator may be wildly incorrect in its estimation of departure points, particularly for coarsening factors m of practical interest, where $m\delta t = \mathcal{O}(1)$. Schematic examples of this issue are shown in Figure 4.10, for a modest coarsening factor of $m = 8$.

The examples show that departure points are located accurately when integrating along characteristics with fine-grid-scale accuracy, but they are located highly inaccurately when the integration is performed with coarse-grid-scale accuracy. For the 1st-order method (left panel), some of the departure point approximations are wildly far away from their true values. For the 3rd-order method (right panel), the departure points are estimated much more accurately, but some characteristics have crossed as they approach the departure point, which clearly cannot happen. In fact, the time-step size of the numerical integrator should be limited by the condition that the characteristics do not cross [86].

For variable-wave-speed problems (4.58) and (4.59), we have conducted numerical experiments using the backward Euler coarse-grid operator (4.68), where rather than using exact numerical integration to locate departure points, they have been approximated. In general, we find that if the coarse-grid departure points are approximated using coarse-grid-scale accuracy (i.e., the red dashed lines in Figure 4.10), then MGRIT does not perform robustly, in the sense that it performs well for some problems, but very poorly on others.

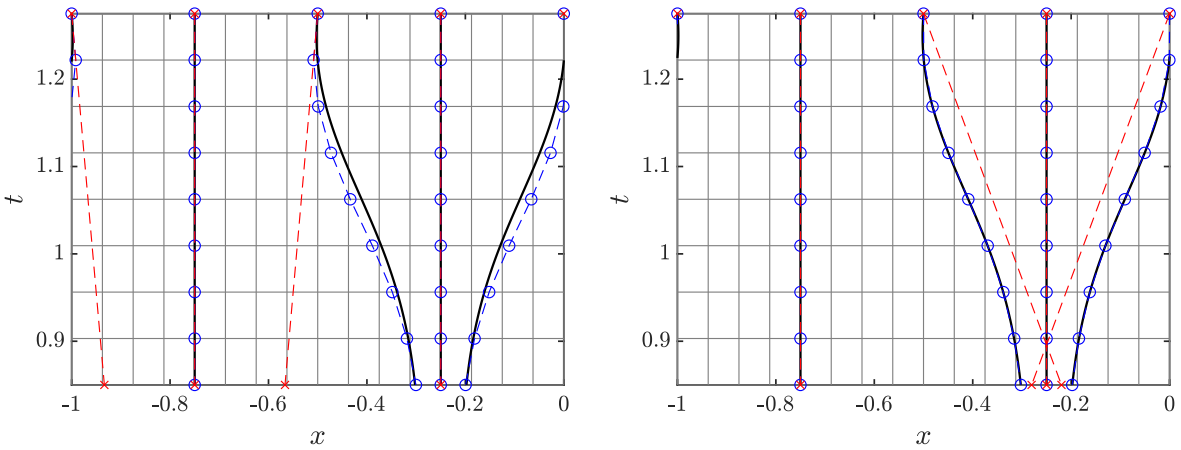

FIGURE 4.10: Every 4th coarse-grid characteristic on the left half of the discretized spatial domain Ω over a particular coarse-grid time interval for advection problem (4.53) with wave-speed given by (4.58). The fine-grid time step-size is $\delta t = 0.85 \times 2/(2^5) \approx 0.0531$, and the coarsening factor is $m = 8$. Solid black lines are exact characteristics. Blue circles represent integration backwards along characteristics using $m = 8$ ERK steps of size δt. Red crosses represent integration backwards along characteristics using a single ERK step of size $m\delta t$. **Left:** ERK method has accuracy $r = 1$. **Right:** ERK method has accuracy $r = 3$.

Two examples for which the solver performs poorly are shown in Figure 4.11.

In practice, our interpolation is of finite accuracy, of course, and therefore there is interaction between the interpolation and the numerical integration that takes place during each of the m fine-grid time steps that define the ideal coarse-grid operator. For this reason, we do not necessarily expect that an effective coarse-grid operator has to locate departure

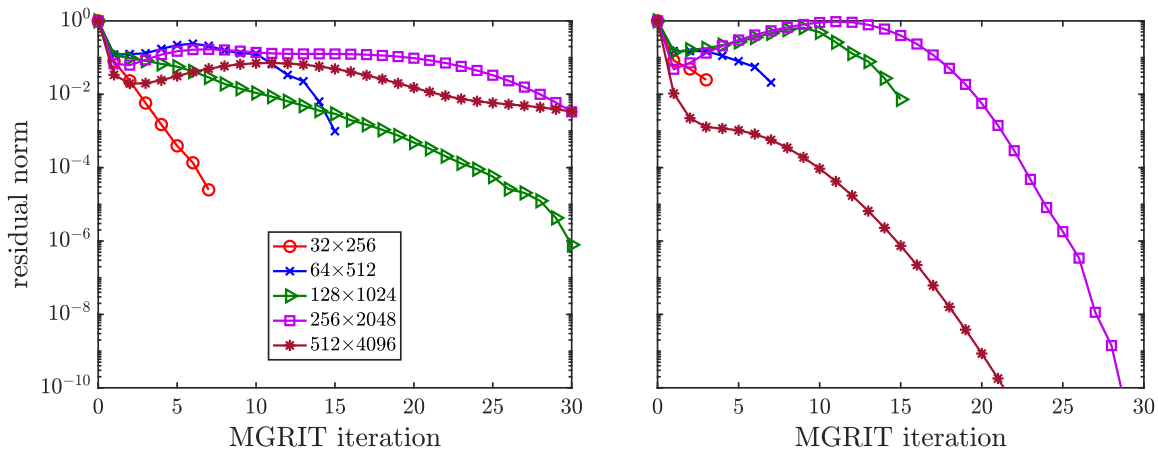

FIGURE 4.11: Two-norm of the space-time residual (relative to its initial value) as a function of MGRIT iteration. The residual is shown for several mesh resolutions $n_x \times n_t$, as indicated in the legend. The PDE is (4.53) with wave-speed given by (4.58). The fine-grid operator is $\mathcal{S}_{p,r}^{(t_n,\delta t)}$, and the coarse-grid operator is the backward Euler operator $\mathcal{B}_{p+1}^{(t_n,m\delta t)}\mathcal{S}_{p,r}^{(t_n,m\delta t)}$. Departure points on the coarse grid are located by using a single ERK step of size $m\delta t$. **Left:** The coarsening factor is $m = 16$, and the discretization orders are $p = r = 1$. **Right:** The coarsening factor is $m = 32$, and the discretization orders are $p = r = 3$.

points with an accuracy of $\mathcal{O}(m\delta t^{r+1})$, but given the results in Figure 4.11 it stands to reason that it needs to approximate them more accurately than rediscretization does, at least in general. Thus, for the inexact, semi-Lagrangian discretization $\mathcal{S}_{p,r}^{(t_n,\delta t)}$ of advection problem (4.53) we propose the following backward Euler coarse-grid operator

$$\Phi^{(t_n,m\delta t)} = \mathcal{B}_{p+1}^{(t_n,m\delta t)}\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}, \tag{4.78}$$

in which the r_* is used to signify that the coarse-grid semi-Lagrangian operator $\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}$ should locate departure points with an accuracy that is in some sense roughly comparable to that of the fine-grid operator $\mathcal{S}_{p,r}^{(t_n,\delta t)}$. The matrix $\mathcal{B}_{p+1}^{(t_n,m\delta t)}$ is still defined as it was in (4.68).

The question now is how should $\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}$ approximate coarse-grid departure points. One option is that it use the same numerical integrator as the fine-grid operator, but rather than taking a single large step of $m\delta t$, it should m smaller steps of size δt (i.e., the blue curves in Figure 4.10). This of course would make $\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}$ much more expensive than $\mathcal{S}_{p,r}^{(t_n,m\delta t)}$, because it is doing work at the fine-grid scale. However, $\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}$ would still be significantly less expensive than the ideal coarse-grid operator because it only does one set of interpolations per time step rather than m of them.

MGRIT iteration counts for numerical tests using this expensive coarse-grid operator are shown in Table 4.1. We test the solver on both variable wave-speeds (4.58) and (4.59), and we also test the $\alpha = 1$ case for comparison. In all tests, MGRIT converges in a number of iterations much less than $n_t/(2m)$, and the residuals monotonically decrease with the iteration (not shown for the sake of brevity). For a given problem and discretization, iteration counts seem roughly constant as the mesh is refined. There is small growth for some problems, but this is perhaps due to the solver not yet settling into an asymptotic convergence rate with respect to the mesh size. Interestingly, convergence seems to deteriorate as the discretization order is increased, which is the opposite of the trend for the optimized coarse-grid operators developed in Chapter 2 for Eulerian discretizations of the constant-wave-speed problem. This perhaps suggests that better coarse-grid operators exist than the one we propose here, particularly for the higher-order discretizations. Furthermore, for the 3rd- and 5th-order discretizations, the constant-wave-speed problem typically results in the highest iteration counts. In any event, the convergence reported in Table 4.1 is fast for the multigrid solution of hyperbolic problems.

4.3.3 Inexact departure points: A scalable strategy

Given that the backward Euler coarse-grid operator (4.78) leads to fast MGRIT convergence when using fine-grid-scale integration to approximate coarse-grid departure points

		$\alpha = 1$				$\alpha(x,t) = (4.58)$				$\alpha(x,t) = (4.59)$			
		m				m				m			
p, r	$n_x \times n_t$	4	8	16	32	4	8	16	32	4	8	16	32
$1, 1$	32×256	12	9	7	4	8	9	8	4	10	9	7	4
	128×1024	12	10	9	9	10	10	9	10	10	9	8	8
	512×4096	12	10	9	9	11	10	10	10	12	10	9	8
$3, 3$	32×256	18	12	8	4	11	12	8	4	11	9	7	4
	128×1024	20	15	13	10	12	14	12	12	13	11	10	9
	512×4096	20	15	13	12	14	13	14	14	17	14	12	10
$5, 5$	32×256	21	13	8	4	12	13	8	4	12	10	8	4
	128×1024	28	20	16	12	15	17	14	13	15	13	11	9
	512×4096	29	20	18	16	15	16	17	18	21	16	14	11

TABLE 4.1: Number of iterations for two-level MGRIT to reduce the space-time residual 2-norm by 10 orders of magnitude. The PDE is (4.53) with wave-speed $\alpha(x,t)$ indicated in the top row of the table. The fine-grid operator is $\mathcal{S}_{p,r}^{(t_n,\delta t)}$, and the coarse-grid operator is the backward Euler operator (4.78). The coarse-grid semi-Lagrangian operators $\mathcal{S}_{p,r_*}^{(t_n,\delta t)}$ estimate departure points by integrating with m steps of size δt along characteristics. For $n_t = 4096$, the number of MGRIT iterations $n_t/(2m)$ needed to reach the exact solution for $m = (4, 8, 16, 32)$ is $(512, 256, 128, 64)$.

(see Table 4.1), we now develop a less expensive way to approximate them.

Recall that in applying the coarse-grid semi-Lagrangian operator $\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}$, we need to compute the departure point at time $t = t_n$ of the coarse-grid characteristics $\xi_i^{(t_n,m\delta t)}(t)$ that arrive at $(x,t) = (x_i, t_n + m\delta t)$. When time-stepping across this interval on the fine grid with the operators $\mathcal{S}_{p,r}^{(t_n+k\delta t,\delta t)}$, $k \in \{0, 1, \ldots, m-1\}$, we map out the trajectories of the fine-grid characteristics $\xi_i^{(t_n+k\delta t,\delta t)}(t)$ over the intervals $t \in [t_n + k\delta t, t_n + (k+1)\delta t]$ (see the gold lines in Figure 4.12). In computing these fine-grid characteristics, we in effect map out the vector field that dictates the flow of any characteristic across the coarse space-time slab $(x,t) \in \Omega \times [t_n, t_n + m\delta t]$ with fine-grid-scale accuracy. The idea we propose now is to approximately propagate a coarse-grid characteristic through this space-time slab by recycling the fine-grid characteristics to guide its path, following the schematic shown in Figure 4.12.

For simplicity of notation, let us only consider the first coarse time interval $t \in [0, m\delta t]$. For shorthand, we denote fine-grid departure points on this interval by $f_i^{(k)} \equiv \xi_i^{(k\delta t,\delta t)}(k\delta t)$, $k \in \{0, 1, \ldots, m-1\}$. Furthermore, let $c_i^{(k)}$, $k \in \{0, 1, \ldots, m-1\}$, denote the x location at which the coarse-grid characteristic $\xi_i^{(0,m\delta t)}(t)$ lands at time $t = k\delta t$. That is, $c_i^{(k)} = \xi_i^{(0,m\delta t)}(k\delta t)$. Over the last fine-grid time-step interval $t \in [(m-1)\delta t, m\delta t]$, the coarse-grid characteristic is the same as the fine-grid characteristic, since they both arrive

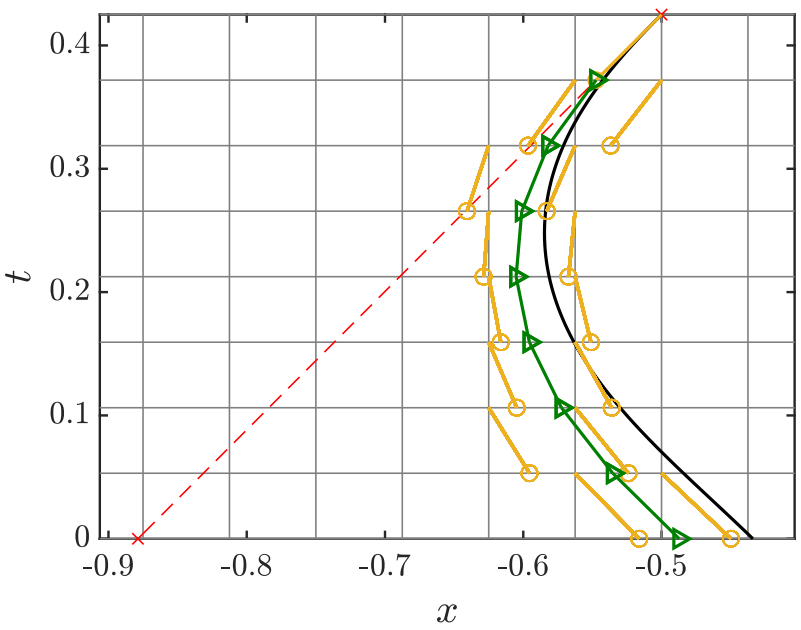

FIGURE 4.12: Evolution of a coarse-grid characteristic of advection problem (4.53) with wave-speed given by (4.58) using piecewise linear interpolation of fine-grid characteristics. Note that only a subset of the spatial domain $x \in \Omega = (-1, 1)$ is shown to better highlight the detail of the characteristic. The black curve is the exact coarse-grid characteristic. The green curve is the coarse-grid characteristic approximated with the interpolation strategy (i.e., the triangle marker at time $t = k\delta t$ is $c_i^{(k)}$ from (4.80)). The gold lines are the fine-grid characteristics that are the nearest neighbours of the approximate coarse-grid characteristic (i.e., the left and right circle markers at time $t = k\delta t$ are, respectively, $f_{j-1}^{(k)}$ and $f_j^{(k)}$ from (4.80)). These fine-grid characteristics were determined by a single ERK step of size δt. The red dashed line is the coarse-grid characteristic approximated by a single ERK step of size $m\delta t$. The fine-grid time step size is $\delta t = 0.85 \times 2/(2^5)$, and the coarsening factor is $m = 8$. The ERK method used to generate the fine- and coarse-grid characteristics has accuracy $r = 1$.

at $(x, t) = (x_i, m\delta t)$, and therefore they intersect the x axis at the same location,

$$c_i^{(m-1)} = f_i^{(m-1)}. \tag{4.79}$$

Using this as a final-time condition of sorts, the remaining intersection points of the coarse-grid characteristic can be estimated by carrying out the following update in sequence

$$c_i^{(k)} \approx \frac{f_j^{(k)} - f_{j-1}^{(k)}}{h} \left(c_i^{(k+1)} - x_j \right) + f_j^{(k)}, \quad \text{for } k = m-2, \ldots, 1, 0, \tag{4.80}$$

where j is such that x_j is the mesh point to the right of $c_i^{(k+1)}$. A schematic of this procedure is shown in Figure 4.12. Upon terminating the update (4.80), we have an approximation for $c_i^{(0)}$, which is the departure point of the coarse-grid characteristic $\xi_i^{(0,m\delta t)}(t)$.

The update formula (4.80) is based on nearest neighbour, piecewise linear interpolation to estimate $c_i^{(k)}$. Specifically, at some time $t = (k+1)\delta t$, we know the trajectories of the two

fine-grid characteristics over $t \in [k\delta t, (k+1)\delta t]$ that lie on either side of the take-off point of the characteristic whose trajectory we desire on this time interval (in Figure 4.12, the two gold characteristics on either side of the green characteristic). Therefore, to estimate the trajectory of the characteristic in the middle, we may fit a linear function to the process of mapping take-off points at $t = (k+1)\delta t$ into departure points at time $t = k\delta t$. That is, the x location of a characteristic at time $t = k\delta t$ that passes through the take-off point $(x, t) = (x_*, (k+1)\delta t)$ can be estimated as

$$\frac{f_j^{(k)} - f_{j-1}^{(k)}}{h}(x_* - x_j) + f_j^{(k)}, \quad x_* \in [x_{j-1}, x_j]. \tag{4.81}$$

This is equivalent to the formula (4.80), in which x^* is $c_i^{(k+1)}$, the take-off point of the coarse-grid characteristic at time $t = (k+1)\delta t$.

The illustration in Figure 4.12 shows how this strategy has the potential to approximate coarse-grid departure points much more accurately than a single coarse-grid step of the ERK scheme used to integrate fine-grid characteristics.

To test the effectiveness of this strategy, we run the same set of tests as were used to generate the results of Table 4.1, where m ERK steps of size δt were used. We do not report the iteration counts here, however, because we find that they are identical to those in Table 4.1, with the exception of four entries, which differed only by one iteration. These results suggest that the linear interpolation strategy is just as effective for locating coarse-grid departure points as taking m ERK steps of size δt. It should also be noted that if the wave-speed is spatially independent, $\alpha(x, t) \equiv \alpha(t)$, then it is possible to show that the linear interpolation strategy yields the same estimates for departure point as taking m steps of size δt with the fine-grid ERK scheme.

Recall that the motivation for this section was to estimate coarse-grid departure points in a way that is less expensive than taking m ERK steps of size δt. However, since the linear interpolation strategy we have proposed requires taking $m - 1$ steps, it cannot be significantly cheaper than using m steps of an ERK scheme.[10] Supposing the linear interpolation strategy yields sufficiently accurate departure points, in the sense that it does not strongly deteriorate MGRIT convergence, then it does have the significant advantage over stepping at the fine-grid-scale with an ERK scheme that it is scalable to multiple levels. That is, say, for example, we have a three-level method in which we coarsen by m on each level. The linear interpolation strategy takes $\mathcal{O}(m)$ work to estimate a departure points on the first coarse level, but if it is then applied recursively on the coarsest level, it

[10] Whether it is cheaper or not depends on the number of stages of the ERK scheme and the cost of evaluating the wave-speed. Recall that an s-stage ERK scheme requires s evaluations of the wave-speed per time-step. The linear interpolation strategy requires no evaluations of the wave-speed. In any event, the number of FLOPs for either strategy scales as $\mathcal{O}(m)$.

requires only $\mathcal{O}(m)$ work there to estimate departure points. In general, if the strategy is applied recursively throughout a multilevel solver, it requires only $\mathcal{O}(m)$ work to estimate departure points, independent of the level they occur on.

In stark contrast, if an ERK method is to be used to estimate departure points on coarse levels, by our previous arguments regarding the inaccuracy of taking large time-steps, it must do so by taking many small time-steps on the size of the fine-grid δt. That is, using an ERK method to estimate coarse-grid departure points on a coarse level $\ell > 0$ requires $\mathcal{O}(m^\ell)$ work. This is clearly not a scalable strategy, recalling that in a multigrid algorithm we typically want to coarsen down to a constant number of points, independent of the fine-grid problem size. In Section 4.4.1 we extend our coarse-grid operator to the multilevel setting, and we compare these two approaches of locating coarse-grid departure points.

Finally, it is worth remarking that the linear interpolation characteristic tracking strategy proposed here is expensive from a memory perspective. Estimating departure points on a coarse level requires storing all departure points on the level above it. Future research will involve developing strategies that are less memory intensive.

4.4 Extensions in one spatial dimension

This section considers two extensions of the coarse-grid operator: Section 4.4.1 extends it to the multilevel setting, and Section 4.4.2 shows how it may be applied to solve advection-diffusion problems.

4.4.1 The multilevel setting

In this section, we generalize the two-level, backward Euler coarse-grid operator (4.78) from the previous section so that it can be applied within a multilevel algorithm. Let $\ell \in \mathbb{N}_0$ be the level index, and assume that the time-step size on level ℓ is $m^\ell \delta t$. For notational simplicity, it is useful to introduce the following shorthand for the function in (4.70),

$$\boldsymbol{\varphi}_{p+1}^{(t_n, m^\ell \delta t)} \equiv \varphi_{p+1}\Big(\boldsymbol{\varepsilon}^{(t_n, m^{\ell-1}\delta t)}, \ldots, \boldsymbol{\varepsilon}^{(t_n+(m-1)m^{\ell-1}\delta t, m^{\ell-1}\delta t)}, \boldsymbol{\varepsilon}^{(t_n, m^\ell \delta t)}\Big). \tag{4.82}$$

That is, $\boldsymbol{\varphi}_{p+1}^{(t_n, m^\ell \delta t)}$ approximates the coefficient vector appearing in the leading-order term of the difference $\mathcal{S}_{p,\infty}^{(t_n, m^\ell \delta t)} - \prod_{k=0}^{m-1} \mathcal{S}_{p,\infty}^{(t_n+km^{\ell-1}\delta t, m^{\ell-1}\delta t)}$. Or, in other words, given the level $\ell - 1$ semi-Lagrangian operators $\mathcal{S}_{p,\infty}^{(t_n+km^{\ell-1}\delta t, m^{\ell-1}\delta t)}$, $k \in \{0, \ldots, m-1\}$, (4.82) approximates the coefficient vector of the leading-order term in the difference between

the *ideal* level ℓ operator $\prod_{k=0}^{m-1} \mathcal{S}_{p,\infty}^{(t_n+km^{\ell-1}\delta t, m^{\ell-1}\delta t)}$ and the *rediscretized* level ℓ operator $\mathcal{S}_{p,\infty}^{(t_n,m^{\ell}\delta t)}$.

To develop a truly multilevel operator based on the backward Euler operator (4.78), it is first instructive to consider a three-level algorithm. Given the level $\ell = 1$ backward Euler operators $\mathcal{B}_{p+1}^{(t_n+km\delta t,m\delta t)}\mathcal{S}_{p,\infty}^{(t_n+km\delta t,m\delta t)}$ for $k \in \{0,\ldots,m-1\}$, consider the associated ideal level $\ell = 2$ operator and the following sequence of approximations to it,

$$\Phi_{\text{ideal}}^{(t_n,m^2\delta t)} = \prod_{k=0}^{m-1} \Phi^{(t_n+km\delta t,m\delta t)}, \tag{4.83}$$

$$= \prod_{k=0}^{m-1} \left(\mathcal{B}_{p+1}^{(t_n+km\delta t,m\delta t)}\mathcal{S}_{p,r_*}^{(t_n+km\delta t,m\delta t)}\right), \tag{4.84}$$

$$\approx \left(\prod_{k=0}^{m-1} \mathcal{B}_{p+1}^{(t_n+km\delta t,m\delta t)}\right)\left(\prod_{k=0}^{m-1} \mathcal{S}_{p,r_*}^{(t_n+km\delta t,m\delta t)}\right), \tag{4.85}$$

$$\approx \left(\prod_{k=0}^{m-1} \mathcal{B}_{p+1}^{(t_n+km\delta t,m\delta t)}\right)\left(\mathcal{B}_{p+1}^{(t_n,m^2\delta t)}\mathcal{S}_{p,r_*}^{(t_n,m^2\delta t)}\right), \tag{4.86}$$

$$= \left(\prod_{k=0}^{m-1} \left[I - \operatorname{diag}\left(\boldsymbol{\varphi}_{p+1}^{(t_n+km\delta t,m\delta t)}\right)\mathcal{D}_{p+1}\right]^{-1}\right)\left[I - \operatorname{diag}\left(\boldsymbol{\varphi}_{p+1}^{(t_n,m^2\delta t)}\right)\mathcal{D}_{p+1}\right]^{-1}\mathcal{S}_{p,r_*}^{(t_n,m^2\delta t)}, \tag{4.87}$$

$$\approx \left[I - \operatorname{diag}\left(\sum_{k=0}^{m-1} \boldsymbol{\varphi}_{p+1}^{(t_n+km\delta t,m\delta t)}\right)\mathcal{D}_{p+1}\right]^{-1}\left[I - \operatorname{diag}\left(\boldsymbol{\varphi}_{p+1}^{(t_n,m^2\delta t)}\mathcal{D}_{p+1}\right)\right]^{-1}\mathcal{S}_{p,r_*}^{(t_n,m^2\delta t)}, \tag{4.88}$$

$$\approx \left[I - \operatorname{diag}\left(\sum_{k=0}^{m-1} \boldsymbol{\varphi}_{p+1}^{(t_n+km\delta t,m\delta t)} + \boldsymbol{\varphi}_{p+1}^{(t_n,m^2\delta t)}\right)\mathcal{D}_{p+1}\right]^{-1}\mathcal{S}_{p,r_*}^{(t_n,m^2\delta t)}. \tag{4.89}$$

The approximation in (4.85) is that the backward Euler and semi-Lagrangian operators commute. For general wave-speeds these operators do not commute, but note that when the wave-speed is independent of space they do commute (the diagonal matrices built from the $\boldsymbol{\varphi}_{p+1}$ vectors are constant, and $\mathcal{D}_{p+1}$ and the semi-Lagrangian operators are circulant).

The approximation in (4.86) is that the m successive semi-Lagrangian steps of size $m\delta t$ have been approximated using our existing two-level approximation. That is, the m steps are replaced by a single semi-Lagrangian step of $m^2\delta t$ followed by a backward Euler step that approximately corrects for the lowest-order difference between their truncation errors.

The approximation in (4.88) arises from placing the m backward Euler factors inside the inverse, and then keeping only the lowest-order terms in their product, recalling $\mathcal{D}_{p+1} = h^{p+1}\operatorname{diag}\left(\frac{\partial^{p+1}}{\partial x^{p+1}}\right)$+higher-order terms. This approximation can be understood

as a Taylor series interpretation of the standard rediscretization approach typically employed for backward Euler operators, in which they are rediscretized/reapplied on the coarse level with a time-step size that is m times larger.

The final approximation of (4.89) arises from placing both backward Euler matrices under the inverse, and truncating the highest-order term from their product, which is proportional to $\mathcal{D}_{p+1}\mathcal{D}_{p+1} = h^{2(p+1)}\,\mathrm{diag}\left(\frac{\partial^{2(p+1)}}{\partial x^{2(p+1)}}\right)$+higher-order terms.

Notice that the final approximation (4.89) has the same structure as the level $\ell = 1$ operator (4.78) proposed for the two-level algorithm, in that it is a semi-Lagrangian operator followed by a backward Euler correction. Based on this, we propose the following level $\ell > 0$ time-stepping operators for evolving solutions from $t_n \to t_n + m^\ell \delta t$

$$\Phi^{(t_n, m^\ell \delta t)} = \left[I - \mathrm{diag}\left(\boldsymbol{\nu}_{p+1}^{(t_n, m^\ell \delta t)}\right)\mathcal{D}_{p+1}\right]^{-1} \mathcal{S}_{p,r_*}^{(t_n, m^\ell \delta t)}, \quad \ell \in \mathbb{N}, \tag{4.90}$$

in which the vector of dissipation coefficients is defined recursively by

$$\boldsymbol{\nu}_{p+1}^{(t_n, m^\ell \delta t)} = \begin{cases} \boldsymbol{\varphi}_{p+1}^{(t_n, m^\ell \delta t)}, & \ell = 1, \\ \displaystyle\sum_{k=0}^{m-1} \boldsymbol{\nu}_{p+1}^{(t_n + k m^{\ell-1}\delta t, m^{\ell-1}\delta t)} + \boldsymbol{\varphi}_{p+1}^{(t_n, m^\ell \delta t)}, & \ell > 1. \end{cases} \tag{4.91}$$

An interesting question is how one should interpret the coarse-grid operator (4.90). Fortunately, some insight may be gained by examining it in the event that the wave-speed is constant.

Lemma 4.10 (Constant-coefficient multilevel operator)**.** *Suppose that the wave-speed α is constant, such that the mesh-normalized distance from departure points on level ℓ to their east neighbouring mesh points is the constant $\varepsilon^{(m^\ell \delta t)}$. Then, the coarse-grid operator* (4.90) *is*

$$\Phi^{(m^\ell \delta t)} = \left[I - \left(f_{p+1}\left(\varepsilon^{(m^\ell \delta t)}\right) - m^\ell f_{p+1}\left(\varepsilon^{(\delta t)}\right)\right)\mathcal{D}_{p+1}\right]^{-1} \mathcal{S}_p^{(m^\ell \delta t)}, \quad \ell \in \mathbb{N}. \tag{4.92}$$

In other words, the coarse-grid operator (4.92) *on level $\ell \in \mathbb{N}$ is the same as the two-level backward Euler operator for the constant-coefficient problem defined by* (4.22) *if a coarsening factor of m^ℓ is used to coarsen from the fine level to the coarse level rather than m.*

Proof. See Appendix B.5. ■

An immediate corollary of this result and Lemma 4.6—which stated the backward Euler coarse-grid operator is unconditionally stable with respect to all problem parameters,

including the coarsening factor—is that for the constant-wave-speed advection problem, the coarse-grid operator (4.90) is unconditionally stable on any level $\ell \in \mathbb{N}$.

We now present results of our numerical tests using a multilevel solver that employs the coarse-grid operator (4.90). In these tests, we use a constant coarsening factor of m on all levels, and continue to coarsen until doing so would result in fewer than two points in time. To locate departure points on coarse levels, the linear interpolation strategy of Section 4.3.3 is employed recursively. MGRIT V-cycle iteration counts are given in Table 4.2. The iteration counts are very close to those in Table 4.1 for the two-level solvers that used m steps of the fine-grid ERK scheme to locate coarse-grid departure points. In fact, almost all iteration counts are identical, with several problems requiring just two more iterations in the multilevel case, and a single problem requiring three more iterations. That iteration counts are effectively the same in the two-level and multilevel cases means that the multilevel operator (4.90) performs well, and indicates that the coarse-grid operator leads to a scalable multilevel solver. Note that it is not uncommon to see multigrid iterations strongly increase for hyperbolic problems when transitioning from two to many levels (see, e.g., [58, 108, 109, 50]). These promising results also indicate that our linear interpolation strategy for approximating coarse-grid departure points does so with sufficient accuracy, even on much coarser levels. Recall that it approximates departure

		$\alpha = 1$				$\alpha(x,t) = (4.58)$				$\alpha(x,t) = (4.59)$			
		m				m				m			
p, r	$n_x \times n_t$	4	8	16	32	4	8	16	32	4	8	16	32
$1, 1$	32×256	12	9	7	4	9	9	8	4	10	9	7	4
	128×1024	13	10	9	9	10	10	9	10	10	9	8	8
	512×4096	13	10	9	9	11	10	10	10	12	10	9	8
$3, 3$	32×256	18	12	8	4	11	12	8	4	12	10	7	4
	128×1024	21	15	13	10	14	14	12	12	14	12	10	9
	512×4096	21	15	13	12	14	13	14	14	17	14	12	10
$5, 5$	32×256	21	13	8	4	12	13	8	4	13	10	8	4
	128×1024	29	20	16	12	16	17	14	13	16	13	11	9
	512×4096	32	21	18	16	16	16	17	18	21	17	14	11

TABLE 4.2: Number of MGRIT V-cycles to reduce the space-time residual 2-norm by 10 orders of magnitude. The PDE is (4.53) with wave-speed $\alpha(x,t)$ indicated in the top row of the table. The fine-grid operator is $\mathcal{S}_{p,r}^{(t_n,\delta t)}$, and the coarse-grid operator is the dissipatively corrected operator (4.90). Departure points on coarse levels are located by recursively applying the linear interpolation strategy of Section 4.3.3. A coarsening factor of m is used on all levels, and the mesh is coarsened until doing so would result in fewer than two points in time. At the largest resolution of $n_t = 4096$, this yields solvers with $6, 4, 3$, and 3 levels for $m = 4, 8, 16$, and 32, respectively.

points with $\mathcal{O}(m)$ work, independent of level. This justifies our previous assertion from Section 4.3.2 that an effective coarse-grid operator likely does not need to highly accurately locate departure points.

We now also take this opportunity to explore approximating the action of the matrix inverse that appears in the coarse-grid operator (4.90). Up until this point in our numerical tests, we have been exactly applying this inverse by using a direct linear solver. Recall that in Section 4.2.4, for the constant-wave-speed problem, we bounded the condition number of this matrix, and showed that it scales as $\mathcal{O}(m)$ (see Corollary 4.7). We also argued that it should be well-approximated by a sparse matrix. Here we test these ideas by using an iterative solver rather than a direct solver for the linear systems that arise when applying the operator. The test problems are taken to be those of largest mesh resolution shown in Table 4.2. In terms of iterative solvers, we have tested stationary linear iterations such as Jacobi, and Gauss–Seidel. We have also considered polynomial approximation methods such as Chebyshev iteration, and GMRES. Out of all of these, GMRES seems to be by far the most efficient. Iteration counts are shown in Table 4.3 when

		$\alpha = 1$				$\alpha(x,t) = (4.58)$				$\alpha(x,t) = (4.59)$			
		m				m				m			
p, r	k	4	8	16	32	4	8	16	32	4	8	16	32
$1, 1$	0	✗	✗	82	47	✗	✗	82	47	✗	✗	72	41
	2	16	16	10	20	20	17	18	23	20	19	21	22
	4	13	10	9	9	11	11	10	10	12	10	9	9
	∞	13	10	9	9	11	10	10	10	12	10	9	8
$3, 3$	0	✗	✗	84	48	✗	✗	80	46	✗	✗	66	37
	2	21	17	14	18	19	17	15	18	19	17	18	19
	4	21	15	13	12	14	13	14	14	17	14	12	10
	∞	21	15	13	12	14	13	14	14	17	14	12	10
$5, 5$	0	✗	✗	86	49	✗	✗	76	44	✗	✗	63	36
	2	31	22	19	18	21	20	19	20	22	18	17	18
	4	32	21	18	16	16	16	17	18	21	17	14	11
	∞	32	21	18	16	16	16	17	18	21	17	14	11

TABLE 4.3: Number of MGRIT V-cycles to reduce the space-time residual 2-norm by 10 orders of magnitude. The problem sizes are all $n_x \times n_t = 512 \times 4096$. The linear solve within the coarse-grid operator (4.90) is approximated with k iterations of GMRES; $k = \infty$ is used to denote a direct solve. Otherwise, the problem set up is the same as it was in Table 4.2. An '✗' denotes a solve that did not converge within 100 iterations.

a fixed number of GMRES iterations is used on each linear system.[11] Remarkably, on all of the test problems, only ~ 4 GMRES iterations are needed before the number of MGRIT iterations is the same as if a direct solver is used (the $k = \infty$ row). Better understanding the rapid convergence of GMRES for this problem will be a subject of future research. In summary, we have demonstrated that our coarse-grid operator yields fast convergence on high-order discretizations for complicated advection problems, and that the expensive components of the operator such as inverting the backward Euler matrix and tracking coarse-grid characteristics may be approximated effectively with very little cost.

4.4.2 Application to an advection-diffusion problem

In this section, we demonstrate with a simple example how our coarse-grid operator is applicable to a broader class of problems than simply pure advection equations.

As a specific example, let us consider the following one-dimensional advection-diffusion equation

$$\frac{\partial u}{\partial t} + \alpha(t)\frac{\partial u}{\partial x} = \beta\frac{\partial^2 u}{\partial x^2}, \tag{4.93}$$

subject to periodic boundary conditions in space, and with constant diffusivity $\beta \geq 0$. On the finest level, we will discretize this problem in a mixed fashion, with a semi-Lagrangian method to treat the advection term, and a standard one-step method-of-lines approach for the diffusion term.

The Lagrangian formulation of (4.93) is

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = \alpha(t), \tag{4.94}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t), t) = \beta\frac{\partial^2}{\partial x^2}u(\xi(t), t). \tag{4.95}$$

Since the wave-speed $\alpha(t)$ is independent of x, characteristics are equispaced at any given time t, which essentially means that our discretization of (4.94) and (4.95) can be written in the split form

$$\boldsymbol{u}_{n+1} = \mathcal{M}^{(t_n,\delta t)}\mathcal{S}_{p,r}^{(t_n,\delta t)}\boldsymbol{u}_n \equiv \Phi^{(t_n,\delta t)}\boldsymbol{u}_n, \tag{4.96}$$

[11] For $k = 0$ (i.e., no GMRES iterations), the dissipative correction from the coarse-grid operator (4.90) is not applied, so that the coarse-grid operator consists of just a semi-Lagrangian operator. That is, this represents rediscretization on coarse levels. In all the $k = 0$ cases, the solver is divergent (it does not converge in significantly fewer than $n_t/(2m)$ iterations), and the space-time residual grows exponentially before it begins to decay as the iteration count gets closer to $n_t/(2m)$.

in which $\mathcal{M}^{(t_n,\delta t)}$ is the one-step operator associated with a method-of-lines discretization of the diffusion equation $\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2}$, and the semi-Lagrangian method $\mathcal{S}_{p,r}^{(t_n \delta t)}$ is used to interpolate the solution at departure points at time t_n. For definiteness, in our numerical examples, the method-of-lines operator will consist of a finite-difference discretization for the diffusion term and an SDIRK method in time (see Appendix A.1 for Butcher tableaux).

The ideal coarse-grid operator stepping from $t_n \to t_n + m\delta t$ based on (4.96) is then

$$\Phi_{\text{ideal}}^{(t_n,m\delta t)} = \prod_{k=0}^{m-1} \left(\mathcal{M}^{(t_{n+k},\delta t)} \mathcal{S}_{p,r}^{(t_{n+k},\delta t)} \right) \approx \left(\prod_{k=0}^{m-1} \mathcal{M}^{(t_{n+k},\delta t)} \right) \left(\prod_{k=0}^{m-1} \mathcal{S}_{p,r}^{(t_{n+k},\delta t)} \right). \tag{4.97}$$

The approximation introduced here is that the semi-Lagrangian and method-of-lines discretizations commute; note that in the special case of periodic boundary conditions, and spatially independent wave-speed, these two operators do commute since all of the underlying operators involved are circulant. Therefore, the ideal coarse-grid operator (4.97) is the product of the ideal coarse-grid operators associated with the individual discretizations. Based on this, we propose the following coarse-grid operator to approximate (4.97)

$$\Phi^{(t_n,m\delta t)} = \mathcal{M}^{(t_n,m\delta t)} \mathcal{B}_{p+1}^{(t_n,m\delta t)} \mathcal{S}_{p,r_*}^{(t_n,m\delta t)}. \tag{4.98}$$

That is, for the method-of-lines discretization, we simply rediscretize the problem with the larger time step, since this typically works very well for such problems [28, 32, 33, 35]. To approximate the ideal semi-Lagrangian operator, we employ the dissipatively corrected backward Euler coarse-grid operator (4.78). In a multilevel method, the same form of coarse-grid operator should be applied but with the two-level operator (4.78) replaced with its multilevel generalization (4.90).

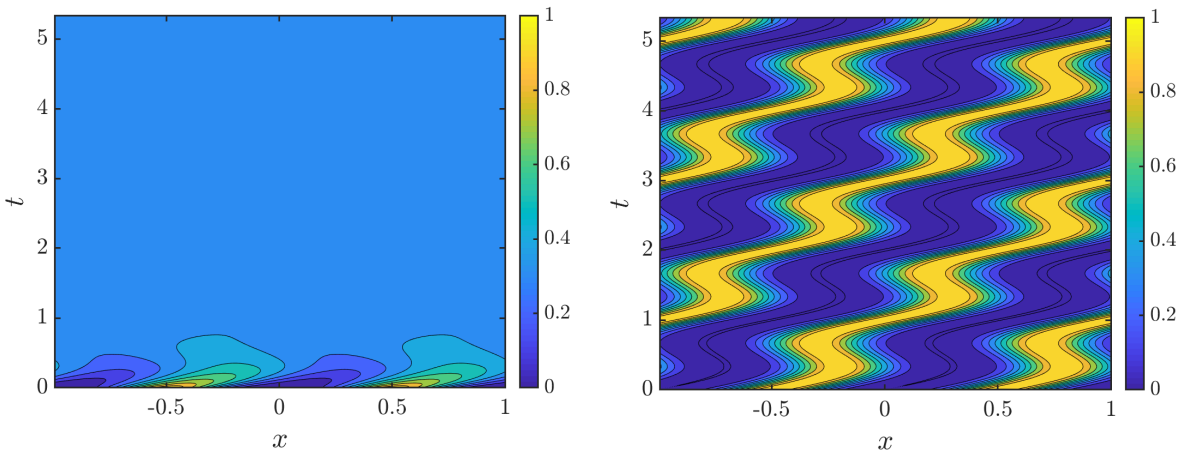

FIGURE 4.13: Numerical solution of advection-diffusion problem (4.93) for parameters given in (4.99), and when discretized with $\Phi^{(t_n,\delta t)} = \mathcal{M}_{2,2}^{(t_n,\delta t)} \mathcal{S}_{3,3}^{(t_n,\delta t)}$. **Left:** $\beta = 10^{-1}$. **Right:** $\beta = 10^{-5}$. The plots shown here correspond to the $n_x \times n_t = 256 \times 1024$ MGRIT solves shown in Table 4.4.

Now we present numerical test results using this coarse-grid operator. As a specific example, we consider the numerical solution of (4.93) with

$$\alpha(t) = \frac{1}{2} + \cos(2\pi t), \quad (x,t) \in (-1,1) \times (0,T), \quad u(x,0) = \sin^4(\pi x), \tag{4.99}$$

for several values of diffusivity $\beta \geq 0$. Plots of the numerical solution are shown in Figure 4.13, and MGRIT iteration counts are given in Table 4.4. We use multilevel V-cycles with a coarsening factor of $m = 8$ on each level. Ensuring that we do not coarsen to fewer than two points in time, the two smaller resolution solves ($n_x \times n_t = 64 \times 256, 128 \times 512$) use three levels, and the largest uses four levels ($n_x \times n_t = 256 \times 1024$). On coarse levels, departure points are estimated via the linear interpolation strategy from Section 4.3.3.

Iteration counts are shown for both the rediscretized operator, and the dissipatively corrected backward Euler operator (4.98). Iteration counts are also shown for two different discretizations, one with lower order and one with higher order. In all cases, the dissipatively corrected coarse-grid operator is at least as fast as rediscretization. As β decreases (i.e., the problem becomes more advective), there is only a small increase in the number of iterations for the dissipatively corrected operator, while those for rediscretization increase substantially, and they are not bounded significantly below $n_t/(2m)$. Furthermore,

		$\mathcal{M}_{2,1}^{(t_n,m\delta t)}\mathcal{S}_{1,1_*}^{(t_n,m\delta t)}$			$\mathcal{M}_{2,1}^{(t_n,m\delta t)}\mathcal{B}_{2}^{(t_n,m\delta t)}\mathcal{S}_{1,1_*}^{(t_n,m\delta t)}$		
	$n_x \times n_t$	64×256	128×512	256×1024	64×256	128×512	256×1024
β	10^{-1}	9	8	8	8	8	8
	10^{-3}	14	18	19	8	7	7
	10^{-5}	15	23	37	9	9	9
	0	15	23	37	9	9	9

		$\mathcal{M}_{2,2}^{(t_n,m\delta t)}\mathcal{S}_{3,3_*}^{(t_n,m\delta t)}$			$\mathcal{M}_{2,2}^{(t_n,m\delta t)}\mathcal{B}_{4}^{(t_n,m\delta t)}\mathcal{S}_{3,3_*}^{(t_n,m\delta t)}$		
	$n_x \times n_t$	64×256	128×512	256×1024	64×256	128×512	256×1024
β	10^{-1}	5	5	5	5	5	5
	10^{-3}	12	9	6	7	6	5
	10^{-5}	15	23	37	10	11	11
	0	15	23	38	10	11	11

TABLE 4.4: Number of MGRIT V-cycles to reduce the two-norm of the space-time residual by 10 orders of magnitude for advection-diffusion problem (4.93) with parameters given in (4.99). A coarsening factor of $m = 8$ is used on each level, and coarsening is performed until fewer than two points in time would remain. The fine-grid time step is $\delta t = \frac{2}{3}h$. The **left**-side of the table uses rediscretization on coarse levels, and the **right** uses the dissipatively corrected backward Euler coarse-grid operator (4.98). The **top** half of the table uses 2nd-order finite differences and SDIRK1 for the diffusion discretization, and a 1st-order semi-Lagrangian method for the advection discretization. The **bottom** half of the table uses 2nd-order finite differences and SDIRK2 for the diffusion discretization, and a 3rd-order semi-Lagrangian method for the advection discretization. Note that for $n_t = (256, 512, 1024)$, $n_t/(2m) = (16, 32, 64)$.

iteration counts for the dissipatively corrected operator are scalable with respect to mesh size for all values of β, while those for rediscretization are scalable with respect to mesh size only for the smallest values of β, the most diffusive problems. These results show that the potential applications for our coarse-grid operator extend beyond pure advection problems.

4.5 Two spatial dimensions

In this section, we demonstrate how our coarse-grid operator developed for one-dimensional problems can be extended to two dimensions. The two-dimensional semi-Lagrangian discretization is described in Section 4.5.1. The coarse-grid operator is developed in Section 4.5.2, and numerical tests are presented in Section 4.5.3.

4.5.1 The semi-Lagrangian discretization

Consider the two-dimensional, variable-wave-speed advection problem

$$\frac{\partial u}{\partial t} + \alpha(x,y,t)\frac{\partial u}{\partial x} + \beta(x,y,t)\frac{\partial u}{\partial y} = 0, \quad (x,y) \in \Omega, \tag{4.100}$$

again subject to periodic boundary conditions in space. We define a discrete mesh on Ω as the tensor product of one-dimensional meshes in the x- and y-directions, respectively. For simplicity, we assume the mesh is composed of n_x points with spacing of h in both directions, for a total of n_x^2 points.

Let us define a characteristic of (4.100) as a curve in x-y-t space parameterized by $(x,y,t) = (\xi(t), \eta(t), t)$. Then, the Lagrangian formulation of (4.100) is

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = \alpha(\xi(t), \eta(t), t), \tag{4.101}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(t) = \beta(\xi(t), \eta(t), t), \tag{4.102}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t), \eta(t), t) = 0. \tag{4.103}$$

The coupled equations (4.101) and (4.102) describe the time evolution of the x- and y-components of a characteristic curve, respectively. Equation (4.103) describes the time evolution of the solution of (4.100) along the characteristic $(x,y) = (\xi(t), \eta(t))$.

The semi-Lagrangian discretization of the Lagrangian equations (4.101), (4.102), and (4.103) works much the same as the discretization for their one-dimensional analogues. That is, departure points are first located by integrating backwards along characteristics,

and the solution is then approximated at each departure point by an interpolating polynomial fit through the nearest neighbouring mesh points. The approximation at the new time is then simply equal to the interpolated value, since by the evolution equation (4.103) the PDE solution remains constant along any given characteristic.

Let $\big(\xi_{ij}^{(t_n,\delta t)}(t), \eta_{ij}^{(t_n,\delta t)}(t)\big)$ be the characteristic that passes through the arrival point $(x, y, t) = (x_i, y_j, t_n + \delta t)$, which is a point on the mesh. Then, we seek the departure point/foot of this characteristic at time $t = t_n$, which is $(x, y, t) = \big(\xi_{ij}^{(t_n,\delta t)}(t_n), \eta_{ij}^{(t_n,\delta t)}(t_n), t_n\big)$. This departure point is located by solving the following final-time problem over $t \in [t_n, t_n + \delta t)$

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_{ij}^{(t_n,\delta t)}(t) = \alpha\big(\xi_{ij}^{(t_n,\delta t)}(t), \eta_{ij}^{(t_n,\delta t)}(t), t\big), \quad \xi_{ij}^{(t_n,\delta t)}(t_n + \delta t) = x_i, \tag{4.104}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta_{ij}^{(t_n,\delta t)}(t) = \beta\big(\xi_{ij}^{(t_n,\delta t)}(t), \eta_{ij}^{(t_n,\delta t)}(t), t\big), \quad \eta_{ij}^{(t_n,\delta t)}(t_n + \delta t) = y_j. \tag{4.105}$$

Upon locating the departure point by solving this two-dimensional system of ODEs, the solution is estimated at it by evaluating a two-dimensional interpolating polynomial fit through the nearest mesh points. Combining these two steps results in a fully discrete scheme of the form

$$\boldsymbol{u}_{n+1} = \mathcal{S}_{p,r}^{(t_n,\delta t)}\boldsymbol{u}_n. \tag{4.106}$$

Now let us consider the details of the interpolation, since they will be integral for developing our proposed coarse-grid operator, just as they were in the one-dimensional case. As for the one-dimensional case, we decompose the departure point into a neighbouring mesh point and its distance to this point. Specifically, let $(x, y) = \big(x_{ij}^{(t_n,\delta t)}, y_{ij}^{(t_n,\delta t)}\big)$ be the mesh point immediately to the north-east of the departure point $(x, y) = \big(\xi_{ij}^{(t_n,\delta t)}(t_n), \eta_{ij}^{(t_n,\delta t)}(t_n)\big)$. Then, we decompose, the x- and y-components of the departure point as, respectively,

$$\xi_{ij}^{(t_n,\delta t)}(t_n) \equiv x_{ij}^{(t_n,\delta t)} - h\varepsilon_{ij}^{(t_n,\delta t)}, \quad \varepsilon_{ij}^{(t_n,\delta t)} \in [0, 1), \tag{4.107}$$

$$\eta_{ij}^{(t_n,\delta t)}(t_n) \equiv y_{ij}^{(t_n,\delta t)} - h\nu_{ij}^{(t_n,\delta t)}, \quad \nu_{ij}^{(t_n,\delta t)} \in [0, 1), \tag{4.108}$$

where $\varepsilon_{ij}^{(t_n,\delta t)}$ is the (mesh-normalized) distance in the x-direction from the departure point to $x_{ij}^{(t_n,\delta t)}$. Analogously, $\nu_{ij}^{(t_n,\delta t)}$ is the (mesh-normalized) distance in the y-direction from the departure point to $y_{ij}^{(t_n,\delta t)}$; see Figure 4.14 for a schematic example.

The two-dimensional interpolation polynomial is constructed through a tensor product of a one-dimensional interpolation in the x-direction and a one-dimensional interpolation in the y-direction; see [31, pp. 61–62] for further details. Each one-dimensional interpolation uses $p + 1$ points, such that the two-dimensional interpolation uses $(p + 1)^2$ points.

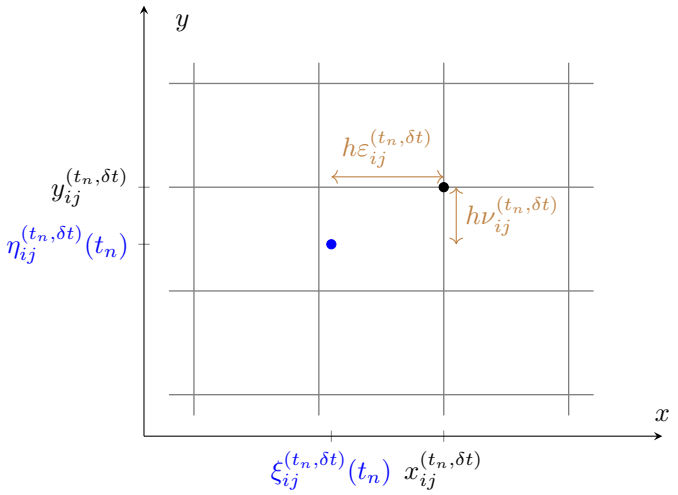

FIGURE 4.14: Decomposition of the departure point $\left(\xi_{ij}^{(t_n,\delta t)}(t_n), \eta_{ij}^{(t_n,\delta t)}(t_n)\right)$ into the sum of the mesh-point $\left(x_{ij}^{(t_n,\delta t)}, y_{ij}^{(t_n,\delta t)}\right)$ to its north-east and its distance from this mesh point $\left(h\varepsilon_{ij}^{(t_n,\delta t)}, h\nu_{ij}^{(t_n,\delta t)}\right)$. All of the mesh points pictured are in the bi-cubic interpolation stencil of the departure point.

4.5.2 The coarse-grid operator

With an understanding of the two-dimensional semi-Lagrangian discretization, we now develop a coarse-grid operator to accompany it. As in the one-dimensional case of Sections 4.2 and 4.3, the key to establishing the truncation error of the two-dimensional semi-Lagrangian method is understanding the error associated with interpolation. Error estimates for multidimensional polynomial interpolation are less well known than in the one-dimensional setting, and are less general, so we provide the estimate required for our purposes in the following lemma. See [47, Sec. 3] and references therein for closely related estimates.

Lemma 4.11 (Polynomial interpolation error in two dimensions)**.** *Let $v(x,y)$ be a function at least $p+1$ times continuously differentiable with respect to both x and y. Let $\mathcal{X}_p(x;y)$ and $\mathcal{Y}_p(y;x)$ be the degree (at most) p polynomials in x and y respectively, interpolating $v(x,y)$ at the $p+1$ nodes $x = \{x_i\}_{i=-\ell(p)}^{r(p)}$ and $p+1$ nodes $y = \{y_j\}_{j=-\ell(p)}^{r(p)}$, respectively. Both sets of interpolation nodes are equispaced by distance h and have $\ell(p), r(p) \geq 0$. Let $v_p(x,y)$ be the two-dimensional interpolating polynomial defined as the tensor product of $\mathcal{X}_p(x;y)$ and $\mathcal{Y}_p(y;x)$. Then, v_p satisfies the following error estimate for any $\varepsilon, \eta \in [0,1)$:*

$$\begin{aligned} &v(x_0 - h\varepsilon, y_0 - h\eta) - v_p(x_0 - h\varepsilon, y_0 - h\eta) \\ &\quad = (-h)^{p+1}\left(f_{p+1}(\varepsilon)\frac{\partial^{p+1}}{\partial x^{p+1}} + f_{p+1}(\eta)\frac{\partial^{p+1}}{\partial y^{p+1}}\right)v(x_0 - h\varepsilon, y_0 - h\eta) + \mathcal{O}(h^{p+2}), \end{aligned} \tag{4.109}$$

Proof. See Appendix B.6. ■

We are now ready to develop an estimate for the truncation error of the semi-Lagrangian method.

Lemma 4.12 (Semi-Lagrangian error for $r = \infty$)**.** *Suppose the solution of* (4.100) *is at least $p+1$ times continuously differentiable with respect to x and y. Let $\mathcal{S}_{p,\infty}^{(t_n,\delta t)}$ be the semi-Lagrangian discretization of* (4.100)*, as described in Section 4.5.1, that exactly locates departure points at time t_n. Finally, let $\boldsymbol{U}(t) \in \mathbb{R}^{n_x^2}$ denote the continuous PDE solution sampled on the discrete spatial mesh at time t. Then, this semi-Lagrangian operator has a local truncation error given by*

$$\begin{aligned}
&\boldsymbol{U}(t_{n+1}) - \mathcal{S}_{p,\infty}^{(t_n,\delta t)}\boldsymbol{U}(t_n) \\
&\quad = (-h)^{p+1}\left[\operatorname{diag}\left(f_{p+1}\left(\boldsymbol{\varepsilon}^{(t_n,\delta t)}\right)\right)\frac{\partial^{p+1}}{\partial x^{p+1}} + \operatorname{diag}\left(f_{p+1}\left(\boldsymbol{\nu}^{(t_n,\delta t)}\right)\right)\frac{\partial^{p+1}}{\partial y^{p+1}}\right]\boldsymbol{U}(t_{n+1}) + \mathcal{O}(h^{p+2}),
\end{aligned} \tag{4.110}$$

in which $\boldsymbol{\varepsilon}^{(t_n,\delta t)}, \boldsymbol{\nu}^{(t_n,\delta t)} \in \mathbb{R}^{n_x^2}$ are vectors whose ijth entries—that is, those associated with mesh point (x_i, y_j)—is equal to the mesh-normalized distances $\varepsilon_{ij}^{(t_n,\delta t)}$, and $\nu_{ij}^{(t_n,\delta t)}$, respectively.

Furthermore, the associated ideal coarse-grid operator defined by time-stepping m times across the interval $t \in [t_n, t_n + m\delta t]$ using the above operator has a local truncation error that may be approximated by

$$\begin{aligned}
&\boldsymbol{U}(t_{n+m}) - \left[\prod_{k=0}^{m-1}\mathcal{S}_{p,\infty}^{(t_n+k\delta t,\delta t)}\right]\boldsymbol{U}(t_n) \\
&\quad \approx (-h)^{p+1}\operatorname{diag}\left(\sum_{k=0}^{m-1} f_{p+1}\left(\boldsymbol{\varepsilon}^{(t_n+k\delta t,\delta t)}\right)\right)\frac{\partial^{p+1}}{\partial x^{p+1}}\boldsymbol{U}(t_{n+m}) \\
&\qquad + (-h)^{p+1}\operatorname{diag}\left(\sum_{k=0}^{m-1} f_{p+1}\left(\boldsymbol{\nu}^{(t_n+k\delta t,\delta t)}\right)\right)\frac{\partial^{p+1}}{\partial y^{p+1}}\boldsymbol{U}(t_{n+m}) + \mathcal{O}(h^{p+2}),
\end{aligned} \tag{4.111}$$

Proof. We omit details of the proof since it follows analogously to the result for the truncation error in one spatial dimension given in Lemma 4.8, with the caveat that one has to invoke the error estimate (4.109) for two-dimensional polynomial interpolation rather than the one-dimensional estimate used previously. ■

Based on our earlier work for the one-dimensional problem (see Section 4.3.1), we can now define a coarse-grid operator with a truncation error which approximately matches—to lowest order—that of the ideal coarse-grid operator. Specifically, supposing that the spatial DOFs are ordered row-wise lexicographically, we propose the following coarse-grid

operator

$$\Phi_{p,r_*}^{(t_n,\delta t)} = \mathcal{B}_{p+1}\mathcal{S}_{p,r_*}^{(t_n,m\delta t)}, \tag{4.112}$$

with backward Euler matrix given by

$$\begin{aligned}\mathcal{B}_{p+1} = \Bigg[I_{n_x^2} - \operatorname{diag}\left(f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n,m\delta t)}\big) - \sum_{k=0}^{m-1} f_{p+1}\big(\boldsymbol{\varepsilon}^{(t_n+k\delta t,\delta t)}\big)\right)(I_{n_x}\otimes\mathcal{D}_{p+1}) \\ - \operatorname{diag}\left(f_{p+1}\big(\boldsymbol{\nu}^{(t_n,m\delta t)}\big) - \sum_{k=0}^{m-1} f_{p+1}\big(\boldsymbol{\nu}^{(t_n+k\delta t,\delta t)}\big)\right)(\mathcal{D}_{p+1}\otimes I_{n_x})\Bigg]^{-1}.\end{aligned} \tag{4.113}$$

Here $\mathcal{D}_{p+1} \in \mathbb{R}^{n_x\times n_x}$ is the same one-dimensional finite-difference discretization used previously.

4.5.3 Numerical results

We now report on numerical results that use our coarse-grid operator (4.112). To begin, we consider the simple constant-coefficient problem

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \quad (x,y,t) \in (-1,1)^2 \times (0,T], \tag{4.114}$$

with initial condition $u(x,y,0) = \sin^2\left[\frac{\pi}{2}(x-1)\right]\sin^2\left[\frac{\pi}{2}(y-1)\right]$, and $T \approx 13.6$. The fine-grid time-step size is taken to be $\delta t = 0.85h$. As for the one-dimensional problems studied previously, we consider semi-Lagrangian discretizations using interpolating polynomials of degree (at most) $p = 1, 3, 5$.

The MGRIT iteration counts for our experiments are shown in Table 4.5. We consider two-level solvers using rediscretization (left column) and our proposed coarse-grid operator (4.112) (middle column). Unsurprisingly, rediscretization performs poorly on this problem, diverging in most cases, and where it does not diverge, it is slow and not scalable with respect to mesh resolution. In contrast, our dissipatively corrected operator performs excellently, leading to fast convergence on all problems. Moreover, the convergence appears to be scalable with respect to the mesh resolution for most problems. Interestingly, the convergence deteriorates as the discretization order p is increased, which also occurred in our tests of one-dimensional advection problems (see, e.g., Table 4.1). Moreover, the iteration counts are quite close to those reported for the one-dimensional constant-coefficient problem (left column, Table 4.1). In the right column of Table 4.5, we show iteration counts for a multilevel solver that employs a multilevel generalization of the dissipatively corrected operator (4.112). This multilevel extension is based on our multilevel generalization for the one-dimensional problem, as described in Section 4.4.1, but we omit further

					$\mathcal{B}_{p+1}\mathcal{S}_p^{(m\delta t)}$					
		$\mathcal{S}_p^{(m\delta t)}$			two level			multilevel		
		m			m			m		
p	$n_x^2 \times n_t$	4	8	16	4	8	16	4	8	16
1	$32^2 \times 256$	19	15	8	12	9	7	12	9	7
	$64^2 \times 512$	28	25	15	12	10	8	12	10	8
	$128^2 \times 1024$	40	43	25	12	10	9	12	10	9
3	$32^2 \times 256$	22	15	8	18	12	8	18	12	8
	$64^2 \times 512$	33	26	15	19	14	11	20	14	11
	$128^2 \times 1024$	50	45	26	20	15	12	20	15	12
5	$32^2 \times 256$	24	15	8	21	13	8	21	13	8
	$64^2 \times 512$	37	26	15	26	17	12	26	17	12
	$128^2 \times 1024$	58	47	26	28	19	16	29	20	16

TABLE 4.5: Number of MGRIT iterations to reduce the space-time residual 2-norm by 10 orders of magnitude for the two-dimensional constant-coefficient problem (4.114). The fine-grid operator is $\mathcal{S}_p^{(\delta t)}$. The **left** column uses two-level cycles with a rediscretized coarse-grid operator. The **middle** column uses two-level cycles with a the backward Euler operator (4.112). Six iterations of GMRES are used to approximately solve the linear systems involving the (inverse of) the matrix (4.113) that arise in applying the backward Euler operator. The **right** column uses multilevel cycles with the multilevel extension of the backward Euler operator; coarsening is carried out until fewer than two points in time on the coarsest grid would arise. For $n_t = 1024$, the number of MGRIT iterations $n_t/(2m)$ needed to reach the exact solution for $m = (4, 8, 16)$ is $(128, 64, 32)$.

details here for the sake of brevity. The iteration counts for the multilevel solver are essentially identical to the two-level counts, indicating that the multilevel coarse-grid operator performs excellently.

We now move to consider the following, more challenging, variable-wave-speed problem

$$\frac{\partial u}{\partial t} + \left[\frac{5}{4} - \cos^2(2\pi y)\right]\frac{\partial u}{\partial x} + \left[\frac{5}{4} + \sin^2\left(\frac{\pi x}{2}\right)\right]\frac{\partial u}{\partial y} = 0. \tag{4.115}$$

The problem setup here is taken to be the same as for the previous problem. The iteration counts of the solves are given in Table 4.6.

In large part, the results in Table 4.6 are qualitatively similar to those for the constant-coefficient problem in Table 4.5. That is, rediscretization yields poor MGRIT convergence, while our coarse-grid operator yields rapid convergence that is scalable with respect to problem size, for both two-levels and many levels. To emphasize the significant improvement our coarse-grid operator provides over rediscretization, we show in Figure 4.15 the residual histories for the $m = 4$ solves of the first-order discretization. Interestingly, the

					$\mathcal{B}_{p+1}\mathcal{S}_{p,r_*}^{(m\delta t)}$					
		$\mathcal{S}_{p,r}^{(m\delta t)}$			two level			multilevel		
		m			m			m		
p, r	$n_x^2 \times n_t$	4	8	16	4	8	16	4	8	16
1, 1	$32^2 \times 256$	29	16	8	10	9	7	10	9	7
	$64^2 \times 512$	47	32	16	10	10	9	10	10	9
	$128^2 \times 1024$	93	60	32	10	10	9	10	10	9
3, 3	$32^2 \times 256$	24	16	8	12	11	8	12	11	8
	$64^2 \times 512$	38	27	16	12	12	11	13	12	11
	$128^2 \times 1024$	65	44	30	12	12	12	13	12	12
5, 5	$32^2 \times 256$	23	16	8	15	12	7	15	12	8
	$64^2 \times 512$	38	26	16	14	15	12	15	15	12
	$128^2 \times 1024$	65	44	26	14	15	14	16	15	14

TABLE 4.6: Identical experiments to those shown in Table 4.5, except for the variable-wave-speed advection problem (4.115). On the finest level, departure points are located with a single ERK step of size δt. The departure points in the backward Euler coarse-grid operators on grid level $\ell \in \mathbb{N}$ are located by taking m^ℓ ERK steps of size δt.

iteration counts in Table 4.6 tend to be smaller than for the constant-coefficient problem in Table 4.5, particularly for the $p = 3$ and $p = 5$ discretizations. This same trend also arose for one-dimensional problems (see Table 4.1).

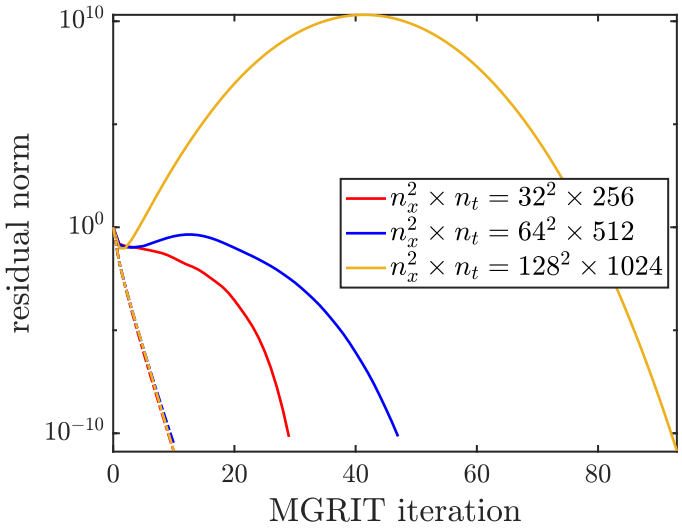

FIGURE 4.15: Two-norm of the space-time residual (relative to its initial value) as a function of two-level MGRIT iteration for the $m = 4$ and $p = r = 1$ solves given in Table 4.6. Solid lines represent solves using a rediscretized coarse-grid operator $\mathcal{S}_{1,1}^{(4\delta t)}$. Broken lines represent solves using the proposed dissipatively corrected coarse-grid operator $\mathcal{B}_2\mathcal{S}_{1,1_*}^{(4\delta t)}$.

Overall, these numerical results show the potential that our proposed coarse-grid operator (4.112) has for parallel-in-time simulations, especially when compared with the standard approach of rediscretization.

4.6 Conclusions

In this chapter we have developed a novel coarse-grid operator for the parallel-in-time solution of semi-Lagrangian discretizations of linear advection or transport problems. This is motivated by the fact that rediscretizing semi-Lagrangian operators on coarse grids leads to divergent MGRIT solvers. Our operator consists of a rediscretized coarse-grid semi-Lagrangian operator followed by a correction, which maps the truncation error of the resulting operator closer to that of the ideal coarse-grid operator. In Section 4.2, both explicit- and implicit-in-time versions of this operator are presented. We argued how these operators can be interpreted as forward and backward Euler time discretizations, respectively, of an advection equation augmented with a high-degree derivative on the right-hand side (Section 4.2.3). Rigorous theory is developed in Section 4.2.4 for constant-wave-speed advection problems, to show that the forward Euler operator has poor stability properties, in the sense that it becomes unstable for larger coarsening factors. However, the same theory is used to prove that the backward Euler operator is unconditionally stable with respect to all problem parameters. Initially developed for the two-level solution of constant-wave-speed problems, the operator is generalized to variable-wave-speed problems in Section 4.3, which includes a novel, and scalable strategy for estimating departure points on coarse levels.

In Section 4.4.1, the operator is extended to the multilevel setting. In all of our numerical tests, it led to fast MGRIT convergence, including for challenging variable-wave-speed problems discretized with 1st-, 3rd-, and 5th-order methods. Furthermore, a proof-of-principle example was shown in Section 4.4.2 for how the operator can be incorporated into the solution of advection-diffusion problems; in the example, convergence was robust with respect to the amount of diffusion in the problem.

Finally, Section 4.5 showed how the operator can be applied to problems in two spatial dimensions, where we obtained fast MGRIT convergence for variable-wave-speed problems using high-order discretizations. Overall, this coarse-grid operator presents a significant advancement in the field of parallel-in-time integration for advection-dominated problems.

Chapter 5

Fully implicit Runge-Kutta methods for method of lines

Runge-Kutta methods have been used repeatedly throughout this thesis in the context of parallel-in-time integration. This chapter changes focus to sequential time integration, considering the numerical solution of fully implicit Runge-Kutta (FIRK) methods that arise in the method-of-lines solution of PDEs. FIRK methods possess many desirable properties compared to their diagonally implicit Runge-Kutta (DIRK) counterparts, yet, unlike DIRK schemes, they do not see widespread use in practice. FIRK methods are challenging to implement efficiently due to the difficulty of computing the stage vectors, which requires solving a large, non-symmetric, block-coupled system of nonlinear algebraic equations. In this chapter, we propose new algorithms for solving this system of nonlinear algebraic equations. Particular emphasis is given to the theoretical analysis of these algorithms, and extensive numerical tests demonstrate the efficacy of our new algorithms.

5.1 Fully implicit Runge-Kutta methods

Let us begin by recalling some key facts about the method of lines, and Runge-Kutta methods. Consider a PDE initial-value problem of the form

$$\frac{\partial u}{\partial t} = L(u,t), \quad t \in (0,T], \quad u(0,\cdot) = u_0, \tag{5.1}$$

with u subject to appropriate boundary conditions in space, and L a (potentially) nonlinear spatial differential operator. Applying the method of lines to this PDE involves first discretizing it in space to arrive at a system of ordinary differential equations (ODEs) in

time of the form

$$\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = \mathcal{N}(\boldsymbol{u}, t), \quad t \in (0, T], \quad \boldsymbol{u}(0) = \boldsymbol{u}_0, \tag{5.2}$$

with $\mathcal{N}\colon \mathbb{R}^N \times [0, T] \to \mathbb{R}^N$ a (potentially) nonlinear, time-dependent spatial discretization. Any ODE solver may now be used to solve (5.2) to generate a fully discrete approximation to PDE (5.1), but here the focus will be on FIRK methods. To this end, consider time discretization of (5.2) using a classical s-stage Runge-Kutta scheme, characterized by the Butcher tableau

$$\begin{array}{c|c} \boldsymbol{c}_0 & A_0 \\ \hline & \boldsymbol{b}_0^\top \end{array} = \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}, \tag{5.3}$$

with Runge-Kutta matrix $A_0 = (a_{ij}) \in \mathbb{R}^{s \times s}$, weight vector $\boldsymbol{b}_0 = (b_1, \ldots, b_s)^\top \in \mathbb{R}^s$, and abscissa $\boldsymbol{c}_0 = (c_1, \ldots, c_s)^\top \in \mathbb{R}^s$. An explicit Runge-Kutta (ERK) method corresponds to a strictly lower triangular A_0, a DIRK method corresponds to a lower triangular A_0, and a FIRK method corresponds to a general matrix A_0 not possessing any triangular structure.

Let $\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$ denote the discrete approximate solution of (5.2) at time $t = t_n$. A Runge-Kutta scheme advances the solution from the current time $t = t_n$ to the new time $t = t_{n+1} := t_n + \delta t$ via a linear combination of stage vectors,

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \delta t \sum_{i=1}^{s} b_i \boldsymbol{k}_i, \qquad \text{where} \tag{5.4}$$

$$\boldsymbol{k}_i = \mathcal{N}\left(\boldsymbol{u}_n + \delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{k}_j, t_n + \delta t c_i\right), \quad i = 1, \ldots, s. \tag{5.5}$$

Equations (5.5) constitute a system of Ns block-coupled, (nonlinear) algebraic equations in the Ns unknowns $(\boldsymbol{k}_i)_{i=1}^s$ to be solved at each time step. The block sparsity structure of this system is characterized by the structure of the matrix A_0 in (5.3). Specifically, for ERK, the right-hand side of equation i in (5.5) only contains stage vectors $\boldsymbol{k}_j$ preceding $\boldsymbol{k}_i$. For DIRK, in addition the preceding $\boldsymbol{k}_j$, the right-hand side of equation i contains $\boldsymbol{k}_i$ itself. Finally, for FIRK, the right-hand side of equation i in (5.5) contains all $\boldsymbol{k}_j$ including $\boldsymbol{k}_i$. The nonlinear system is thus significantly more tightly coupled for FIRK schemes than for ERK and DIRK schemes, where there is a natural order for solving for the s stages $(\boldsymbol{k}_i)_{i=1}^s$ via solving s successive systems of N equations. Furthermore, ERK methods clearly present significantly less of a challenge to compute stage vectors than

implicit Runge-Kutta methods, but for this they pay the penalty that they cannot be A-stable [14, Sec. 23.8], and therefore they are not suitable for the numerical solution of stiff PDEs, for which A-stability is a must. This is why essentially all implicit Runge-Kutta schemes used in practice are A- or L-stable.

Given that FIRK schemes are more difficult to apply than DIRK schemes, two relevant questions are: What advantages do FIRK methods offer over DIRK methods, and are they enough to warrant the extra effort of solving the block coupled equations? FIRK schemes do possess several properties that often make them more desirable than DIRK schemes, with the first such property being their accuracy. An s-stage DIRK scheme has an accuracy limited to order s or $s+1$ [61], while an s-stage FIRK scheme can achieve accuracy up to order $2s$. For example, numerical results will be presented later in this chapter using s-stage Gauss, Radau IIA, and Lobatto IIIC FIRK methods, which have accuracies of order $p = 2s$, $p = 2s - 1$, and $p = 2s - 2$, respectively [48, Tab. 5.13].

A second property is related to the problem-dependent phenomenon of order reduction. Order reduction refers to the situation in which the order of accuracy observed in practice is less than p, and can be as low as the so-called stage order q, which is related to the accuracy that the stage vectors satisfy [61, Sec. 2.2]. Order reduction often arises for stiff problems, such as differential algebraic equations, for example.[1] DIRK schemes have stage order $q = 1$ independent of p, which can be increased to $q = 2$ if an explicit stage is included in the method [61, Sec. 2.2], [48, Ex. 1, Sec. IV.15]. Conversely, FIRK methods can have any stage order; for example, s-stage Gauss, Radau IIA, and Lobatto IIIC methods have stage order equal to s, s, and $s - 1$, respectively [48, Tab. 5.13, and 15.1]. For this reason, if one desires higher than 2nd-order accuracy for stiff problems that suffer from order reduction, then FIRK methods are needed.

Having described some of the important properties of FIRK methods, let us now investigate the underlying structure of the algebraic stage equations (5.5).

5.1.1 Linear ODEs

Let us assume that ODEs (5.2) are linear such that

$$\mathcal{N}(\boldsymbol{u}, t) = \mathcal{L}(t)\boldsymbol{u} + \boldsymbol{g}(t), \tag{5.6}$$

[1] In fact, we have already encountered the phenomenon of order reduction earlier in this thesis in Section 2.5.5 in the context of time-dependent boundary conditions for an advection problem discretized in time using an ERK method. To overcome the order reduction, we adopted a carefully considered strategy for computing the values of stage vectors in the vicinity of the inflow boundary, since if we had computed them naively then the global accuracy of the scheme would have degraded to first or second order, independent of p.

with $\mathcal{L}(t) \in \mathbb{R}^{N\times N}$ a matrix with time-dependent entries, and $\boldsymbol{g}$ a time-dependent forcing term. In this case, system (5.5) can be written as the block linear system

$$\left(\begin{bmatrix} I & & 0 \\ & \ddots & \\ 0 & & I \end{bmatrix} - \delta t \begin{bmatrix} a_{11}\mathcal{L}_1 & \dots & a_{1s}\mathcal{L}_1 \\ \vdots & \ddots & \vdots \\ a_{s1}\mathcal{L}_s & \dots & a_{ss}\mathcal{L}_s \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_s \end{bmatrix}, \tag{5.7}$$

where $\mathcal{L}_i := \mathcal{L}(t_n + \delta t c_i)$ and $\boldsymbol{f}_i := \boldsymbol{g}(t_n + \delta t c_i) + \mathcal{L}(t_n + \delta t c_i)\boldsymbol{u}_n$. Observe that the second block matrix can be written as $\operatorname{diag}(\mathcal{L}_1, \dots, \mathcal{L}_s)\,(A_0 \otimes I_N)$, which motivates the change of variables (which was also applied in [73])

$$\boldsymbol{w} \equiv \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_s \end{bmatrix} = (A_0 \otimes I_N) \begin{bmatrix} \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_s \end{bmatrix} \equiv (A_0 \otimes I_N)\boldsymbol{k}. \tag{5.8}$$

This implicitly assumes that the matrix A_0 is invertible, which is indeed the case for the implicit Runge-Kutta schemes considered here.[2] Let us briefly turn our attention to the Runge-Kutta update (5.4), and make the observation that the required linear combination of stage vectors can be written as a Kronecker product over the scaled stage vectors $\boldsymbol{w}$ using (5.8):

$$\boldsymbol{y} := \sum_{i=1}^{s} b_i \boldsymbol{k}_i = (\boldsymbol{b}_0^\top \otimes I_N)\boldsymbol{k} = (\boldsymbol{d}_0^\top \otimes I_N)\boldsymbol{w}, \quad \text{where} \quad \boldsymbol{d}_0^\top := \boldsymbol{b}_0^\top A_0^{-1}. \tag{5.9}$$

The update then takes the simple form

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \delta t \boldsymbol{y}. \tag{5.10}$$

Under the change of variables (5.8), system (5.7) becomes

$$\left(A_0^{-1} \otimes I_N - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_s \end{bmatrix}. \tag{5.11}$$

Notice this system is sparser than (5.7) since I is sparser than $\mathcal{L}_i$. Furthermore, this change of variables will simplify the analysis appearing later in the chapter.

[2] In fact, the forthcoming Assumption 5.1 states that the eigenvalues of A_0 must have positive real part, of which an immediate implication is that A_0 is invertible.

5.1.2 Nonlinear ODEs

In the case that ODEs (5.2) are nonlinear, it is also advantageous to apply the change of variables (5.8) used in the linear case. Using this change of variables, system (5.5) becomes

$$\mathcal{G}(\boldsymbol{w}) := (A_0^{-1} \otimes I_N) \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_s \end{bmatrix} - \begin{bmatrix} \mathcal{N}(\boldsymbol{u}_n + \delta t \boldsymbol{w}_1, t_n + \delta t c_1) \\ \vdots \\ \mathcal{N}(\boldsymbol{u}_n + \delta t \boldsymbol{w}_s, t_n + \delta t c_s) \end{bmatrix} = 0. \tag{5.12}$$

Notice now that the components of $\boldsymbol{w}$ are only linearly coupled to one another.

Nonlinear system (5.12) is solved via, for example, a Newton-like method, with each Newton iteration requiring the solution of a linearized system of equations. The linearized system matrix is designed to approximate (or equal) the Jacobian of $\mathcal{G}$.

Differentiating system (5.12) leads to a Jacobian given by

$$\mathcal{G}'(\boldsymbol{w}) = A_0^{-1} \otimes I_N - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix}, \tag{5.13}$$

where $\mathcal{L}_i \in \mathbb{R}^{N \times N}$ denotes a linearization of the nonlinear function corresponding to the ith stage vector $\mathcal{N}_i := \mathcal{N}\left(\boldsymbol{u}_n + \delta t \boldsymbol{w}_i, t_n + \delta t c_i\right)$ with respect to $\boldsymbol{w}_i$. Notice this Jacobian matrix is of the same form as the system matrix in (5.11) (excusing the abuse of notation that $\mathcal{L}_i$ previously denoted a genuinely linear operator, but now represents a linearized operator). A Newton-like method applied to nonlinear system (5.12) takes the form

$$\boldsymbol{w}^{(k+1)} \approx \boldsymbol{w}^{(k)} - \mathcal{J}^{-1}\mathcal{G}\big(\boldsymbol{w}^{(k)}\big), \quad \mathcal{J} \approx \mathcal{G}'\big(\boldsymbol{w}^{(k)}\big), \quad k = 0, 1, \ldots \tag{5.14}$$

with nonlinear iteration index k, meaning that $\boldsymbol{w}^{(k)} \approx \boldsymbol{w}$. Taking $\mathcal{J} = \mathcal{G}'(\boldsymbol{w}^{(k)})$ and exactly solving the linear system $\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \mathcal{J}^{-1}\mathcal{G}\big(\boldsymbol{w}^{(k)}\big)$ corresponds to an exact Newton method. Alternatively, forming $\mathcal{J}$ by approximating $\mathcal{G}'(\boldsymbol{w})$ such that $\mathcal{L}_i = \mathcal{L}\,\forall i$ yields a so-called simple Newton method (see, e.g., [15, 8]).

5.1.3 Existing work, assumptions and outline

Moving forward, let $\mathcal{L}$ refer to a spatially linearized (or genuinely linear when appropriate) operator when the stage index is not relevant.

It is now clear that applying FIRK methods to both linear and nonlinear ODEs (5.2) hinges on one's ability to solve $Ns \times Ns$ block linear systems in the form of (5.7)/(5.11).

The fast and efficient solution of this system is typically very difficult for several reasons. The first reason is that typically the dimension N is very large in the context of the method of lines, so direct methods are simply too expensive and, so, iterative methods must be considered instead. Notice also that even in the event that $\mathcal{L}$ is symmetric, the block coupling through A_0/A_0^{-1} is non-symmetric and thus the overall system is non-symmetric. The iterative solution of large, non-symmetric, block-coupled linear systems is a difficult task, and it is for this reason that FIRK methods are seldom used in practice for the numerical simulation of PDEs, despite their excellent stability and accuracy properties. Nonetheless, there has been considerable research on the topic of how to solve systems of the form (5.7)/(5.11).

The earliest work on this topic was in the context of solving ODEs rather than PDEs, and involved making the simplifying assumption that $\mathcal{L}_i = \mathcal{L}\ \forall i$, as happens in a simple Newton method, for example [15, 8]. The motivation for such a simplification is that the system matrix becomes the sum of two Kronecker products, which can then be decomposed by way of a similarity transform into a sequence of smaller $N \times N$ problems. Related approaches that triangularize the system matrix through triangular approximations to A_0 have also been proposed [56, 53].

Most of the more recent work in the context of PDEs has continued to use a simple Newton assumption (or has limited consideration to linear problems with time-independent $\mathcal{L}$). There have been many block preconditioning approaches proposed, such as those of [94, 67], which are based on ideas of Jacobi and Gauss-Seidel splittings of A_0, and have been shown to be effective on parabolic PDEs. A closely-related block preconditioner was proposed in [75] that was based on an approximate factorization of A_0. Like the FIRK algorithms introduced in this chapter, these block-preconditioning approaches allow for the utilization of efficient PDE preconditioners developed for sparse linear operators of the form $\xi I - \zeta\,\mathcal{L}$, for some constants ξ, ζ, which arise in the solution of implicit Euler time integration. Also closely related to the algorithms proposed in this chapter is the algorithm of [4], in which a preconditioning strategy for discontinuous Galerkin time discretizations is introduced. Robustness of the preconditioner in [4] was proven when the spatial discretization is symmetric definite, and a numerical test on a linear advection problem seemed to indicate that it also works well for non-symmetric problems.

The drawback of the simple Newton approximation of course is that if the nonlinear problem at hand is particularly challenging, then the linear convergence rate offered by a simple Newton method may be too slow. Full Newton strategies have been proposed, however, such as that of [73], where system (5.11) is solved by GMRES preconditioned with block ILU techniques, which was shown to be effective on high-order discretizations of the Navier-Stokes equations.

Therefore, while there has been much research on the application of FIRK methods to PDEs, most has been in the context of parabolic and/or symmetric problems, with there being little development for advection-dominated and/or non-symmetric problems, particularly from an analysis point of view. Among other points, this chapter aims to address this apparent void in the literature by providing theoretical analysis of our new FIRK algorithms, with particular emphasis on cases in which $\mathcal{L}$ is highly non-symmetric, and even non-normal.

The algorithms and their analysis presented in this chapter rely on the following assumption regarding eigenvalues of the Butcher matrix A_0 in (5.3) and its inverse A_0^{-1}.

Assumption 5.1. *Assume that all eigenvalues of A_0 (and equivalently A_0^{-1}) have positive real part.*

If an implicit Runge-Kutta method is A-stable, irreducible, and A_0 is invertible (which includes DIRK, Gauss, Radau IIA, and Lobatto IIIC methods, among others), then Assumption 5.1 holds [48].

We now describe a second assumption pertaining to the stable time integration of ODEs (5.2). Specifically, it is important to consider the stability of the associated (linearized) Dalhquist test problem $\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = \mathcal{L}\boldsymbol{u}$. Supposing the solution to the Dalhquist problem itself is stable, a necessary condition for the stable numerical integration of this problem is that the eigenvalues of $\delta t\,\mathcal{L}$ lie inside the stability region of the Runge-Kutta method [76]. If $\mathcal{L}$ is normal, that is, it has a full set of orthonormal eigenvectors, then this is both a necessary and sufficient condition for stability. However, if $\mathcal{L}$ is non-normal, or even non-diagonalizable, which can happen for certain discretizations of hyperbolic equations (see [76, Sec. 10.2]), for example, then the situation becomes more complicated. To this end, we introduce the following assumption with respect to $\mathcal{L}$. Let $W(\mathcal{L})$ denote the field of values of $\mathcal{L}$, which is the subset of the complex plane given by (see, e.g., [5])

$$W(\mathcal{L}) := \{\langle \mathcal{L}\boldsymbol{x}, \boldsymbol{x}\rangle \colon \|\boldsymbol{x}\| = 1\}\,. \tag{5.15}$$

Then,

Assumption 5.2. *Assume that $W(\mathcal{L}) \leq 0$; that is, $W(\mathcal{L})$ is a subset of the left half plane (including the imaginary axis).*

This assumption can be seen as a sufficient condition for the stable time integration of the Dalhquist test problem with A-stable methods—whose stability region is the closed left half plane—regardless of the normality of $\mathcal{L}$.[3] Finally, note that for a normal matrix $\mathcal{L}$,

[3] See our articles [91, 90] for further details.

$W(\mathcal{L})$ is equal to the convex hull of its eigenvalues [5, Prop. 10]. Therefore, for normal $\mathcal{L}$, Assumption 5.2 reduces to aforementioned necessary and sufficient condition that the eigenvalues of $\mathcal{L}$ lie in the closed left half plane for the numerically stable integration with an A-stable method. In particular, we will provide theoretical analysis for cases in which $\mathcal{L}$ is either symmetric negative semi-definite or skew symmetric, which are examples of normal matrices with their eigenvalues in the closed left half plane.

The remainder of this chapter is organised as follows. Section 5.2 introduces an algorithm for linear ODEs. It then examines the efficacy of the underlying preconditioner, first under the assumption that the spatial discretization is symmetric definite and then that it is skew symmetric. Following this, Section 5.3 briefly introduces a simple Newton algorithm for the solution of nonlinear FIRK problems. It then examines the efficacy of the underlying linear preconditioner under the assumption that the spatial discretization is symmetric definite. Section 5.4 generalizes the linear preconditioning theories of Sections 5.2 and 5.3 by removing the assumption of symmetry or skew symmetry on the spatial discretization. Finally, Section 5.5 discusses further some aspects of our `C++` package [89] that implements these algorithms, and presents numerical results for both linear and nonlinear PDEs that confirm the linear preconditioning theory of the earlier sections.

5.2 The linear setting

This section considers the solution of the block linear system (5.11) that results from the application of FIRK methods to linear ODEs (5.6). First Section 5.2.1 proposes a new algorithm for solving this problem, along with some details of its software implementation. An outline and discussion of the sections to follow is then given in Section 5.2.2, and eigenvalue analyses of the proposed preconditioner are the subject of Sections 5.2.3 and 5.2.4.

5.2.1 The algorithm: Preconditioning complex-conjugate pairs

It is first necessary to make the simplifying assumption that the spatial discretization $\mathcal{L}$ is time independent: $\mathcal{L}_i \equiv \mathcal{L}$. Now, let us define the system matrix in (5.11) as

$$\mathcal{M}_s := A_0^{-1} \otimes I_N - I_s \otimes \widehat{\mathcal{L}}, \qquad \text{where} \quad \widehat{\mathcal{L}} := \delta t\, \mathcal{L}\,. \tag{5.16}$$

Throughout the rest of the chapter, we will interchangeably use the operator $\delta t \mathcal{L}$ and its shorthand $\widehat{\mathcal{L}}$ defined here.

As described earlier, applying a FIRK scheme to a linear ODE requires inverting (5.16), and the following Lemma (cited here without proof) provides us a powerful way of expressing its inverse.

Lemma 5.1 (A condensed, slightly modified version of [91, Lemma 5])**.** *Let $\mathcal{M}_s$ be as in (5.16), let $\operatorname{adj}(\mathcal{M}_s)$ be the adjugate of $\mathcal{M}_s$, and let $P_s(x)$ be the characteristic polynomial of A_0^{-1}. Then,*

$$\mathcal{M}_s^{-1} = \big(I_s \otimes [P_s(\widehat{\mathcal{L}})]^{-1}\big) \operatorname{adj}(\mathcal{M}_s). \tag{5.17}$$

Now consider applying the closed-form inverse (5.17) to solve the block system $\mathcal{M}_s \boldsymbol{w} = \boldsymbol{f}$ from (5.11) for the stage vectors $\boldsymbol{w} \in \mathbb{R}^{Ns}$. Ultimately, only the Runge-Kutta update $\boldsymbol{y} = (\boldsymbol{d}_0^\top \otimes I_N)\boldsymbol{w} \in \mathbb{R}^N$ is needed to step to the new time level, and this may be conveniently expressed as

$$\begin{aligned} \boldsymbol{y} &= \big(\boldsymbol{d}_0^\top \otimes I_N\big)\mathcal{M}_s^{-1}\boldsymbol{f}, && (5.18)\\ &= \big(\boldsymbol{d}_0^\top \otimes I_N\big)\big(I_s \otimes [P_s(\widehat{\mathcal{L}})]^{-1}\big) \operatorname{adj}(\mathcal{M}_s)\boldsymbol{f}, && (5.19)\\ &= [P_s(\widehat{\mathcal{L}})]^{-1}\big(\boldsymbol{d}_0^\top \otimes I_N\big) \operatorname{adj}(\mathcal{M}_s)\boldsymbol{f}, && (5.20) \end{aligned}$$

where $\boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)^\top$. Or equivalently, $\boldsymbol{y}$ solves the $N \times N$ linear system

$$P_s(\widehat{\mathcal{L}})\boldsymbol{y} = \boldsymbol{z} := \big(\boldsymbol{d}_0^\top \otimes I_N\big) \operatorname{adj}(\mathcal{M}_s)\boldsymbol{f}. \tag{5.21}$$

Upon solving (5.21), for $\boldsymbol{y}$, the solution at the new time level can simply be evaluated by $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \delta t \boldsymbol{y}$, as in (5.10).

The numerical solution of system (5.21) will be discussed shortly, but first, let us turn our attention to forming the right hand side vector $\boldsymbol{z}$. Due to the structure of $\mathcal{M}_s$ in (5.16) (more specifically, that its elements may be defined over the commutative ring of linear combinations of I and $\widehat{\mathcal{L}}$; see [91] and references therein), its adjugate may be defined over matrix-valued elements. Since $\mathcal{M}_s = A_0^{-1} \otimes I_N - I_s \otimes \widehat{\mathcal{L}}$, let us consider the adjugate of the matrix $A_0^{-1} - xI_s$, which will be denoted as $Q(x)$. The elements in $Q(x)$ are polynomials. More specifically, these can be written in the general form

$$Q(x) := \operatorname{adj}\left(A_0^{-1} - xI_s\right) \in \mathbb{R}^{s\times s}, \quad \text{where} \quad Q_{ij}(x) = \sum_{k=0}^{s-1} q_k^{(ij)} x^k, \tag{5.22}$$

with the sets of coefficients $\{q^{(ij)}\}$ depending on the elements of A_0^{-1}. The adjugate of $\mathcal{M}_s$ can then be defined by evaluating Q at $x = \widehat{\mathcal{L}}$:

$$\operatorname{adj}(\mathcal{M}_s) = \begin{bmatrix} Q_{11}(\widehat{\mathcal{L}}) & \cdots & Q_{1s}(\widehat{\mathcal{L}}) \\ \vdots & & \vdots \\ Q_{s1}(\widehat{\mathcal{L}}) & \cdots & Q_{ss}(\widehat{\mathcal{L}}) \end{bmatrix} \in \mathbb{R}^{Ns \times Ns}. \tag{5.23}$$

Next, observe that the factor appearing in front of this matrix in (5.21) takes inner products with $\boldsymbol{d}_0$ over its columns to form a $N \times Ns$ matrix. There is then an inner product between this matrix and the vector $\boldsymbol{f} \in \mathbb{R}^{Ns}$. In conclusion, this means that the right hand side of (5.21) can be expressed as

$$\boldsymbol{z} = \sum_{j=1}^{s} R_j(\widehat{\mathcal{L}})\boldsymbol{f}_j, \quad \text{where} \quad R_j(x) := \sum_{k=0}^{s-1} \left(\sum_{i=1}^{s} d_i \, q_k^{(ij)} \right) x^k. \tag{5.24}$$

Here R_j is the polynomial arising from taking the inner product of $\boldsymbol{d}_0$ with the jth column of $\operatorname{adj}(\mathcal{M}_s)$ in (5.23).

In evaluating (5.24), the polynomial $R_j(\widehat{\mathcal{L}})$ should not be explicitly formed to compute its action. In our `C++` package [89], this action is computed via a Horner-like scheme (as in the well-known scheme for efficiently evaluating scalar polynomials). Such a scheme is optimal because it only requires the action of $\widehat{\mathcal{L}}$ $s-1$ times.

In our software [89], symbolic expressions for the polynomial coefficients of R_j defined in (5.24) are hard-coded in terms of the entries of arbitrary A_0^{-1} and $\boldsymbol{d}_0$ for various $s \approx \mathcal{O}(1)$ (e.g., currently $s = 1 \to 5$ are supported). Once the user specifies their desired FIRK scheme at run time, the polynomial coefficients are evaluated numerically by substituting into the symbolic expressions numerical values for the entries of A_0^{-1} and $\boldsymbol{d}_0$. The motivation for using hard-coded symbolic expressions written in terms of arbitrary A_0^{-1} is that if a user provides Butcher tableau information (i.e., they specify an A_0^{-1}), they do not need access to software to compute the adjugate of $A_0^{-1} - xI_s$ for symbolic x. To generate the symbolic expressions used in [89], the adjugate in (5.22) has been computed in MATLAB using its `adjoint` function on a symbolic $A_0^{-1} - xI_s$ matrix (i.e., with both A_0^{-1} and x being symbolic variables). Symbolic expressions for each of the coefficients in (5.24) are then constructed using a symbolic $\boldsymbol{d}_0$ vector. Since the resulting symbolic expressions are very lengthy (e.g., the longest formulas for the $s = 5$ coefficients each contain hundreds of appearances of the entries of A_0^{-1}), the symbolic MATLAB expressions are translated in an automated fashion into `C++` friendly expressions so that they may be easily hard coded.

With the knowledge of how to form the right hand side of system (5.21), let us turn our attention to its numerical solution. Remarkably, the original block $Ns \times Ns$ system (5.11)

has been transformed into the single $N \times N$ system (5.21) by exploiting its Kronecker-product structure and the fact that only a linear combination of the s stage vectors is needed for the Runge-Kutta update (5.10). Our lunch is *not completely free*, however, because the characteristic polynomial of A_0^{-1} appearing in (5.21) is of degree s. Letting $\{\lambda_i\}$ denote the s eigenvalues of A_0^{-1}, the characteristic polynomial in (5.21) may be expressed in factored form as

$$P_s(\widehat{\mathcal{L}}) = \prod_{i=1}^{s}(\lambda_i I - \widehat{\mathcal{L}}), \tag{5.25}$$

and hence its inverse may be computed by successively inverting the matrices $(\lambda_i I - \widehat{\mathcal{L}})$ for $i = 1, ..., s$. Unfortunately, however, the eigenvalues of A_0^{-1} for FIRK schemes are complex in general (since A_0 is neither triangular nor symmetric), and, so, inverting factors $(\lambda_i I - \widehat{\mathcal{L}})$ for real-valued matrices $\widehat{\mathcal{L}}$ is likely not very practical using standard PDE preconditioners and existing software because they typically are not applicable to complex systems.

Instead, the key idea proposed here is to combine complex-conjugate pairs in (5.25) and then invert the resulting real, quadratic operator. To this end, let $\lambda_i := \eta_i + \mathrm{i}\beta_i$ denote a complex eigenvalue of A_0^{-1}, for $\eta_i, \beta_i \in \mathbb{R}$, with $\eta_i > 0$ under Assumption 5.1, and without loss of generality $\beta_i \geq 0$. The real quadratic associated with the eigenvalue pair $(\lambda_i, \bar{\lambda}_i) = (\eta_i + \mathrm{i}\beta_i, \eta_i - \mathrm{i}\beta_i)$ is then

$$\mathcal{Q}_i := \left[(\eta_i + \mathrm{i}\beta_i)I - \widehat{\mathcal{L}}\right]\left[(\eta_i - \mathrm{i}\beta_i)I - \widehat{\mathcal{L}}\right] = (\eta_i I - \widehat{\mathcal{L}})^2 + \beta_i^2 I. \tag{5.26}$$

Thus, inverting operator (5.25) (or rather solving the associated linear system (5.21)) can be achieved through inverting sequentially $\frac{s}{2}$ quadratic factors of the form (5.26) for the $\frac{s}{2}$ pairs $(\eta_i, \beta_i)_{i=1}^{s/2}$ if s is even, or $\frac{s-1}{2}$ factors of the form (5.26) and a single, real, linear factor of the form $\zeta_i I - \widehat{\mathcal{L}}$, with ζ_i being a real eigenvalue of A_0^{-1}, if s is odd (recalling A_0^{-1} will have $\frac{s-1}{2}$ complex-conjugate pairs of eigenvalues and a single real eigenvalue if s is odd).

In practice, it is not desirable to directly form or precondition (5.26), due to the overhead cost of large parallel matrix multiplication (if the matrix is even explicitly available, that is), and because many fast parallel methods are not designed for solving a polynomial in $\widehat{\mathcal{L}}$. To get around this, it is proposed to solve (5.26) using a Krylov subspace method because this only requires the action of the quadratic operator. Based on the structure of (5.26) when $\beta_i \approx 0$, a preconditioner of the form $(\gamma_i I - \widehat{\mathcal{L}})^{-2}$ with $\gamma_i \in (0, \infty)$ being a free parameter is proposed.[4] In fact, a naive choice would be to simply use $\gamma_i = \eta_i$, but as the

[4] In Section 5.4, a more general preconditoner with two free constants $\delta_i, \gamma_i \in (0, \infty)$ of the form $(\delta_i I - \widehat{\mathcal{L}})^{-1}(\gamma_i I - \widehat{\mathcal{L}})^{-1}$ is considered. It is shown in Corollary 5.15 of Section 5.4, however, that in some sense the optimal form of this preconditioner is with $\delta_i = \gamma_i$, which is why we consider such a preconditioner here.

forthcoming analysis will show one can do significantly better than this choice.

The preconditioned operator corresponding to an arbitrary complex-conjugate pair of eigenvalues $\eta \pm \mathrm{i}\beta$, denoted $\mathcal{P}_\gamma$, is

$$\mathcal{P}_\gamma := \big[(\eta I - \widehat{\mathcal{L}})^2 + \beta^2 I\big](\gamma I - \widehat{\mathcal{L}})^{-2}, \quad \gamma \in (0, \infty). \tag{5.27}$$

While the preconditioner $(\gamma I - \widehat{\mathcal{L}})^{-2}$ is a quadratic operator, it appears in factored form, and so its action can be applied by two successive applications of the linear factor $(\gamma I - \widehat{\mathcal{L}})^{-1}$. Notice that if $\widehat{\mathcal{L}}$ is symmetric negative semi-definite, then $\mathcal{P}_\gamma$ (5.27) is symmetric positive definite for $\gamma > 0$ and therefore the linear system may be solved using a Krylov method that exploits this, such as the conjugate gradient method (CG), for example.[5] Otherwise, for non-symmetric $\widehat{\mathcal{L}}$, GMRES is typically applied to solve (5.26).

In practice it is undesirable to exactly apply the action of the preconditioner $(\gamma I - \delta t\, \mathcal{L})^{-2}$ at every Krylov iteration since the overall cost of the solve is typically less if it is approximated instead. Therefore, instead of exactly applying the preconditioner, it is approximated by *twice* applying the action of an inexpensive preconditioner for $(\gamma I - \delta t\, \mathcal{L})^{-1}$. For example, the numerical tests in Section 5.5 use a single iteration of AMG to approximate $(\gamma I - \delta t\, \mathcal{L})^{-1}$. Note, however, that all of the analysis in this chapter is done under the assumption that the preconditioner is exactly applied. This is typical of preconditioning analyses in the literature where one proves robustness of a preconditioner under the assumption that it is exactly applied, but then goes on to approximately apply it in practice. Furthermore, for the preconditioning analysis to be indicative of what is observed in practice, it will likely be necessary that the inexpensive preconditioner used to approximate $(\gamma I - \delta t\, \mathcal{L})^{-1}$ does so *relatively well*.

5.2.2 Outline and assumptions for linear eigenvalue analysis

The remainder of Section 5.2 is devoted to analysing the two-norm condition number of the preconditioned operator (5.27). In particular, Section 5.2.3 does so under the assumption that the spatial discretization $\mathcal{L}$ is symmetric negative semi-definite, and Section 5.2.4 does so under the assumption that $\mathcal{L}$ is skew symmetric. In each of these cases, the value of γ in (5.27) is identified that leads to the minimization of a tight upper bound on the condition number of (5.27), and therefore hopefully the most robust preconditioner for the Krylov inversion of the quadratic operator (5.26). It is found that that optimal value of $\gamma = \gamma_*$ is the same for both classes of $\mathcal{L}$. In the symmetric definite case, the condition number of $\mathcal{P}_{\gamma_*}$ is bounded by two, independent of the order of FIRK integration. In the

[5] Note (5.27) is symmetric because one of the trailing $(\gamma I - \widehat{\mathcal{L}})^{-1}$ factors can be pulled to the front of the operator since rational functions of $\widehat{\mathcal{L}}$ commute.

skew-symmetric case, the condition number is found to grow weakly with the number of FIRK stages/FIRK order, but it remains $\mathcal{O}(1)$ for a moderate number of stages/order of accuracy. For example, it is less than 2.5 for some common 5-stage FIRK methods having 8th-, 9th- and 10th-order accuracy.

It should be noted that the conditioning theory of Section 5.2.3 (for the symmetric negative semi-definite case) essentially arrives at the same results as that of [4, Proposition 3.3] (the preconditioned operator proposed there is related to the one discussed here); however, the analysis here was done independently and it contains interesting additional elements that were not considered in [4]. Moreover, it helps to contextualize Sections 5.2.4 and 5.3.3 where a similar style of analysis is applied to more challenging problems.

Only in the case of CG does the condition number of (5.27) translate directly into convergence bounds of the solver (related bounds do apply to GMRES in the case that the preconditioned system is SPD, but this is of minimal interest here). However, generally speaking, the condition number of (5.27) provides an excellent measure of the robustness of any Krylov method's ability to solve the preconditioned linear system. In particular, understanding the dependencies of the condition number on algorithmic parameters (e.g., properties of $\mathcal{L}$, δt, integration order) is necessary to assess its robustness and optimality. In any event, conditioning of $\mathcal{O}(1)$ indicates that Krylov convergence is likely to be fast in practice.

The practical motivation for analysing the preconditioner when $\mathcal{L}$ is symmetric definite is that spatial discretizations for parabolic PDEs often fall into this category. A secondary and theoretical motivation is that such matrices usually simplify significantly the analysis because of their simple properties; for example, some other FIRK algorithms for PDEs have been analysed under similar assumptions [4, 94]. The motivation for considering skew-symmetric $\mathcal{L}$ is not as clear-cut as the symmetric definite case since such an operator would not typically result from the spatial discretization of a *real-world problem*. Nonetheless, closely related matrices arise from non-dissipative discretizations (e.g., central finite-difference discretizations) of advection-dominated PDEs, and it is often the case that *good* discretizations of advection operators will have eigenvalues with large imaginary components, since the eigenvalues of the continuous advection operator are purely imaginary. So, while the skew-symmetric case is unlikely to arise exactly in practice, its analysis undoubtedly provides insight for advection-dominated problems. Recall also that since symmetric negative semi-definite and skew-symmetric matrices are normal with eigenvalues contained in the closed left half plane, they fulfil Assumption 5.2.

To simplify the forthcoming analysis, it will be useful to make the following change of variables from those appearing in (5.27): Assuming without loss of generality that $\beta > 0$,

let us define[6]

$$x := \frac{\eta}{\beta}, \quad y := \frac{\gamma}{\beta}, \quad \mathcal{A} := \frac{1}{\beta}\,\widehat{\mathcal{L}}\,. \tag{5.28}$$

Notice that $x, y > 0$ since $\eta, \beta, \gamma > 0$, and also that $\mathcal{A}$ inherits any symmetry or definiteness properties from $\widehat{\mathcal{L}}$. Applying these new variables to (5.27) results in the slightly simpler preconditioned operator

$$\mathcal{P}_y := \big[(xI - \mathcal{A})^2 + I\big](yI - \mathcal{A})^{-2}, \tag{5.29}$$

which depends on only two parameters rather than three. Further note that the condition number of operators (5.27) and (5.29) is the same.

Since $\mathcal{A}$ is normal, (5.29) is too, and, as such, its two-norm condition number can be expressed it terms of its absolute maximum and minimum eigenvalues:

$$\kappa(\mathcal{P}_y) := \big\|\,\mathcal{P}_y\,\big\|_2 \,\big\|\,\mathcal{P}_y^{-1}\,\big\|_2 = \frac{\max\limits_{\lambda\in\sigma(-\mathcal{A})} |\,\mathcal{F}_y(\lambda)|}{\min\limits_{\lambda\in\sigma(-\mathcal{A})} |\,\mathcal{F}_y(\lambda)|}, \tag{5.30}$$

where $\sigma(-\mathcal{A})$ denotes the spectrum of $-\mathcal{A}$, and the function $\mathcal{F}_y(\lambda)$ is the eigenvalue of $\mathcal{P}_y$ associated with eigenvalue λ of $-\mathcal{A}$,

$$\mathcal{F}_y(\lambda) := \frac{(x+\lambda)^2 + 1}{(y+\lambda)^2}. \tag{5.31}$$

Rather than working with the exact condition number (5.30), the following more tractable upper bound is considered instead:

$$\widetilde{\kappa}(\mathcal{P}_y) := \frac{\max\limits_{\lambda\in\mathcal{I}} |\,\mathcal{F}_y(\lambda)|}{\min\limits_{\lambda\in\mathcal{I}} |\,\mathcal{F}_y(\lambda)|} \geq \kappa(\mathcal{P}_y), \tag{5.32}$$

where $\mathcal{I}$ is the interval defined as

$$\mathcal{I} := \begin{cases} [0,\infty), & \text{if } \mathcal{A} \text{ is symmetric negative semi-definite,} \\ (-\mathrm{i}\infty, \mathrm{i}\infty), & \text{if } \mathcal{A} \text{ is skew symmetric.} \end{cases} \tag{5.33}$$

The bound (5.32) corresponds to what would be the condition number of $\mathcal{P}_y$ if the spectra of $\mathcal{A}$ densely filled their allowable interval (the entirety of the non-positive real line and the entirety of the imaginary line in the symmetric and skew-symmetric cases, respectively). In this sense $\widetilde{\kappa}(\mathcal{P}_y)$ can be interpreted as the maximum condition number of $\mathcal{P}_y$ over

[6]Excluding the case $\beta = 0$ is not of any real significance since it can be seen immediately from (5.27) that if $\beta = 0$, one obtains the identity preconditioned operator by taking $\gamma = \eta$.

the space of either (i) all symmetric negative semi-definite matrices $\mathcal{A}$, or (ii) all skew-symmetric matrices $\mathcal{A}$. It should also be noted that bound (5.32) is tight because it will achieve equality for any matrix $\mathcal{A}$ satisfying both $\max_{\lambda\in\sigma(-\mathcal{A})} |\mathcal{F}_y(\lambda)| = \max_{\lambda\in\mathcal{I}} |\mathcal{F}_y(\lambda)|$ and $\min_{\lambda\in\sigma(-\mathcal{A})} |\mathcal{F}_y(\lambda)| = \min_{\lambda\in\mathcal{I}} |\mathcal{F}_y(\lambda)|$.

While the assumption that the spectra of $\mathcal{A}$ are dense in the interval given in (5.33) is to make the analysis tractable, such an assumption in the symmetric case is not too unrealistic because (loosely speaking) the largest eigenvalue of $-\mathcal{A}$ arising from a diffusion-dominated problem would typically scale as $\mathcal{O}\big(\delta t/h^2\big)$ for spatial mesh size h, and the smallest would be zero in the case of periodic boundary conditions, or would scale as $\mathcal{O}(\delta t)$ for Dirichlet boundary conditions. Conversely, the assumption is not as realistic in the skew-symmetric case because the extremal eigenvalues of $\mathcal{A}$ arising from an advection-dominated problem would typically scale like $\mathcal{O}\big(\pm \mathrm{i}\delta t/h\big)$. Nonetheless, these assumptions are ultimately justified in the end since the condition number $\widetilde{\kappa}(\mathcal{P}_y)$ in (5.32) is found to be of size $\mathcal{O}(1)$ for what is considered to be quite high-order FIRK integration (recalling the condition number of any matrix is bounded below by unity).

5.2.3 Eigenvalue analysis in the linear setting: The symmetric definite case

Lemma 5.2 (Conditioning for symmetric negative semi-definite $\mathcal{A}$)**.** *Suppose that $\mathcal{A}$ is symmetric negative semi-definite, then condition number $\widetilde{\kappa}(\mathcal{P}_y)$ in (5.32) that tightly bounds the condition number of the preconditioned operator $\mathcal{P}_y$ in (5.29) is equal to*

$$\widetilde{\kappa}(\mathcal{P}_y) = \begin{cases} \dfrac{x^2+1}{y^2}, & y \in (0, x], \\ \dfrac{(x^2+1)\big[1+(y-x)^2\big]}{y^2}, & y \in \left[x, \sqrt{x^2+1}\right], \\ 1+(y-x)^2, & y \in \left[\sqrt{x^2+1}, \frac{x^2+1}{x}\right], \\ \dfrac{y^2}{x^2+1}, & y \in \left[\frac{x^2+1}{x}, \infty\right). \end{cases} \tag{5.34}$$

Proof. To compute $\widetilde{\kappa}(\mathcal{P}_y)$ in (5.32) the global extrema of $|\mathcal{F}_y(\lambda)|$ need to be identified for $\lambda \in \mathcal{I} := [0, \infty)$ (see (5.33)). However, observe from (5.31) that $\mathcal{F}_y(\lambda)$ is positive for any such λ since $x, y > 0$; therefore, it is simply the extrema of $\mathcal{F}_y(\lambda)$ that need to be identified. Global extrema may occur at the boundaries of $\mathcal{I}$, or at stationary points in λ (that is, where the derivative of $\mathcal{F}_y$ vanishes). Consider first the behaviour of $\mathcal{F}_y(\lambda)$ at the boundaries of $\mathcal{I}$:

$$\mathcal{F}_y(0) = \frac{1+x^2}{y^2}, \quad \lim_{\lambda\to\infty} \mathcal{F}_y(\lambda) = 1. \tag{5.35}$$

Now consider the derivative. Computing this, one obtains

$$\frac{\mathrm{d}\mathcal{F}_y}{\mathrm{d}\lambda} = 2\frac{(y-x)\lambda - (x^2+1-xy)}{(y+\lambda)^3}. \tag{5.36}$$

The derivative can only vanish at the single point $\lambda = \lambda_*$ defined by

$$\lambda_* := \frac{x^2+1-xy}{y-x} = \frac{1}{x}\frac{1}{y-x}\left(\frac{x^2+1}{x} - y\right), \quad y \neq x. \tag{5.37}$$

Recall that for λ_* to be a stationary point of interest, it must lie in $\mathcal{I} = [0, \infty)$. Based on its factorized form above, it is easy to see that the sign of λ_* is

$$\operatorname{sign}(\lambda_*) = \begin{cases} -1, & y \in (0, x), \\ 1, & y \in \left(x, \frac{x^2+1}{x}\right), \\ 0, & y = \frac{x^2+1}{x}, \\ -1, & y \in \left(\frac{x^2+1}{x}, \infty\right). \end{cases} \tag{5.38}$$

Therefore λ_* is only a stationary point of interest when $y \in \left(x, \frac{x^2+1}{x}\right]$. Evaluating $\mathcal{F}_y(\lambda)$ (5.31) at $\lambda = \lambda_*$ gives the local extremum

$$\mathcal{F}_y(\lambda_*) = \frac{1}{1+(y-x)^2} < 1, \quad y \in \left(x, \frac{x^2+1}{x}\right]. \tag{5.39}$$

Since $\mathcal{F}_y(\lambda_*) < 1$, considering that one of $\mathcal{F}_y(0)$ and $\mathcal{F}_y(\infty)$ is always at least unity (5.35), $\mathcal{F}_y(\lambda_*)$ can never be a contender for the global maximum of $\mathcal{F}_y$. Simple algebra then reveals the global maximum is

$$\max_{\lambda \in \mathcal{I}} \mathcal{F}(\lambda) = \begin{cases} \dfrac{1+x^2}{y^2}, & y \in \left(0, \sqrt{x^2+1}\right], \\ 1, & y \in \left[\sqrt{x^2+1}, \infty\right). \end{cases} \tag{5.40}$$

The global minimum is less straightforward to compute. It is useful first to note the inequalities $x < \sqrt{x^2+1} < \frac{x^2+1}{x}$. Since $\mathcal{F}_y(\infty) = 1$, it is clearly going to be the global minimum for $y \in (0, x]$. Then, since $\mathcal{F}_y(\lambda_*) < 1$, it must be the global minimum on at least the interval $(x, \sqrt{x^2+1}]$ where $\mathcal{F}_y(0) \geq 1$. The minimum of $\mathcal{F}_y(0)$ and $\mathcal{F}_y(\lambda_*)$ over $y \in \left(\sqrt{x^2+1}, \frac{x^2+1}{x}\right)$ is less clear. Notice though that both these functions are strictly decreasing on this interval and that they are in fact equal at the right end-point: $\mathcal{F}_{(x^2+1)/x}(0) = \mathcal{F}_{(x^2+1)/x}(\lambda_*) = \frac{x^2}{x^2+1}$. It must be therefore that $\mathcal{F}_y(\lambda_*)$ is the smaller of

the two. Summarizing this discussion, the global minimum of $\mathcal{F}_y$ is given by

$$\min_{\lambda\in[0,\infty)} \mathcal{F}(\lambda) = \begin{cases} 1, & y \in (0, x], \\ \dfrac{1}{1+(y-x)^2}, & y \in \left[x, \frac{x^2+1}{x}\right], \\ \dfrac{x^2+1}{y^2}, & y \in \left[\frac{x^2+1}{x}, \infty\right). \end{cases} \tag{5.41}$$

Taking the ratio of (5.40) to (5.41) gives the condition number of (5.34). ■

Lemma 5.3 (Optimal preconditioning for symmetric negative semi-definite $\mathcal{A}$)**.** *The tight condition number bound $\widetilde{\kappa}(\mathcal{P}_y)$ from (5.34) in Lemma 5.2 is minimized over $y \in (0, \infty)$ at $y = y_*$, where*

$$y_* = \sqrt{x^2+1}. \tag{5.42}$$

Moreover, the minimum value of $\widetilde{\kappa}(\mathcal{P}_y)$ in (5.34) is

$$\widetilde{\kappa}(\mathcal{P}_{y_*}) = 2\left(x^2 - x\sqrt{x^2+1} + 1\right). \tag{5.43}$$

Proof. First observe from (5.34) that $\widetilde{\kappa}(\mathcal{P}_y)$ is strictly decreasing on the interval $y \in (0, x)$ where it is equal to $\frac{x^2+1}{y^2}$, while it is strictly increasing on the interval $y \in \left(\frac{x^2+1}{x}, \infty\right)$ where it is equal to $\frac{y^2}{x^2+1}$. Furthermore, $\widetilde{\kappa}(\mathcal{P}_y)$ is strictly increasing over $y \in \left(\sqrt{x^2+1}, \frac{x^2+1}{x}\right)$ where it is equal to $1 + (y-x)^2$.

By the continuity of $\widetilde{\kappa}(\mathcal{P}_y)$ with respect to y, the global minimum of the function must lie in the interval $y \in \left[x, \sqrt{1+x^2}\right]$. To this end, consider the derivative of $\widetilde{\kappa}(\mathcal{P}_y)$ in this interval, which may be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}y}\widetilde{\kappa}(\mathcal{P}_y) = -\frac{2(x^2+1)}{xy^3}\left(\frac{x^2+1}{x} - y\right), \quad y \in \left(x, \sqrt{x^2+1}\right). \tag{5.44}$$

Clearly this function is negative for any $y \in \left(0, \frac{x^2+1}{x}\right)$, and as noted in the previous proof, $\frac{x^2+1}{x} > \sqrt{x^2+1}$. Therefore, $\frac{\mathrm{d}}{\mathrm{d}y}\widetilde{\kappa}(\mathcal{P}_y) < 0$ when $y \in \left(x, \sqrt{x^2+1}\right)$, and, so, $\widetilde{\kappa}(\mathcal{P}_y)$ is strictly decreasing. The global minimum of $\widetilde{\kappa}(\mathcal{P}_y)$ must occur at the right hand boundary of this interval, $y = y_* = \sqrt{x^2+1}$. Finally, evaluating (5.34) at $y = y_*$ results in (5.43). ■

Remark 5.4 (Original variables η, β, γ, and $\widehat{\mathcal{L}}$)**.** *Restated in the original variables used to define x, y, and $\mathcal{A}$ in (5.28), Lemma 5.3 states that when $\widehat{\mathcal{L}}$ is symmetric negative semi-definite, $\widetilde{\kappa}(\mathcal{P}_\gamma)$ is minimized over $\gamma \in (0, \infty)$ at $\gamma = \gamma_*$, where*

$$\gamma_* = \sqrt{\eta^2 + \beta^2}, \tag{5.45}$$

and its associated minimum value is

$$\widetilde{\kappa}(\mathcal{P}_{\gamma_*}) = 2\left(\frac{\eta^2}{\beta^2} - \frac{\eta}{\beta}\sqrt{1+\frac{\eta^2}{\beta^2}} + 1\right). \tag{5.46}$$

Furthermore, it is interesting to consider preconditioning with the naive choice of $\gamma = \eta$, for which the condition number may be evaluated using (5.34) to give

$$\widetilde{\kappa}(\mathcal{P}_\eta) = \widetilde{\kappa}(\mathcal{P}_x) = 1 + \frac{1}{x^2} = 1 + \frac{\beta^2}{\eta^2}. \tag{5.47}$$

As discussed previously, the result of Lemma 5.2 and Lemma 5.3 is essentially the same as that of [4, Proposition 3.3]. Using the notation that $\eta \mapsto \alpha$ and $\gamma \mapsto \mu$, one can see that (5.46) is equivalent to [4, (3.24)].

From Remark 5.4 it is apparent that the condition number bounds for a fixed γ do not depend on η and β independently but on their ratio; indeed, this trend will continue throughout the rest of this chapter, where bounds will typically depend on β^2/η^2. It is therefore pertinent to consider the size of this ratio for FIRK methods of interest. Table 5.1 provides values for some Gauss, Radau IIA, and Lobatto IIIC schemes that will be used in the numerical results of Section 5.5. Clearly $\max_i \beta_i^2/\eta_i^2$ for a given family of FIRK schemes tends to increase with the number of stages/order, but importantly, it seems not to grow rapidly, and remains $\mathcal{O}(1)$ for a moderate number of stages. Nonetheless, there is likely to be a degradation in Krylov convergence as the number of stages/order of the FIRK method is increased due to there being an increase in values of $\max_i \beta_i^2/\eta_i^2$. Furthermore, for FIRK schemes with multiple complex-conjugate pairs of eigenvalues (e.g., the 4- and 5-stage methods in Table 5.1), it can be anticipated that the preconditioned Krylov solver will not perform equally well on all of the different pairs, since those associated with larger ratios of β_i^2/η_i^2 will converge more slowly.

A plot of the condition number bound $\widetilde{\kappa}(\mathcal{P}_\gamma)$ as a function of $\frac{\beta^2}{\eta^2}$ is shown in the left-hand side of Figure 5.1 for the optimal choice of $\gamma = \gamma_*$ and the naive choice of $\gamma = \eta$. Clearly

TABLE 5.1: Approximate values of β_i^2/η_i^2, where $\{\eta_i \pm \mathrm{i}\beta_i\}$ are the complex-conjugate eigenvalue pairs of $A_0^{-1} \in \mathbb{R}^{s\times s}$, for s-stage Gauss, Radau IIA, and Lobatto IIIC methods with $s \in \{2,3,4,5\}$. Gauss methods have an order of accuracy equal to $2s$, Radau IIA methods $2s-1$, and Lobatto IIIC methods $2s-2$. A_0^{-1} has $\frac{s}{2}$ complex-conjugate eigenvalue pairs when s is even, and $\frac{s-1}{2}$ when s is odd.

s	2	3	4		5	
	β_1^2/η_1^2	β_1^2/η_1^2	β_1^2/η_1^2	β_2^2/η_2^2	β_1^2/η_1^2	β_2^2/η_2^2
Gauss	0.33	0.91	0.09	1.60	0.27	2.36
Radau IIA	0.50	1.29	0.11	2.21	0.32	3.20
Lobatto IIIC	1.00	2.21	0.13	3.51	0.38	4.88

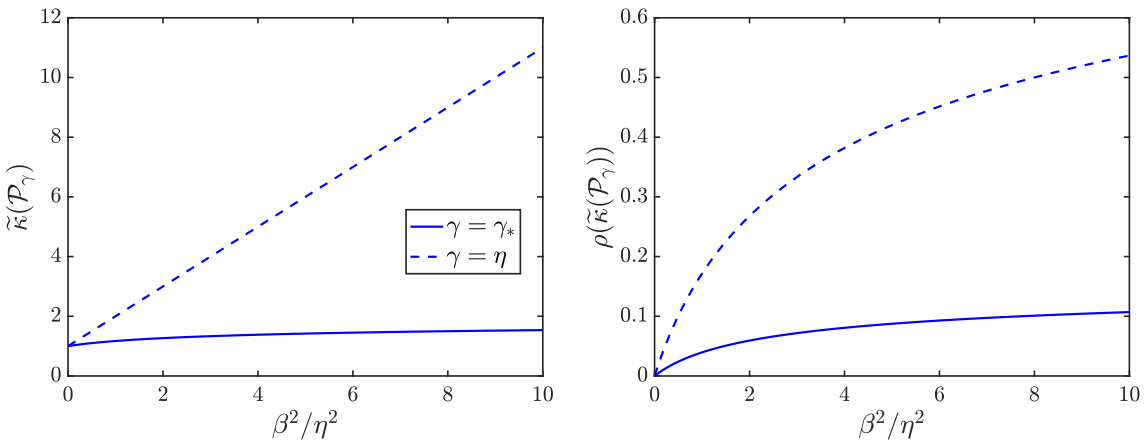

FIGURE 5.1: *Linear FIRK:* Eigenvalue-based analysis when $\widehat{\mathcal{L}}$ is symmetric negative semi-definite. *Left:* Condition number $\widetilde{\kappa}(\mathcal{P}_\gamma)$ (5.32) for the optimal choice of $\gamma = \gamma_*$ (see (5.46)), and the naive choice of $\gamma = \eta$ (see (5.47)). *Right:* Associated upper bounds on the CG convergence factor ρ (5.48).

the optimal choice leads to a much smaller condition number that is bounded with respect to $\frac{\beta^2}{\eta^2}$. In fact, in [4] it was shown that $\widetilde{\kappa}(\mathcal{P}_{\gamma_*}) \leq 2$.

As mentioned at the end of Section 5.2.1, the preconditioned operator $\mathcal{P}_\gamma$ (5.27) is symmetric positive definite when $\widehat{\mathcal{L}}$ is symmetric negative semi-definite, and, so, the associated linear system may be solved with CG which is more efficient for such problems than GMRES, for example. It is therefore of interest to develop CG convergence bounds based on the above condition number bound. To this end, if $\boldsymbol{e}_k$ denotes the algebraic error after k preconditioned CG iterations, then it may be bounded by [82, (6.128)]

$$\|\boldsymbol{e}_k\|_{\mathcal{P}_\gamma} \leq 2\big[\rho(\kappa(\mathcal{P}_\gamma))\big]^k \|\boldsymbol{e}_0\|_{\mathcal{P}_\gamma} \leq 2\big[\rho(\widetilde{\kappa}(\mathcal{P}_\gamma))\big]^k \|\boldsymbol{e}_0\|_{\mathcal{P}_\gamma}, \quad \rho(z) := \frac{\sqrt{z}-1}{\sqrt{z}+1}, \tag{5.48}$$

where $\|\cdot\|_{\mathcal{P}_\gamma} = \sqrt{\langle \mathcal{P}_\gamma \cdot, \cdot \rangle}$ denotes the norm defined by the SPD operator $\mathcal{P}_\gamma$. The convergence factor $\rho(\widetilde{\kappa}(\mathcal{P}_\gamma))$ is plotted in the right panel of Figure 5.1 for the specific cases of $\gamma = \gamma_*$ (5.46) and $\gamma = \eta$ (5.47). For the optimal choice of γ convergence is very fast and remains so with increasing $\frac{\beta^2}{\eta^2}$, while for the naive choice convergence is much slower and deteriorates with increasing $\frac{\beta^2}{\eta^2}$. This really highlights the importance of using γ_* over η. Moreover, since $\widetilde{\kappa}(\mathcal{P}_{\gamma_*}) \leq 2$, it follows from (5.48) that $\rho(\widetilde{\kappa}(\mathcal{P}_{\gamma_*})) \leq 1 - \frac{2}{\sqrt{2}+1} \lesssim 0.172$. Such a small convergence factor independent of spatial discretization parameters and time-step size confirm the robustness of the proposed preconditioner.

5.2.4 Eigenvalue analysis in the linear setting: The skew-symmetric case

Lemma 5.5 (Conditioning for skew-symmetric $\mathcal{A}$)**.** *Suppose that $\mathcal{A}$ is skew symmetric, then the square of the condition number $\widetilde{\kappa}(\mathcal{P}_y)$ in (5.32) that tightly bounds the condition*

number of the preconditioned operator $\mathcal{P}_y$ in (5.29) is equal to

- $x \in (0,1]$:

$$\widetilde{\kappa}^2(\mathcal{P}_y) = \begin{cases} \dfrac{\widehat{\mathcal{F}}_y(0)}{\widehat{\mathcal{F}}_y(\pm\mu_*)}, & y \in \left(0, \sqrt{x^2+1}\right], \\ \dfrac{1}{\widehat{\mathcal{F}}_y(\pm\mu_*)}, & y \in \left[\sqrt{x^2+1}, \infty\right), \end{cases} \tag{5.49}$$

- $x \in (1,\infty)$:

$$\widetilde{\kappa}^2(\mathcal{P}_y) = \begin{cases} \widehat{\mathcal{F}}_y(0), & y \in \left(0, \sqrt{x^2-1}\right], \\ \dfrac{\widehat{\mathcal{F}}_y(0)}{\widehat{\mathcal{F}}_y(\pm\mu_*)}, & y \in \left[\sqrt{x^2-1}, \sqrt{x^2+1}\right], \\ \dfrac{1}{\widehat{\mathcal{F}}_y(\pm\mu_*)}, & y \in \left[\sqrt{x^2+1}, \frac{x^2+1}{\sqrt{x^2-1}}\right], \\ \dfrac{1}{\widehat{\mathcal{F}}_y(0)}, & y \in \left[\frac{x^2+1}{\sqrt{x^2-1}}, \infty\right), \end{cases} \tag{5.50}$$

where

$$\widehat{\mathcal{F}}_y(0) = \left(\frac{x^2+1}{y^2}\right)^2, \quad \widehat{\mathcal{F}}_y(\pm\mu_*) = \frac{4x^2}{y^4 + 2(1-x^2)y^2 + (x^2+1)^2}. \tag{5.51}$$

Proof. In this instance the eigenvalues $\mathcal{F}_y(\lambda)$ of $\mathcal{P}_y$ are complex because the eigenvalues λ of $-\mathcal{A}$ are imaginary. For this reason, it becomes simpler to consider the extrema of $|\mathcal{F}_y(\lambda)|^2$ rather than $|\mathcal{F}_y(\lambda)|$ and hence the square of condition number (5.32) rather than the condition number itself. To this end, let us define the squared magnitude of $\mathcal{F}_y$ as the function

$$\widehat{\mathcal{F}}_y(\mu) := |\mathcal{F}_y(\mathrm{i}\mu)|^2 = \frac{\mu^4 + 2(x^2-1)\mu^2 + (x^2+1)^2}{(y^2+\mu^2)^2}, \quad \mu \in (-\infty, \infty), \tag{5.52}$$

where the eigenvalues of $-\mathcal{A}$ have been parametrized as $\lambda = \mathrm{i}\mu$ with $\mu \in \mathbb{R}$. This function is symmetric about $\mu = 0$, so it is sufficient to analyze its behaviour on $\mu \in [0, \infty)$.

Global extrema of $\widehat{\mathcal{F}}_y$ with respect to μ may either occur at domain boundaries or where its derivative vanishes. Considering first the boundary, one has

$$\lim_{\mu\to\infty} \widehat{\mathcal{F}}_y(\mu) = \lim_{\mu\to\infty} \frac{1 + 2\frac{x^2-1}{\mu^2} + \frac{(x^2+1)^2}{\mu^4}}{1 + 2\frac{y^2}{\mu^2} + \frac{y^4}{\mu^4}} = \begin{cases} 1^+, & a \le 0, \\ 1^-, & a > 0, \end{cases} \qquad \text{where} \quad a := 1 - x^2 + y^2. \tag{5.53}$$

This means that the derivative of $\widehat{\mathcal{F}}_y(\mu)$ satisfies the following bounds: $\lim_{\mu\to\infty} \frac{\mathrm{d}\widehat{\mathcal{F}}_y}{\mathrm{d}\mu} > 0$ if $a > 0$, and $\lim_{\mu\to\infty} \frac{\mathrm{d}\widehat{\mathcal{F}}_y}{\mathrm{d}\mu} < 0$ if $a \leq 0$. The reason for also considering here the sign of the derivative will become clear shortly. Next consider stationary points of $\widehat{\mathcal{F}}_y$. The derivative of (5.52) can be expressed as

$$\frac{\mathrm{d}\,\widehat{\mathcal{F}}_y}{\mathrm{d}\mu} = 4\mu \frac{a\mu^2 - c}{(\mu^2 + y^2)^3}, \qquad \text{where} \quad c := (1 + x^2)^2 + (1 - x^2)y^2. \tag{5.54}$$

Clearly there are at most two points where the derivative vanishes on $[0, \infty)$:

$$\mu = 0, \quad \mu = \mu_* := \sqrt{\frac{c}{a}}. \tag{5.55}$$

The stationary point μ_* is not defined when $a = 0$, and is to be discarded when $\mathrm{sign}(a) \neq \mathrm{sign}(c)$ since it would no longer be real.

Let us further examine the stationary point $\mu = 0$. To this end, applying a Taylor series expansion to (5.54) about $\mu = 0$ gives

$$\frac{\mathrm{d}\,\widehat{\mathcal{F}}_y}{\mathrm{d}\mu} = \frac{4\mu}{y^6}\left[-c + \left(a + \frac{3c}{y^2}\right)\mu^2 + \mathcal{O}\big(\mu^4\big)\right]. \tag{5.56}$$

Clearly when $c > 0$, $\frac{\mathrm{d}\,\widehat{\mathcal{F}}_y}{\mathrm{d}\mu} < 0$ at $\mu = 0^+$ and thus $\mu = 0$ is a local maximum ($\widehat{\mathcal{F}}_y$ is decreasing at $\mu = 0^+$ if its derivative is negative there). Conversely, when $c < 0$, $\frac{\mathrm{d}\,\widehat{\mathcal{F}}_y}{\mathrm{d}\mu} > 0$ at $\mu = 0^+$ and thus $\mu = 0$ is a local minimum. Calculations below will show that $a > 0$ whenever $c = 0$, and thus $\mu = 0$ is also a local minimum when $c = 0$. This information combined with the fact that there is at most one stationary point (i.e., μ_*) on $(0, \infty)$, and that the sign of the derivative of $\widehat{\mathcal{F}}_y$ is known as $\mu \to \infty$ from (5.53) allows us to classify the global extrema of $\widehat{\mathcal{F}}_y$ as follows.

1. If $c \leq 0$ then $\mu = 0$ is a local minimum so that $\widehat{\mathcal{F}}_y$ is increasing at $\mu = 0^+$
   - If $a > 0$ then $\widehat{\mathcal{F}}_y$ is increasing as $\mu \to \infty$, so there cannot be a turning point on $(0, \infty)$. It must be the case that μ_* is imaginary (or possibly a point of inflection), $\widehat{\mathcal{F}}_y(0)$ is the global minimum, and $\widehat{\mathcal{F}}_y(\infty)$ is the global maximum.
   - If $a \leq 0$ then $\widehat{\mathcal{F}}_y$ is decreasing as $\mu \to \infty$, so there has to be a local (and global) maximum on $(0, \infty)$. It must be the case that the global minimum is the $\min(\widehat{\mathcal{F}}_y(0), \widehat{\mathcal{F}}_y(\infty))$, and the global maximum is $\widehat{\mathcal{F}}_y(\mu_*)$.
2. If $c > 0$ then $\mu = 0$ is a local maximum so that $\widehat{\mathcal{F}}_y$ is decreasing at $\mu = 0^+$
   - If $a \leq 0$ then $\widehat{\mathcal{F}}_y$ is decreasing as $\mu \to \infty$, so there cannot be a turning point on $(0, \infty)$. It must be the case that μ_* is imaginary (or possibly a point of inflection), $\widehat{\mathcal{F}}_y(\infty)$ is the global minimum, and $\widehat{\mathcal{F}}_y(0)$ is the global maximum.

- If $a > 0$ then $\widehat{\mathcal{F}}_y$ is increasing as $\mu \to \infty$, there has to be a local (and global) minimum on $(0, \infty)$. The global minimum is $\widehat{\mathcal{F}}_y(\mu_*)$, and the global maximum is $\max(\widehat{\mathcal{F}}_y(0), \widehat{\mathcal{F}}_y(\infty))$.

This information is conveniently summarized in Table 5.2.

TABLE 5.2: Classification of global extrema of $\widehat{\mathcal{F}}_y$ with respect to signs of coefficients a and c.

	global minimum	global maximum
$c \leq 0, a > 0$	$\widehat{\mathcal{F}}_y(0)$	$\widehat{\mathcal{F}}_y(\infty)$
$c \leq 0, a \leq 0$	$\min(\widehat{\mathcal{F}}_y(0), \widehat{\mathcal{F}}_y(\infty))$	$\widehat{\mathcal{F}}_y(\mu_*)$
$c > 0, a \leq 0$	$\widehat{\mathcal{F}}_y(\infty)$	$\widehat{\mathcal{F}}_y(0)$
$c > 0, a > 0$	$\widehat{\mathcal{F}}_y(\mu_*)$	$\max(\widehat{\mathcal{F}}_y(0), \widehat{\mathcal{F}}_y(\infty))$

In order to further classify the extrema with respect to the parameters x and y, it is necessary to calculate the signs of the constants a (5.53) and c (5.54). To this end, consider the following factorisations of c

$$c = \begin{cases} (1-x^2)\left(y^2 + \dfrac{(x^2+1)^2}{1-x^2}\right), & x \in (0,\infty) \setminus 1, \quad (5.57) \\ (x^2-1)\left(\dfrac{x^2+1}{\sqrt{x^2-1}} + y\right)\left(\dfrac{x^2+1}{\sqrt{x^2-1}} - y\right), & x \in (1,\infty). \quad (5.58) \end{cases}$$

From (5.54) and (5.57), $c > 0$ for $x \in (0, 1]$; it is also clear that $a > 0$ whenever $x \in (0, 1]$. Using the information from Table 5.2 when a and c are positive yields for $x \in (0, 1]$

$$\max_{\mu \in (-\infty,\infty)} \widehat{\mathcal{F}}_y(\mu) = \begin{cases} \widehat{\mathcal{F}}_y(0), & y \in \left(0, \sqrt{x^2+1}\right], \\ \widehat{\mathcal{F}}_y(\pm\infty), & y \in \left[\sqrt{x^2+1}, \infty\right), \end{cases} \tag{5.59}$$

$$\min_{\mu \in (-\infty,\infty)} \widehat{\mathcal{F}}_y(\mu) = \widehat{\mathcal{F}}_y(\pm\mu_*). \tag{5.60}$$

Taking the ratio of the maximum to the minimum and substituting $\widehat{\mathcal{F}}_y(\pm\infty) = 1$ gives the condition number (5.49).

Now consider the case of $x \in (1, \infty)$ which requires more careful attention. From (5.58), the sign of c is equal to that of $\left(\frac{x^2+1}{\sqrt{x^2-1}} - y\right)$ since the first two factors are positive. The constant a is non-positive for $y \in (0, \sqrt{x^2-1}]$, and positive for $y \in (\sqrt{x^2-1}, \infty)$. Since $\sqrt{x^2-1} < \sqrt{x^2+1} < \frac{x^2+1}{\sqrt{x^2-1}}$, the single sign change of a occurs on the interval where c is

positive. Piecing this together, the signs of the constants when $x \in (1, \infty)$ are

$$a \leq 0, \quad c > 0, \quad y \in \left(0, \sqrt{x^2 - 1}\right], \tag{5.61}$$

$$a > 0, \quad c > 0, \quad y \in \left(\sqrt{x^2 - 1}, \frac{x^2 + 1}{\sqrt{x^2 - 1}}\right], \tag{5.62}$$

$$a > 0, \quad c \leq 0, \quad y \in \left(\frac{x^2 + 1}{\sqrt{x^2 - 1}}, \infty\right). \tag{5.63}$$

Appealing to Table 5.2, the extrema of (5.52) when $x \in (1, \infty)$ can be calculated to be

$$\max_{\mu \in (-\infty, \infty)} \widehat{\mathcal{F}}_y(\mu) = \begin{cases} \widehat{\mathcal{F}}_y(0), & y \in \left(0, \sqrt{x^2 + 1}\right], \\ \widehat{\mathcal{F}}_y(\pm\infty), & y \in \left[\sqrt{x^2 + 1}, \infty\right), \end{cases} \tag{5.64}$$

$$\min_{\mu \in (-\infty, \infty)} \widehat{\mathcal{F}}_y(\mu) = \begin{cases} \widehat{\mathcal{F}}_y(\pm\infty), & y \in \left(0, \sqrt{x^2 - 1}\right], \\ \widehat{\mathcal{F}}_y(\pm\mu_*), & y \in \left(\sqrt{x^2 - 1}, \frac{x^2+1}{\sqrt{x^2-1}}\right], \\ \widehat{\mathcal{F}}_y(0), & y \in \left(\frac{x^2+1}{\sqrt{x^2-1}}, \infty\right). \end{cases} \tag{5.65}$$

Taking the ratio of the maximum to the minimum and substituting $\widehat{\mathcal{F}}_y(\pm\infty) = 1$ yields (5.50). ■

Lemma 5.6 (Optimal preconditioning for skew-symmetric $\mathcal{A}$)**.** *The tight condition number bound $\widetilde{\kappa}(\mathcal{P}_y)$ for which the square is given by (5.49) and (5.50) in Lemma 5.5 is minimized over $y \in (0, \infty)$ at $y = y_*$, where*

$$y_* = \sqrt{x^2 + 1}. \tag{5.66}$$

Moreover, the minimum value of $\widetilde{\kappa}(\mathcal{P}_y)$ is

$$\widetilde{\kappa}(\mathcal{P}_{y_*}) = \sqrt{1 + \frac{1}{x^2}}. \tag{5.67}$$

Proof. Let us first consider the bound (5.49) which applies for $x \in (0, 1]$. To this end, consider the derivatives of the functions that piecewise define it, which can be conveniently expressed as

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{\widehat{\mathcal{F}}_y(0)}{\widehat{\mathcal{F}}_y(\pm\mu_*)}\right) = -\frac{1}{y^5}\left(\frac{x^2 + 1}{x}\right)^2 c(x, y), \tag{5.68}$$

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{1}{\widehat{\mathcal{F}}_y(\pm\mu_*)}\right) = \frac{y}{x^2} a(x, y), \tag{5.69}$$

where the (now) functions a and c are defined in (5.53) and (5.54), respectively. Recall from the proof of Lemma 5.5 that a and c are always positive when $x \in (0, 1]$. Derivative (5.68) is therefore always negative when $y > 0$, and thus $\widetilde{\kappa}^2(\mathcal{P}_y)$ in (5.49) is strictly decreasing on $y \in \left(0, \sqrt{x^2+1}\right)$ where it is equal to $\frac{\widehat{\mathcal{F}}_y(0)}{\widehat{\mathcal{F}}_y(\pm\mu_*)}$. Conversely, derivative (5.69) is always positive for $y > 0$, which means that $\widetilde{\kappa}^2(\mathcal{P}_y)$ is strictly increasing for $y \in \left(\sqrt{x^2+1}, \infty\right)$ where it is equal to $\frac{1}{\widehat{\mathcal{F}}_y(\pm\mu_*)}$. By the continuity of $\widetilde{\kappa}^2(\mathcal{P}_y)$, it must be the case that its global minimum occurs at the interface $y = \sqrt{x^2+1}$.

Now consider the bound $\widetilde{\kappa}^2(\mathcal{P}_y)$ in (5.50) that applies for $x \in (1, \infty)$. From (5.62), $c > 0$ when $y \in \left(\sqrt{x^2-1}, \frac{x^2+1}{\sqrt{x^2-1}}\right)$, so $\widetilde{\kappa}^2(\mathcal{P}_y)$ must be strictly decreasing for $y \in \left(\sqrt{x^2-1}, \sqrt{x^2+1}\right)$ because its derivative is negative (5.68) ($\widetilde{\kappa}^2(\mathcal{P}_y) = \frac{\widehat{\mathcal{F}}_y(0)}{\widehat{\mathcal{F}}_y(\pm\mu_*)}$ on this interval). Conversely, from (5.62) and (5.63) $a > 0$ for any $y > \sqrt{x^2-1}$, so $\widetilde{\kappa}^2(\mathcal{P}_y)$ must be strictly increasing for $y \in \left(\sqrt{x^2+1}, \frac{x^2+1}{\sqrt{x^2-1}}\right)$ because its derivative is negative (5.69) ($\widetilde{\kappa}^2(\mathcal{P}_y) = \frac{1}{\widehat{\mathcal{F}}_y(\pm\mu_*)}$ on this interval). By continuity of the function, $\widetilde{\kappa}^2(\mathcal{P}_y)$ must have a local minimum at the interface $y = \sqrt{x^2+1}$. To see that this local minimum is indeed a global minimum, observe that $\widehat{\mathcal{F}}_y(0) = \left(\frac{x^2+1}{y}\right)^2$ (5.51) is a strictly decreasing function for $y > 0$, and therefore $\widetilde{\kappa}^2(\mathcal{P}_y)$ is strictly decreasing on $y \in \left(0, \sqrt{x^2-1}\right)$ where it is equal to $\widehat{\mathcal{F}}_y(0)$. Conversely, $\widetilde{\kappa}^2(\mathcal{P}_y)$ is strictly increasing on $y \in \left(\frac{x^2+1}{\sqrt{x^2-1}}, \infty\right)$ where it is equal to $\frac{1}{\widehat{\mathcal{F}}_y(0)}$.

Thus, the minimum value of $\widetilde{\kappa}^2(\mathcal{P}_y)$ for both $x \in (0, 1]$ (5.49) and $x \in (1, \infty)$ (5.50) is reached when $y = \sqrt{x^2+1}$. Note that $y = \sqrt{x^2+1}$ is also the minimizer of $\widetilde{\kappa}(\mathcal{P}_y)$ itself since the function is non-negative.

Finally, the minimum value of $\widetilde{\kappa}(\mathcal{P}_y)$ given by (5.67) follows from evaluating either (5.49) or (5.50) at $y = y_*$ and then taking the square root. ■

Remark 5.7 (Original variables η, β, γ, and $\widehat{\mathcal{L}}$)**.** *Restated in the original variables used to define x, y, and $\mathcal{A}$ in (5.28), Lemma 5.6 states that when $\widehat{\mathcal{L}}$ is skew symmetric, $\widetilde{\kappa}(\mathcal{P}_\gamma)$ is minimized over $\gamma \in (0, \infty)$ at $\gamma = \gamma_*$, where*

$$\gamma_* = \sqrt{\eta^2 + \beta^2}, \tag{5.70}$$

and the associated minimum value is

$$\widetilde{\kappa}(\mathcal{P}_{\gamma_*}) = \sqrt{1 + \frac{\beta^2}{\eta^2}}. \tag{5.71}$$

Furthermore, it is interesting to consider preconditioning with the naive choice of $\gamma = \eta$, for which the condition number may be evaluated using (5.49) or (5.50) to give

$$\widetilde{\kappa}(\mathcal{P}_\eta) = \widetilde{\kappa}(\mathcal{P}_x) = \sqrt{\frac{\widehat{\mathcal{F}}_x(0)}{\widehat{\mathcal{F}}_x(\pm\mu_*)}} = \frac{1}{2}\sqrt{4 + \frac{\beta^2}{\eta^2}}\left(1 + \frac{\beta^2}{\eta^2}\right). \tag{5.72}$$

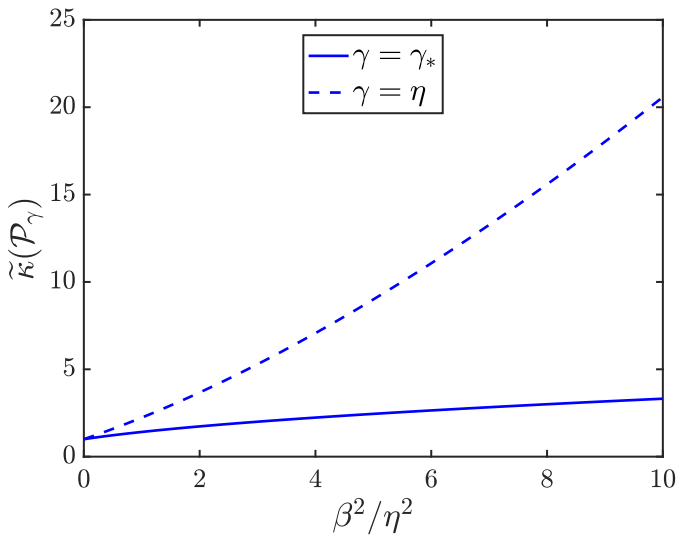

FIGURE 5.2: *Linear FIRK:* Eigenvalue-based analysis when $\widehat{\mathcal{L}}$ is skew symmetric. The condition number bound $\widetilde{\kappa}(\mathcal{P}_\gamma)$ is shown for the optimal choice of $\gamma = \gamma_*$ (see (5.71)), and the naive choice of $\gamma = \eta$ (see (5.72)).

The condition number bound $\widetilde{\kappa}(\mathcal{P}_\gamma)$ is shown in Figure 5.2 for the optimal choice of $\gamma = \gamma_*$ and the naive choice of $\gamma = \eta$. Clearly the optimal choice leads to a much smaller condition number. Notice, however, that even for the optimal choice of γ_* there is slow growth in the condition number with $\frac{\beta^2}{\eta^2}$; this is qualitatively different to when $\widehat{\mathcal{L}}$ is symmetric negative semi-definite, where the condition number is bounded by two (see Figure 5.1). This is yet another manifestation of the well-known fact that iteratively solving highly non-symmetric linear systems is often much more challenging than their symmetric counterparts. Nonetheless, the condition number for γ_* shown in Figure 5.2 remains $\mathcal{O}(1)$ for values of $\frac{\beta^2}{\eta^2}$ that are associated with very high-order FIRK methods (for example, 8th, 9th, and 10th order; see Table 5.1), which shows the efficacy of the proposed preconditioner, even for highly non-symmetric $\widehat{\mathcal{L}}$.

Finally, notice that the optimal value of γ_* (5.70) when $\widehat{\mathcal{L}}$ is skew symmetric is the same optimal value of γ_* as when $\widehat{\mathcal{L}}$ is symmetric negative semi-definite (5.45). This suggests that perhaps the value of $\sqrt{\eta^2 + \beta^2}$ is optimal for more general $\widehat{\mathcal{L}}$. This is indeed shown to be the case in Section 5.4 where the condition number of the preconditioned operator is re-examined without assumptions on the symmetry or normality of $\widehat{\mathcal{L}}$.

5.3 The nonlinear setting

This section considers the solution of the nonlinear stage equations (5.12) that result from the application of FIRK methods to nonlinear ODEs (5.2). First, Section 5.3.1 proposes a simple Newton algorithm for the solution of these nonlinear equations. A brief outline and discussion of the remainder of the section is then given in Section 5.3.2, and an eigenvalue-based analysis of the linear preconditioner in the simple Newton method is the subject of Section 5.3.3.[7]

5.3.1 The algorithm: Simple Newton with real Schur decomposition

As discussed in Section 5.1.2, the nonlinear system of algebraic equations $\mathcal{G}(\boldsymbol{w}) = 0$ given by (5.12) that define the FIRK stage vectors $\boldsymbol{w}$ is solved using a Newton-like method. More specifically, recall that at each Newton-like iteration a linear system of the form

$$\mathcal{J}\big(\boldsymbol{w}^{(k)}\big)\boldsymbol{\Delta}^{(k+1)} = -\mathcal{G}\big(\boldsymbol{w}^{(k)}\big), \quad \boldsymbol{\Delta}^{(k+1)} := \boldsymbol{w}^{(k+1)} - \boldsymbol{w}^{(k)}, \tag{5.73}$$

is (approximately) solved to produce a new nonlinear iterate $\boldsymbol{w}^{(k+1)} \approx \boldsymbol{w}$ based on the previous nonlinear iterate $\boldsymbol{w}^{(k)}$, where $\mathcal{J}\big(\boldsymbol{w}^{(k)}\big)$ is some approximation to the true Jacobian of $\mathcal{G}\big(\boldsymbol{w}^{(k)}\big)$. Traditionally in the FIRK literature, the approximate Jacobian $\mathcal{J}$ taken to have a Kronecker-product structure, resulting in a so-called simple Newton method [15, 8]. This is the approach we consider here also. Specifically, recalling in the nonlinear context that $(\mathcal{L}_i)_{i=1}^s$ are linearizations of the nonlinear operators $(\mathcal{N}_i)_{i=1}^s$, we define $\mathcal{J}$ as the approximation to $\mathcal{G}\big(\boldsymbol{w}^{(k)}\big)$ in which $(\mathcal{L}_i)_{i=1}^s$ are replaced by a constant $\mathcal{L}$:

$$\mathcal{G}'(\boldsymbol{w}) = A_0^{-1} \otimes I_N - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} \approx \mathcal{J} := A_0^{-1} \otimes I_N - \delta t I_s \otimes \mathcal{L}\,. \tag{5.74}$$

A primary motivation for considering a simple Newton algorithm is that solving a linear system with matrix $\mathcal{J}$ in (5.74) can be decomposed into smaller problems by means of a similarity transform of A_0^{-1}. The algorithm has several drawbacks, however, with one being that only linear convergence can be achieved rather than the quadratic rate of an exact Newton method.

So, let us now discuss the numerical solution of linear system (5.73) when $\mathcal{J}$ is given as in (5.74). As for the linear FIRK algorithm (see Section 5.2.1), we introduce the shorthand

[7] The interested reader is directed to our article [90] for a discussion on closely related, but more sophisticated Newton-like methods than the simple Newton method presented here.

$\widehat{\mathcal{L}} := \delta t\,\mathcal{L}$. Next, let $Q_0 R_0 Q_0^\top = A_0^{-1}$ be the real Schur decomposition of A_0^{-1}, where $Q_0 \in \mathbb{R}^{s\times s}$ is orthogonal (i.e., $Q_0 Q_0^\top = I$), and $R_0 \in \mathbb{R}^{s\times s}$ is block upper triangular. Specifically, each block of R_0 either (i) is equal to a real eigenvalue ζ_i of A_0^{-1}, or (ii) corresponds to a complex-conjugate eigenvalue pair $\eta_i \pm \mathrm{i}\beta_i$ of A_0^{-1} and takes the form of a 2×2 block $\begin{bmatrix} \eta_i & \phi_i \\ -\beta_i^2/\phi_i & \eta_i \end{bmatrix}$, for some constant $\phi_i \in \mathbb{R}$. The approximate linearized system (5.73) may then be transformed as

$$\big(Q_0 R_0 Q_0^\top \otimes I_N - I_s \otimes \widehat{\mathcal{L}}\big)\boldsymbol{\Delta}^{(k+1)} = -\,\mathcal{G}\left(\boldsymbol{w}^{(k)}\right), \tag{5.75}$$

$$\big(Q_0 \otimes I_N\big)\big(R_0 \otimes I_N - I_s \otimes \widehat{\mathcal{L}}\big)\big(Q_0^\top \otimes I_N\big)\boldsymbol{\Delta}^{(k+1)} = -\,\mathcal{G}\left(\boldsymbol{w}^{(k)}\right), \tag{5.76}$$

$$\big(R_0 \otimes I_N - I_s \otimes \widehat{\mathcal{L}}\big)\big[\big(Q_0^\top \otimes I_N\big)\boldsymbol{\Delta}^{(k+1)}\big] = -\big(Q_0^\top \otimes I_N\big)\,\mathcal{G}\left(\boldsymbol{w}^{(k)}\right). \tag{5.77}$$

Due to the block upper triangular structure of R_0, the system matrix in (5.77) is block upper triangular. This system can therefore be solved by block backward substitution, which requires inverting each of the diagonal blocks. Diagonal blocks corresponding to real-valued eigenvalues ζ_i of A_0^{-1} take the form $(\zeta_i I - \widehat{\mathcal{L}})$, and may be solved using standard preconditioning techniques (note this is equivalent to the linear operator that arises in an implicit Euler discretization). Conversely, blocks corresponding to complex-conjugate eigenvalue pairs $\eta_i \pm \mathrm{i}\beta_i$ of A_0^{-1} are 2×2 block matrices of the form

$$\begin{bmatrix} \eta_i I - \widehat{\mathcal{L}} & \phi_i I \\ -\frac{\beta_i^2}{\phi_i} I & \eta_i I - \widehat{\mathcal{L}} \end{bmatrix}. \tag{5.78}$$

We propose to invert blocks (5.78) using Krylov methods (e.g., GMRES) with block lower-triangular preconditioners of the form

$$\begin{bmatrix} \eta_i I - \widehat{\mathcal{L}} & 0 \\ -\frac{\beta_i^2}{\phi_i} I & \widehat{S}_{\gamma_i} \end{bmatrix}^{-1}, \tag{5.79}$$

where $\widehat{S}_{\gamma_i}$ is some approximation to the Schur complement S_i of (5.78),

$$S_i = \eta_i I - \widehat{\mathcal{L}} + \beta_i^2 (\eta_i I - \widehat{\mathcal{L}})^{-1}. \tag{5.80}$$

Based on the structure of (5.80) when $\beta_i \approx 0$, preconditioners of the following form are considered:

$$\widehat{S}_{\gamma_i} = \gamma_i I - \widehat{\mathcal{L}}, \quad \gamma_i \in (0, \infty) \tag{5.81}$$

with γ_i a free parameter. Notice that if $\beta_i \approx 0$, then $\gamma_i = \eta_i$ is likely a good choice, but as the forthcoming analysis will demonstrate, one can make a significantly better choice

than $\gamma_i = \eta_i$ for general $\beta_i \not\approx 0$.

Moving forward, let us drop the subscript i, and instead refer to an arbitrary complex-conjugate eigenvalue pair $\eta \pm \mathrm{i}\beta$. When applying GMRES to block 2×2 operators preconditioned with lower triangular preconditioners of the form (5.79), convergence is exactly defined by convergence of GMRES applied to the preconditioned Schur complement [93], which in this instance is

$$\mathcal{S}_\gamma := S\widehat{S}_\gamma^{-1} = \big[(\eta I - \widehat{\mathcal{L}}) + \beta^2(\eta I - \widehat{\mathcal{L}})^{-1}\big](\gamma I - \widehat{\mathcal{L}})^{-1}. \tag{5.82}$$

Notice that in applying the triangular preconditioner of (5.79), we must carry out linear solves for the matrices $\eta_i I - \widehat{\mathcal{L}}$ and $\widehat{S}_{\gamma_i}$. In practice, we do not perform these solves exactly, but instead approximate them using an inexpensive preconditioner, such as a single iteration of AMG, for example. However, the forthcoming analysis assumes that these solves are carried out exactly. This situation is analogous to that for the analysis we conducted previously for the linear problem (see Section 5.2.1).

5.3.2 Outline and assumptions for nonlinear eigenvalue analysis

The remainder of Section 5.3 is devoted to analysing the two-norm condition number of the preconditioned Schur complement (5.82), and thus the robustness of a preconditioned Krylov method applied to the larger 2×2 block system (5.78). In particular, Section 5.3.3 does so under the assumption that the spatial discretization $\mathcal{L}$ is symmetric negative semi-definite. A value of $\gamma = \gamma_*$ in the preconditioned Schur complement (5.82) is identified that minimizes a tight upper bound on its condition number. The associated condition number of $\mathcal{S}_{\gamma_*}$ is small but grows weakly with FIRK integration order; for example, it is less than two for Gauss, Radau IIA, and Lobatto IIIC schemes with up to five stages (see also 5.1). Note that no preconditioning analysis is presented for the case where $\mathcal{L}$ is skew symmetric. This is because eigenvalue analysis of the nonlinear algorithm for skew-symmetric $\mathcal{L}$ is significantly more complicated than it was in the linear setting (see Section 5.2.4), and because the theory in Section 5.4 covers general non-symmetric $\mathcal{L}$ (i.e., not only those that are skew symmetric).

The forthcoming analysis is similar in spirit to those of Sections 5.2.3 and 5.2.4 for the linear FIRK algorithm which is perhaps not surprising given that the preconditioned operator $\mathcal{P}_\gamma$ (5.27) in the linear setting has a closely related structure to that of the preconditioned Schur complement $\mathcal{S}_\gamma$ (5.82) (this connection will be discussed further in Section 5.4). In fact, the same scaled variables and style of condition number bound are now reintroduced here, so the reader is referred back to the discussion in Section 5.2.2 where they were

originally presented for further details. Recall the scaled variables

$$x := \frac{\eta}{\beta}, \quad y := \frac{\gamma}{\beta}, \quad \mathcal{A} := \frac{1}{\beta}\widehat{\mathcal{L}}. \tag{5.83}$$

Applying this change of variables to (5.82) yields the simpler preconditioned Schur complement

$$\mathcal{S}_y := \left[(xI - \mathcal{A}) + (xI - \mathcal{A})^{-1}\right](yI - \mathcal{A})^{-1}, \tag{5.84}$$

which has the same condition number as $\mathcal{S}_\gamma$. When $\mathcal{A}$ is symmetric negative semi-definite, the two-norm condition number $\kappa(\mathcal{S}_y)$ of (5.84) is given and tightly bounded by

$$\kappa(\mathcal{S}_y) = \frac{\max\limits_{\lambda\in\sigma(-\mathcal{A})} |\mathcal{F}_y(\lambda)|}{\min\limits_{\lambda\in\sigma(-\mathcal{A})} |\mathcal{F}_y(\lambda)|} \leq \frac{\max\limits_{\lambda\in[0,\infty)} |\mathcal{F}_y(\lambda)|}{\min\limits_{\lambda\in[0,\infty)} |\mathcal{F}_y(\lambda)|} =: \widetilde{\kappa}(\mathcal{S}_y), \tag{5.85}$$

where $\sigma(-\mathcal{A})$ denotes the spectrum of $-\mathcal{A}$, and the function $\mathcal{F}_y(\lambda)$ is the eigenvalue of $\mathcal{S}_y$ associated with an eigenvalue λ of $-\mathcal{A}$,

$$\mathcal{F}_y(\lambda) = \frac{(x+\lambda)^2 + 1}{(x+\lambda)(y+\lambda)}. \tag{5.86}$$

5.3.3 Eigenvalue analysis in the nonlinear setting: The symmetric definite case

Lemma 5.8 (Conditioning for symmetric negative semi-definite $\mathcal{A}$)**.** *Suppose that $\mathcal{A}$ is symmetric negative semi-definite, then the condition number $\widetilde{\kappa}(\mathcal{S}_y)$ in (5.85) that tightly bounds the condition number of the preconditioned Schur complement $\mathcal{S}_y$ is equal to*

- $x \in (0, 1]$:

$$\widetilde{\kappa}(\mathcal{S}_y) = \begin{cases} \mathcal{F}_y(0), & y \in (0, x], \\ \dfrac{\mathcal{F}_y(0)}{\mathcal{F}_y(\lambda_+)}, & y \in \left[x, x + \frac{1}{x}\right], \\ \dfrac{1}{\mathcal{F}_y(\lambda_+)}, & y \in \left[x + \frac{1}{x}, \infty\right], \end{cases} \tag{5.87}$$

- $x \in (1, \infty)$:

$$\widetilde{\kappa}(\mathcal{S}_y) = \begin{cases} \mathcal{F}_y(0), & y \in (0, x], \\ \dfrac{\mathcal{F}_y(0)}{\mathcal{F}_y(\lambda_+)}, & y \in \left[x, x + \frac{1}{x}\right], \\ \dfrac{1}{\mathcal{F}_y(\lambda_+)}, & y \in \left[x + \frac{1}{x}, x\frac{x^2+1}{x^2-1}\right], \\ \dfrac{1}{\mathcal{F}_y(0)}, & y \in \left[x\frac{x^2+1}{x^2-1}, \infty\right), \end{cases} \tag{5.88}$$

where

$$\mathcal{F}_y(0) = \frac{1}{y}\left(x + \frac{1}{x}\right), \quad \mathcal{F}_y(\lambda_+) = \frac{2}{1 + \sqrt{1 + (y - x)^2}}. \tag{5.89}$$

Proof. The bound $\widetilde{\kappa}(\mathcal{S}_y)$ is calculated by identifying the global extrema of $|\mathcal{F}_y(\lambda)|$ over $\lambda \in [0, \infty) =: \mathcal{I}$. Notice first from (5.86) that $\mathcal{F}_y(\lambda) > 0$ for any $\lambda \in \mathcal{I}$ since $x, y > 0$, and therefore $|\mathcal{F}_y(\lambda)| = \mathcal{F}_y(\lambda)$ (in other words, $\mathcal{S}_y$ is positive definite when $\mathcal{A}$ is symmetric negative semi-definite).

Global extrema may occur at the boundaries of $\mathcal{I}$, or at stationary points (where the derivative of $\mathcal{F}_y$ vanishes). Considering first the boundaries of $\mathcal{I}$, one has

$$\mathcal{F}_y(0) = \frac{1}{y}\left(x + \frac{1}{x}\right), \quad \lim_{\lambda\to\infty} \mathcal{F}_y(\lambda) = \lim_{\lambda\to\infty} \frac{1 + \frac{2x}{\lambda} + \frac{1+x^2}{\lambda^2}}{1 + \frac{x+y}{\lambda} + \frac{xy}{\lambda^2}} = \begin{cases} 1^+, & y \in (0, x], \\ 1^-, & y \in (x, \infty). \end{cases} \tag{5.90}$$

The latter result means that the derivative of $\mathcal{F}_y(\lambda)$ can be bounded as: $\lim_{\lambda\to\infty} \frac{\mathrm{d}\mathcal{F}_y}{\mathrm{d}\lambda} < 0$ if $y \in (0, x]$, and $\lim_{\lambda\to\infty} \frac{\mathrm{d}\mathcal{F}_y}{\mathrm{d}\lambda} > 0$ if $y \in (x, \infty)$. The reason for considering the sign of the derivative here will become clear shortly.

Now let us consider the derivative of $\mathcal{F}_y$, which may be expressed as

$$\frac{\mathrm{d}\mathcal{F}_y}{\mathrm{d}\lambda} = \frac{(y - x)\lambda^2 - 2(x^2 - xy + 1)\lambda - \left[x(x^2 + 1) + y(1 - x^2)\right]}{(x + \lambda)^2(y + \lambda)^2}. \tag{5.91}$$

Stationary points occur when the numerator of this function vanishes; solving this quadratic equation reveals the pair of solutions $\lambda = \lambda_\pm$,

$$\lambda_\pm = \frac{x^2 + 1 - xy \pm \sqrt{1 + (y - x)^2}}{y - x}, \quad y \in (0, \infty) \setminus x, \tag{5.92}$$

and the single solution $\lambda = -x \notin \mathcal{I}$ when $y = x$. Clearly the latter solution is not a stationary point of interest. It now needs to be determined when exactly $\lambda_\pm \in \mathcal{I}$. To this

end, consider evaluating the function at these stationary points, which yields

$$\mathcal{F}_y(\lambda_\pm) = \frac{2}{1 \pm \sqrt{1 + (y - x)^2}}, \quad y \in (0, \infty) \setminus x. \tag{5.93}$$

Clearly $\mathcal{F}_y(\lambda_-) < 0$, and thus it must be the case that $\lambda_- \notin \mathcal{I}$ for $y \in (0, \infty) \setminus x$ since $\mathcal{F}_y(\lambda) > 0 \, \forall \lambda \in \mathcal{I}$. Therefore, λ_- is to be discarded as stationary point of interest. Conversely, however, $\mathcal{F}_y(\lambda_+) > 0$ for $y \in (0, \infty) \setminus x$, but this is not a sufficient condition that $\lambda_+ \in \mathcal{I}$ since it can be the case that $\mathcal{F}_y$ (5.86) is positive for $\lambda < 0$. Indeed, further analysis reveals that λ_+ has sign changes on $y \in (0, \infty) \setminus x$, but this quickly becomes complicated, so let us pursue another path forward.

Keeping in mind that the sign of the derivative of $\mathcal{F}_y$ as $\lambda \to \infty$ is known from (5.90), and that there is at most one stationary point $\lambda = \lambda_+$ on $\mathcal{I}$, let us consider the slope of $\mathcal{F}_y$ as $\lambda \to 0^+$. From (5.91), the sign of the derivative can be determined as

- $x \in (0, 1]$:

$$\operatorname{sign}\left(\lim_{\lambda \to 0^+} \frac{\mathrm{d}\mathcal{F}_y}{\mathrm{d}\lambda}\right) = -1, \tag{5.94}$$

- $x \in (1, \infty)$:

$$\operatorname{sign}\left(\lim_{\lambda \to 0^+} \frac{\mathrm{d}\mathcal{F}_y}{\mathrm{d}\lambda}\right) = \begin{cases} -1, & y \in (0, \hat{y}(x)), \\ 0, & y = \hat{y}(x), \\ 1, & y \in (\hat{y}(x), \infty), \end{cases} \tag{5.95}$$

  where,

$$\hat{y}(x) := x\frac{x^2 + 1}{x^2 - 1} > x + \frac{1}{x} > x, \quad \text{when} \quad x > 1. \tag{5.96}$$

  Combined with the information above, this can now be used to classify the extrema of $\mathcal{F}_y$ as follows.

  1. $x \in (0, 1]$: $\mathcal{F}_y$ is increasing as $\lambda \to 0^+$.
     (a) $y \in (0, x]$: $\mathcal{F}_y$ is decreasing as $\lambda \to \infty$, so there cannot be a turning point on $(0, \infty)$. It must be that $\lambda_+ \notin \mathcal{I}$ (or it is possibly a point of inflection), $\mathcal{F}_y(\infty)$ is the global minimum, and $\mathcal{F}_y(0)$ is the global maximum.
     (b) $y \in (x, \infty)$: $\mathcal{F}_y$ is increasing as $\lambda \to \infty$, so there has to be a local (and global) minimum on $(0, \infty)$. It must be that $\mathcal{F}_y(\lambda_+)$ is the global minimum, and the global maximum is $\max(\mathcal{F}_y(0), \mathcal{F}_y(\infty))$.
  2. $x \in (1, \infty)$:

(a) $y \in (0, x]$: $\mathcal{F}_y$ is increasing as $\lambda \to 0^+$, and is decreasing as $\lambda \to \infty$, so there cannot be a turning point on $(0, \infty)$. It must be that $\lambda_+ \notin \mathcal{I}$ (or it is possibly a point of inflection), $\mathcal{F}_y(\infty)$ is the global minimum, and $\mathcal{F}_y(0)$ is the global maximum.

(b) $y \in (x, \hat{y}(x))$: $\mathcal{F}_y$ is increasing as $\lambda \to 0^+$, and is increasing as $\lambda \to \infty$, so there has to be a local (and global) minimum on $(0, \infty)$. It must be that $\mathcal{F}_y(\lambda_+)$ is the global minimum, and the global maximum is $\max(\mathcal{F}_y(0), \mathcal{F}_y(\infty))$.

(c) $y = \hat{y}(x)$: $\mathcal{F}_y$ is constant as $\lambda \to 0^+$, and is increasing as $\lambda \to \infty$, so there cannot be a turning point on $(0, \infty)$. It must be that $\lambda_+ \notin \mathcal{I}$ (or it is possibly a point of inflection), $\mathcal{F}_y(0)$ is the global minimum, and $\mathcal{F}_y(\infty)$ is the global maximum.

(d) $y \in (\hat{y}(x), \infty)$: $\mathcal{F}_y$ is increasing as $\lambda \to 0^+$, and is increasing as $\lambda \to \infty$, so there cannot be a turning point on $(0, \infty)$. It must be that $\lambda_+ \notin \mathcal{I}$ (or it is possibly a point of inflection), $\mathcal{F}_y(0)$ is the global minimum, and $\mathcal{F}_y(\infty)$ is the global maximum.

Combining all of this allows for the global maximum of $\mathcal{F}_y$ to be computed as

$$\max_{\lambda \in \mathcal{I}} \mathcal{F}_y(\lambda) = \begin{cases} \mathcal{F}_y(0), & y \in \left(0, x + \frac{1}{x}\right], \\ \mathcal{F}_y(\infty), & y \in \left[x + \frac{1}{x}, \infty\right), \end{cases} \tag{5.97}$$

and the global minimum as

- $x \in (0, 1]$:

$$\min_{\lambda \in \mathcal{I}} \mathcal{F}_y(\lambda) = \begin{cases} \mathcal{F}_y(\infty), & y \in (0, x], \\ \mathcal{F}_y(\mu_+), & y \in [x, \infty), \end{cases} \tag{5.98}$$

- $x \in (1, \infty)$:

$$\min_{\lambda \in \mathcal{I}} \mathcal{F}_y(\lambda) = \begin{cases} \mathcal{F}_y(\infty), & y \in (0, x], \\ \mathcal{F}_y(\mu_+), & y \in [x, \hat{y}(x)], \\ \mathcal{F}_y(0), & y \in [\hat{y}(x), \infty). \end{cases} \tag{5.99}$$

Finally, the condition numbers (5.87) and (5.88) follow from taking the ratio of (5.97) to (5.98) and (5.99) and substituting $\mathcal{F}_y(\infty) = 1$. ■

Lemma 5.9 (Optimal preconditioning for symmetric negative semi-definite $\mathcal{A}$)**.** *The tight condition number bound $\widetilde{\kappa}(\mathcal{S}_y)$ in (5.87)/(5.88) from Lemma 5.8 is minimized over $y \in$*

$(0, \infty)$ *at* $y = y_*$, *where*

$$y_* = x + \frac{1}{x}. \tag{5.100}$$

Moreover, the minimum value of $\widetilde{\kappa}(\mathcal{S}_y)$ *is*

$$\widetilde{\kappa}(\mathcal{P}_{y_*}) = \frac{1}{2}\left(1 + \sqrt{1 + \frac{1}{x^2}}\right). \tag{5.101}$$

Proof. The function $\mathcal{F}_y(0) = \frac{1}{y}\left(x + \frac{1}{x}\right)$ is strictly decreasing for $y > 0$, and thus $\widetilde{\kappa}(\mathcal{S}_y)$ is strictly decreasing on $(0, x)$ where it is equal to $\mathcal{F}_y(0)$. Conversely, when $x \in (1, \infty)$, $\widetilde{\kappa}(\mathcal{S}_y)$ is strictly increasing on $\left(x\frac{x^2+1}{x^2-1}, \infty\right)$ where it is equal to $\frac{1}{\mathcal{F}_y(0)}$.

The function $\frac{1}{\mathcal{F}_y(\lambda_+)} = \frac{1+\sqrt{1+(y-x)^2}}{2}$ is clearly increasing whenever $y > x$, and thus $\widetilde{\kappa}(\mathcal{S}_y)$ is strictly increasing for $y \in \left(x + \frac{1}{x}, \infty\right)$ when $x \in (0, 1]$, and for $y \in \left(x + \frac{1}{x}, x\frac{x^2+1}{x^2-1}\right)$ when $x \in (1, \infty)$ where it is equal to $\frac{1}{\mathcal{F}_y(\lambda_+)}$.

In summary, $\widetilde{\kappa}(\mathcal{S}_y)$ is strictly decreasing on $y \in (0, x)$, and is it strictly increasing on $y \in \left(x + \frac{1}{x}, \infty\right)$ (except possibly at the point $y = x\frac{x^2+1}{x^2-1}$ when $x \in (1, \infty)$ where it may only be increasing). Therefore the global minimum of $\widetilde{\kappa}(\mathcal{S}_y)$ must occur on $y \in \left[x, x + \frac{1}{x}\right]$, where it is equal to $\frac{\mathcal{F}_y(0)}{\mathcal{F}_y(\lambda_+)}$. The behaviour of this function is not immediately obvious, so let us consider its derivative, which may be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{\mathcal{F}_y(0)}{\mathcal{F}_y(\lambda_+)}\right) = -\frac{x^2+1}{2xy^2\sqrt{1+(y-x)^2}}\left(x^2 + 1 - xy + \sqrt{1+(y-x)^2}\right). \tag{5.102}$$

Note that since $x > 0$, it is the case that $x^2 + 1 - xy > 0$ whenever $y < x + \frac{1}{x}$. Therefore the second term in the above product is positive whenever $y < x + \frac{1}{x}$, and clearly the first is always negative so that $\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{\mathcal{F}_y(0)}{\mathcal{F}_y(\lambda_+)}\right) < 0$ whenever $y < x + \frac{1}{x}$, and thus $\widetilde{\kappa}(\mathcal{S}_y)$ is strictly decreasing for $y \in \left(x, x + \frac{1}{x}\right)$.

By the continuity of $\widetilde{\kappa}(\mathcal{S}_y)$ in y, it must be the case that it is minimized at the interface $y = y_* = x + \frac{1}{x}$. Evaluating (5.87)/(5.88) at $y = x + \frac{1}{x}$ yields (5.101). ■

Remark 5.10 (Original variables η, β, γ, and $\widehat{\mathcal{L}}$)**.** *Restated in the original variables used to define* x, y, *and* $\mathcal{A}$ *in* (5.83), *Lemma 5.9 states that when* $\widehat{\mathcal{L}}$ *is symmetric negative semi-definite,* $\widetilde{\kappa}(\mathcal{S}_\gamma)$ *is minimized over* $\gamma \in (0, \infty)$ *at* $\gamma = \gamma_*$, *where*

$$\gamma_* = \eta + \frac{\beta^2}{\eta}, \tag{5.103}$$

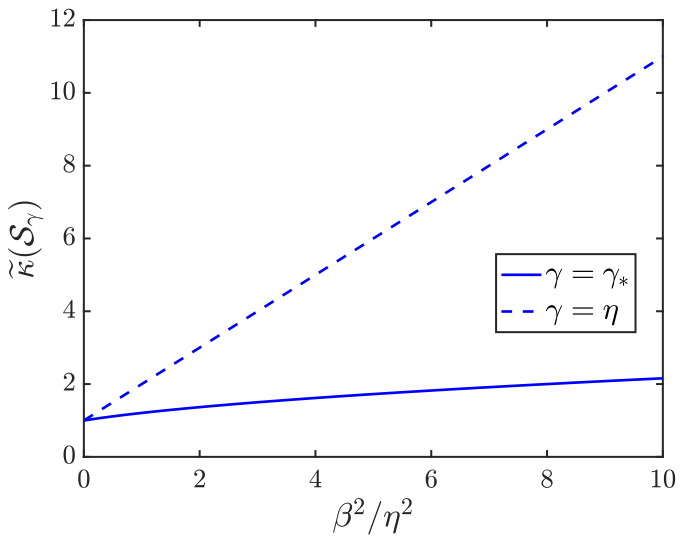

FIGURE 5.3: *Nonlinear FIRK:* Eigenvalue-based analysis when $\widehat{\mathcal{L}}$ is symmetric negative semi-definite. The condition number bound $\widetilde{\kappa}(\mathcal{S}_\gamma)$ is shown for the optimal choice of $\gamma = \gamma_*$ (see (5.104)), and the naive choice of $\gamma = \eta$ (see (5.105)).

and the associated minimum value is

$$\widetilde{\kappa}(\mathcal{S}_{\gamma_*}) = \frac{1}{2}\left(1 + \sqrt{1 + \frac{\beta^2}{\eta^2}}\right). \tag{5.104}$$

Furthermore, it is interesting to consider preconditioning with the naive choice of $\gamma = \eta$, for which the condition number may be evaluated using (5.87)/(5.88) *to give*

$$\kappa(\mathcal{S}_\eta) = \kappa(\mathcal{S}_x) = 1 + \frac{1}{x^2} = 1 + \frac{\beta^2}{\eta^2}. \tag{5.105}$$

The condition number upper bound $\widetilde{\kappa}(\mathcal{S}_\gamma)$ is shown in Figure 5.3 for the optimal choice of $\gamma = \gamma_*$ and for the naive choice of $\gamma = \eta$. The optimal choice clearly yields a much smaller condition number than $\gamma = \eta$. While $\widetilde{\kappa}(\mathcal{S}_{\gamma_*})$ is unbounded with respect to $\frac{\beta^2}{\eta^2}$ (see (5.104)), it grows only very slowly and remains $\mathcal{O}(1)$ for moderate values of $\frac{\beta^2}{\eta^2}$ that correspond to high-order FIRK integration. For example, for 5-stage Gauss, Radau IIA, and Lobatto IIIC, $\frac{\beta^2}{\eta^2}$ does not exceed five (see Table 5.1), so the condition number (5.104) is always less than 1.725. This indicates that Krylov convergence will be fast in practice, even for high-order FIRK integration.

5.4 Field-of-values-based linear preconditioning theory

In this section, the condition number of the preconditioned operator arising in the linear FIRK algorithm (see Section 5.2.1) is analyzed again, and so too is the condition number

of the preconditioned Schur complement that arises in the nonlinear FIRK algorithm (see Section 5.3.1). The previous eigenvalue-based condition number analyses of Sections 5.2.3, 5.2.4 and 5.3.3 assumed that the spatial discretization $\mathcal{L}$ was symmetric or skew symmetric. The analysis in this section removes these assumptions of symmetry or skew symmetry. Here, we instead works with the much more general Assumption 5.2: The field of values of $\mathcal{L}$ lies in the closed left half plane. That is, we make no assumption on the symmetry of $\mathcal{L}$, let alone its normality.[8]

The analysis in this section provides upper bounds on condition numbers of preconditioned operators over the space of all operators $\mathcal{L}$ satisfying Assumption 5.2, and perhaps not surprisingly, such bounds are pessimistic in cases when $\mathcal{L}$ is symmetric definite. Therefore if one is genuinely interested in solving PDEs with symmetric definite $\mathcal{L}$, then the eigenvalue analyses of Sections 5.2.3 and 5.3.3 should be considered instead. On the other hand, it turns out that the upper bounds derived here over the space of operators $\mathcal{L}$ satisfying Assumption 5.2 achieve equality for skew-symmetric $\mathcal{L}$. In fact, much of the general theory derived in this section was motivated by results learned in the skew-symmetric eigenvalue analysis of Section 5.2.4, as the careful reader may notice.

Let us begin by observing that the preconditioned operator $\mathcal{P}_\gamma$ (5.27) in the linear algorithm and the preconditioned Schur complement $\mathcal{S}_\gamma$ (5.82) in the nonlinear algorithm are closely related. Specifically, they may be written as

$$\mathcal{P}_\gamma = \big[(\eta I - \widehat{\mathcal{L}})^2 + \beta^2 I\big](\gamma I - \widehat{\mathcal{L}})^{-1}(\gamma I - \widehat{\mathcal{L}})^{-1}, \tag{5.106}$$

$$\mathcal{S}_\gamma = \big[(\eta I - \widehat{\mathcal{L}})^2 + \beta^2 I\big](\eta I - \widehat{\mathcal{L}})^{-1}(\gamma I - \widehat{\mathcal{L}})^{-1}. \tag{5.107}$$

Given the structure of these operators, it becomes of interest to consider a more general preconditioned operator that depends on two separate constants δ, γ:

$$\mathcal{G}_{\delta,\gamma} := \big[(\eta I - \widehat{\mathcal{L}})^2 + \beta^2 I\big](\delta I - \widehat{\mathcal{L}})^{-1}(\gamma I - \widehat{\mathcal{L}})^{-1}, \quad \delta, \gamma \in (0, \infty). \tag{5.108}$$

Operators (5.106) and (5.107) are then special cases of this general preconditioned operator:

$$\mathcal{P}_\gamma \equiv \mathcal{G}_{\gamma,\gamma}, \quad \mathcal{S}_\gamma \equiv \mathcal{G}_{\eta,\gamma}\,. \tag{5.109}$$

The remainder of this section is structured as follows. First, Theorem 5.11 derives tight bounds on the 2-norm condition number of $\mathcal{G}_{\delta,\gamma}$ (5.108) over all $\widehat{\mathcal{L}}$ that satisfy Assumption 5.2, and further derives the value of $\gamma = \gamma_* = \gamma_*(\delta)$ that minimizes this upper bound

[8] Note that symmetric and skew symmetric matrices are examples of normal matrices, since a real matrix A is normal if $A^\top A = AA^\top$.

for any $\delta \in (0, \infty)$. Corollary 5.15 then shows that the optimal preconditioned operator of the form (5.108) is $\mathcal{G}_{\gamma_*,\gamma_*}$, in the sense that it has the minimum maximum condition number over all $\widehat{\mathcal{L}}$ satisfying Assumption 5.2:

$$\kappa(\mathcal{G}_{\gamma_*,\gamma_*}) = \min_{\delta,\gamma\in(0,\infty)} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{G}_{\delta,\gamma}), \quad \text{with } \gamma_* = \sqrt{\eta^2 + \beta^2}. \tag{5.110}$$

Since $\mathcal{G}_{\gamma_*,\gamma_*} = \mathcal{P}_{\gamma_*}$, an immediate consequence of this is that the preconditioner $(\gamma_* I - \widehat{\mathcal{L}})^{-2}$ used in $\mathcal{P}_{\gamma_*}$ is optimal (in the above sense) over the space of general preconditioners $(\delta I - \widehat{\mathcal{L}})^{-1}(\gamma I - \widehat{\mathcal{L}})^{-1}$ for $\delta, \gamma \in (0, \infty)$. This is the justification for having restricted $\mathcal{P}_\gamma$ in Section 5.2 to use preconditioners of the form $(\gamma I - \widehat{\mathcal{L}})^{-2}$ rather than the more general form $(\delta I - \widehat{\mathcal{L}})^{-1}(\gamma I - \widehat{\mathcal{L}})^{-1}$. Corollary 5.15 also provides tight bounds on the condition number of $\mathcal{P}_{\gamma_*} = \mathcal{G}_{\gamma_*,\gamma_*}$, where the optimal constant in this case is $\gamma_* = \sqrt{\eta^2 + \beta^2}$. Notice the optimal constant is the same as that identified in Sections 5.2.3 and 5.2.4 where eigenvalue analysis was used under the assumption that $\mathcal{L}$ was symmetric definite or skew symmetric, respectively.

Finally, Corollary 5.16 uses the results of Theorem 5.11 to identify the value of $\gamma = \gamma_*(\eta)$ that leads to the minimization of the maximum condition number of $\mathcal{S}_\gamma$ in (5.107) over all $\widehat{\mathcal{L}}$ satisfying Assumption 5.2. Corollary 5.16 also provides tight bounds on the condition number of $\mathcal{S}_{\gamma_*}$, where the optimal constant in this case is $\gamma_* = \eta + \frac{\beta^2}{\eta}$. Notice that the optimal constant is the same as that identified in Section 5.3.3 using eigenvalue analysis under the assumption that $\mathcal{L}$ was symmetric definite.

Theorem 5.11 (Optimal preconditioning with (5.108))**.** *Suppose Assumptions 5.1 and 5.2 hold, that is, $\eta > 0$ and $W(\widehat{\mathcal{L}}) \leq 0$, and suppose $\widehat{\mathcal{L}} \in \mathbb{R}^{N\times N}$. Let $\mathcal{G}_{\delta,\gamma}$ denote the preconditioned operator in (5.108), and $\kappa(\mathcal{G}_{\delta,\gamma})$ denote its two-norm condition number. Finally, define γ_* by*

$$\gamma_* := \frac{\eta^2 + \beta^2}{\delta}. \tag{5.111}$$

Then

$$\kappa(\mathcal{G}_{\delta,\gamma_*}) \leq \frac{1}{2\eta}\left(\delta + \frac{\eta^2 + \beta^2}{\delta}\right) = \frac{1}{2\eta}(\delta + \gamma_*). \tag{5.112}$$

Moreover, (i) bound (5.112) is tight when considered over all $\widehat{\mathcal{L}} \in \mathbb{R}^{N\times N}$ satisfying Assumption 5.2 in the sense that $\exists\, \widehat{\mathcal{L}}$ such that (5.112) holds with equality, and (ii) $\gamma = \gamma_*$ is optimal in the sense that, without further assumptions on $\widehat{\mathcal{L}}$, γ_* minimizes a tight upper

bound on $\kappa(\mathcal{G}_{\delta,\gamma})$:

$$\gamma_*(\delta) = \arg\min_{\gamma\in(0,\infty)} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{G}_{\delta,\gamma}), \quad \textit{or} \quad \kappa(\mathcal{G}_{\delta,\gamma_*(\delta)}) = \min_{\gamma\in(0,\infty)} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{G}_{\delta,\gamma}). \tag{5.113}$$

Proof. To aid in the readability of the proof of this Theorem, it is proved by the following sequence of three lemmas, which successively prove: the upper bound (5.112); tightness of this upper bound; and, the optimality of γ_*. Together, the three lemmas complete the proof. ■

Lemma 5.12 (Upper bound)**.** *Under the assumptions of Theorem 5.11, the upper bound $\kappa(\mathcal{G}_{\delta,\gamma_*}) \leq \frac{1}{2\eta}\left(\delta + \frac{\eta^2+\beta^2}{\delta}\right)$ in (5.112) holds.*

Proof. The square of the condition number of $\mathcal{G}_{\delta,\gamma}$ is given by

$$\kappa^2(\mathcal{G}_{\delta,\gamma}) = \| \mathcal{G}_{\delta,\gamma} \|^2 \| \mathcal{G}_{\delta,\gamma}^{-1} \|^2 = \max_{\boldsymbol{v}\neq 0} \frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} \frac{1}{\min\limits_{\boldsymbol{v}\neq 0} \frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2}}, \tag{5.114}$$

where, for real-valued $\widehat{\mathcal{L}}$, the max and min can be obtained by restricting $\boldsymbol{v}$ to be real valued. The key step in establishing the upper bound (5.112) is bounding $\| \mathcal{G}_{\delta,\gamma} \|^2$ and $\| \mathcal{G}_{\delta,\gamma}^{-1} \|^2$ from above, which will be done by bounding $\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2/\|\boldsymbol{v}\|^2$ from above and below, respectively. Considering the form of the preconditioned operator $\mathcal{G}_{\delta,\gamma}$ in (5.108), it is advantageous to make the substitution $\boldsymbol{v} \mapsto (\gamma I - \widehat{\mathcal{L}})(\delta I - \widehat{\mathcal{L}})\boldsymbol{w}$. The quantity $\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2$ can then be expanded for real-valued $\boldsymbol{v}$ (and, thus, real-valued $\boldsymbol{w}$) as

$$\begin{aligned}
\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2 &= \left\|[(\eta I - \widehat{\mathcal{L}})^2 + \beta^2]\boldsymbol{w}\right\|^2, \\
&= \left\|[(\eta^2+\beta^2)\boldsymbol{w} - 2\eta\, \widehat{\mathcal{L}}\, \boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w}\right\|^2 \\
&= \left\|(\eta^2+\beta^2)\boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w}\right\|^2 - 4\eta(\eta^2+\beta^2)\langle \widehat{\mathcal{L}}\, \boldsymbol{w}, \boldsymbol{w}\rangle - 4\eta\langle \widehat{\mathcal{L}}(\widehat{\mathcal{L}}\, \boldsymbol{w}), \widehat{\mathcal{L}}\, \boldsymbol{w}\rangle + 4\eta^2 \left\| \widehat{\mathcal{L}}\, \boldsymbol{w}\right\|^2.
\end{aligned} \tag{5.115}$$

Similarly, expanding $\|\boldsymbol{v}\|^2$ yields

$$\begin{aligned}
\|\boldsymbol{v}\|^2 &= \left\|(\gamma I - \widehat{\mathcal{L}})(\delta I - \widehat{\mathcal{L}})\boldsymbol{w}\right\|^2, \\
&= \left\| \delta\gamma\boldsymbol{w} - (\delta+\gamma)\, \widehat{\mathcal{L}}\, \boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w}\right\|^2, \\
&= \left\| \delta\gamma\boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w}\right\|^2 - 2\delta\gamma(\delta+\gamma)\langle \widehat{\mathcal{L}}\, \boldsymbol{w}, \boldsymbol{w}\rangle - 2(\delta+\gamma)\langle \widehat{\mathcal{L}}(\widehat{\mathcal{L}}\, \boldsymbol{w}), \widehat{\mathcal{L}}\, \boldsymbol{w}\rangle + (\delta+\gamma)^2 \left\| \widehat{\mathcal{L}}\, \boldsymbol{w}\right\|^2.
\end{aligned} \tag{5.116}$$

The key ratio in (5.114) can be written in the form

$$\frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} = \frac{c_0(\boldsymbol{w})f_0(\boldsymbol{w}) + c_1 f_1(\boldsymbol{w}) + c_2 f_2(\boldsymbol{w}) + c_3 f_3(\boldsymbol{w})}{f_0(\boldsymbol{w}) + f_1(\boldsymbol{w}) + f_2(\boldsymbol{w}) + f_3(\boldsymbol{w})}, \tag{5.117}$$

where for $\delta, \gamma > 0$, the following functions and constants have been defined

$$\begin{aligned}
f_0 &:= \left\| \delta\gamma \boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w} \right\|^2 \geq 0, & c_0 &:= \frac{\left\|(\eta^2+\beta^2)\boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w}\right\|^2}{\left\| \delta\gamma \boldsymbol{w} + \widehat{\mathcal{L}}^2\, \boldsymbol{w}\right\|^2} \geq 0,\\
f_1 &:= -2\delta\gamma(\delta+\gamma)\langle \widehat{\mathcal{L}}\, \boldsymbol{w}, \boldsymbol{w}\rangle \geq 0, & c_1 &:= \frac{\eta^2+\beta^2}{\delta\gamma}\frac{2\eta}{\delta+\gamma} > 0,\\
f_2 &:= -2(\delta+\gamma)\langle \widehat{\mathcal{L}}(\widehat{\mathcal{L}}\, \boldsymbol{w}), \widehat{\mathcal{L}}\, \boldsymbol{w}\rangle \geq 0, & c_2 &:= \frac{2\eta}{\delta+\gamma} > 0,\\
f_3 &:= (\delta+\gamma)^2 \left\| \widehat{\mathcal{L}}\, \boldsymbol{w}\right\|^2 \geq 0, & c_3 &:= \left(\frac{2\eta}{\delta+\gamma}\right)^2 > 0.
\end{aligned} \tag{5.118}$$

Note that functions f_1 and f_2 are non-negative by assumption of $W(\widehat{\mathcal{L}}) \leq 0$, while for all $\boldsymbol{w} \neq \boldsymbol{0}$, it must hold that either $c_0 f_0 > 0$ or $c_3 f_3 > 0$ (or both, because $c_3 f_3 = 0$ i.f.f. $\widehat{\mathcal{L}}\, \boldsymbol{w} = \boldsymbol{0}$, which implies $c_0 f_0 > 0$ for $\boldsymbol{w} \neq \boldsymbol{0}$).

Since all of the addends in the numerator and denominator of (5.117) are non-negative, and at least one addend in each is positive, (5.117) can simply be bounded as

$$\min\{c_0, c_1, c_2, c_3\} =: c_{\min} \leq \frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} \leq c_{\max} := \max\{c_0, c_1, c_2, c_3\}.$$

Applying these bounds to the norms in (5.114) yields

$$\| \mathcal{G}_{\delta,\gamma} \| \leq \sqrt{c_{\max}}, \quad \| \mathcal{G}_{\delta,\gamma}^{-1} \| \leq \frac{1}{\sqrt{c_{\min}}}. \tag{5.119}$$

Bounding $c_{\min}$ and $c_{\max}$ for general $\gamma \in (0, \infty)$ is difficult due to the difficulty of bounding c_0 (5.118). Specifically, the sign of $\langle \widehat{\mathcal{L}}^2\, \boldsymbol{w}, \boldsymbol{w}\rangle$ (which appears in expanding the squared norms in both the numerator and denominator of c_0) is not known for general $\widehat{\mathcal{L}}$, noting that the sign of $W(\widehat{\mathcal{L}})$ does not determine that of $W(\widehat{\mathcal{L}}^2)$. However, observe from (5.118) that the judicious choice of $\gamma = \gamma_* := (\eta^2+\beta^2)/\delta$ yields $c_0(\boldsymbol{w}) = 1$. Moreover, in the final part of this proof (i.e., in the proof of Lemma 5.14) it will be shown that $\gamma = \gamma_*$ is optimal, and, as such, moving forward let us limit our consideration to $\gamma = \gamma_*$ when bounding $c_{\min}$ and $c_{\max}$.

Letting $\gamma = \gamma_* := (\eta^2+\beta^2)/\delta$, from (5.118) one has $c_0 = 1 \geq c_1 = c_2 = \sqrt{c_3} = 2\eta/(\delta+\gamma_*)$, where the inequality $1 \geq 2\eta/(\delta+\gamma_*)$ follows by noting the equivalent relation $(\delta^2 - 2\eta\delta + \eta^2) + \beta^2 \geq 0$ for all $\eta, \delta > 0$. Thus, for $\gamma = \gamma_*$, the bounds in (5.119)

are given by

$$\| \mathcal{G}_{\delta,\gamma_*} \| \leq 1, \quad \| \mathcal{G}_{\delta,\gamma_*}^{-1} \| \leq \frac{\delta + \gamma_*}{2\eta} = \frac{1}{2\eta}\left(\delta + \frac{\eta^2 + \beta^2}{\delta}\right) \tag{5.120}$$

Applying these bounds to the condition number (5.114) yields the upper bound in (5.112).

■

Let us now show that bound (5.112) is tight. We do so by construction, showing that equality is achieved for certain matrices that satisfy Assumption 5.2.

Lemma 5.13 (Tightness)**.** *Under the assumptions of Theorem 5.11, $\exists\ \widehat{\mathcal{L}}$ such that the upper bound $\kappa(\mathcal{G}_{\delta,\gamma_*}) \leq \frac{1}{2\eta}\left(\delta + \frac{\eta^2+\beta^2}{\delta}\right)$ in (5.112) holds with equality.*

Proof. Note that the min/max of $\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2/\|\boldsymbol{v}\|^2$ over $\boldsymbol{v}$ for real-valued $\mathcal{G}_{\delta,\gamma}$ is equivalent when minimizing over real or complex $\boldsymbol{v}$; let us now consider complex $\boldsymbol{v}$ for theoretical purposes. To that end, let $\boldsymbol{v} = (\gamma I - \widehat{\mathcal{L}})(\delta I - \widehat{\mathcal{L}})\boldsymbol{w}$, but suppose that $(\mathrm{i}\xi, \boldsymbol{w})$ is an eigenpair of $\widehat{\mathcal{L}}$, with ξ a real number and $\boldsymbol{w}$ a complex eigenvector. Plugging into the expression for $\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2$ in (5.115) and $\|\boldsymbol{v}\|^2$ in (5.116), and taking the ratio as in (5.117), define the following function of ξ:

$$\mathcal{H}_{\delta,\gamma}(\xi) := \left.\frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2}\right|_{\widehat{\mathcal{L}}\,\boldsymbol{w}=\mathrm{i}\xi\boldsymbol{w}} = \frac{|(\eta - \mathrm{i}\xi)^2 + \beta^2|^2}{|(\delta\gamma - \xi^2 - \mathrm{i}(\delta+\gamma)\xi|^2} = \frac{(\delta\gamma_* - \xi^2)^2 + (2\eta\xi)^2}{(\delta\gamma - \xi^2)^2 + [\xi(\delta+\gamma)]^2}, \tag{5.121}$$

where the fact that $\delta\gamma_* = \eta^2 + \beta^2$ has been made use of. By virtue of restricting that $\boldsymbol{w}$ be an eigenvector (rather than any vector in $\mathbb{R}^N$), from (5.114) one has

$$\frac{1}{\| \mathcal{G}_{\delta,\gamma}^{-1} \|^2} = \min_{\boldsymbol{v}\neq 0} \frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} \leq \mathcal{H}_{\delta,\gamma}(\xi) \leq \max_{\boldsymbol{v}\neq 0} \frac{\| \mathcal{G}_{\delta,\gamma}\, \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} = \| \mathcal{G}_{\delta,\gamma} \|^2. \tag{5.122}$$

In other words, any value of $1/\mathcal{H}_{\delta,\gamma}(\xi)$ serves as a lower bound on $\| \mathcal{G}_{\delta,\gamma}^{-1} \|^2$, while any value of $\mathcal{H}_{\delta,\gamma}(\xi)$ serves as a lower bound on $\| \mathcal{G}_{\delta,\gamma} \|^2$. Therefore, the ratio of any two values of $\mathcal{H}_{\delta,\gamma}(\xi)$ provides a lower bound on $\| \mathcal{G}_{\delta,\gamma} \|^2/\| \mathcal{G}_{\delta,\gamma}^{-1} \|^2 = \kappa^2(\mathcal{G}_{\delta,\gamma})$.

Let us now show that bound (5.112) on $\kappa(\mathcal{G}_{\delta,\gamma_*})$ is tight. Considering (5.121) at the judiciously chosen eigenvalues of $\mathrm{i}\xi = \{0, \pm\mathrm{i}\sqrt{\delta\gamma_*}\}$, one has

$$\mathcal{H}_{\delta,\gamma}(0) = \frac{\gamma_*^2}{\gamma^2}, \qquad \mathcal{H}_{\delta,\gamma}(\pm\sqrt{\delta\gamma_*}) = \frac{(2\eta)^2\gamma_*}{\delta(\gamma-\gamma_*)^2 + \gamma_*(\delta+\gamma)^2}. \tag{5.123}$$

First observe from (5.122) and (5.123) that $\| \mathcal{G}_{\delta,\gamma_*} \|^2 \geq \mathcal{H}_{\delta,\gamma_*}(0) = 1$, and thus the upper bound on $\| \mathcal{G}_{\delta,\gamma_*} \|$ given by (5.120) achieves equality for a matrix $\widehat{\mathcal{L}}$ having an eigenvalue of $\xi = 0$. Secondly, observe from (5.122) and (5.123) that $\| \mathcal{G}_{\delta,\gamma_*}^{-1} \|^2 \geq 1/\mathcal{H}_{\delta,\gamma_*}(\pm\sqrt{\delta\gamma_*}) =$

$[(\delta+\gamma_*)/(2\eta)]^2$, and thus the upper bound on $\| \mathcal{G}_{\delta,\gamma_*}^{-1} \|$ given by (5.120) achieves equality for a matrix $\widehat{\mathcal{L}}$ having eigenvalues $\mathrm{i}\xi = \pm\mathrm{i}\sqrt{\delta\gamma_*}$. Therefore, bound (5.112) on $\kappa(\mathcal{G}_{\delta,\gamma_*})$ achieves equality for any matrix $\widehat{\mathcal{L}}$ having eigenvalues $\{0, \pm\mathrm{i}\sqrt{\delta\gamma_*}\}$. ■

Lastly, having shown that (5.112) is tight, let us now show that $\gamma = \gamma_*$ is optimal in terms of minimizing the maximum condition number over all $\widehat{\mathcal{L}} \in \mathbb{R}^{N\times N}$ satisfying Assumption 5.2, by showing that $\exists$ matrices $\widehat{\mathcal{L}}$ for which $\kappa(\mathcal{G}_{\delta,\gamma}) > \kappa(\mathcal{G}_{\delta,\gamma_*})$ for any $\gamma \in (0,\infty) \setminus \gamma_*$.

Lemma 5.14 (Optimality of γ_*)**.** *Under the assumptions of Theorem 5.11, $\gamma = \gamma_*$ is optimal in the sense that it minimizes a tight upper bound on $\kappa(\mathcal{G}_{\delta,\gamma})$ over $\gamma \in (0,\infty)$: $\gamma_* = \underset{\gamma\in(0,\infty)}{\arg\min} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{G}_{\delta,\gamma})$, as in (5.113).*

Proof. As in the proof of Lemma 5.13, consider a matrix $\widehat{\mathcal{L}}$ with eigenvalues $\{0, \pm\mathrm{i}\sqrt{\delta\gamma_*}\}$, such that $\kappa(\mathcal{G}_{\delta,\gamma_*}) = (\delta+\gamma_*)/(2\eta)$. Also from the proof of Lemma 5.13, let us consider the lower bound $\mathcal{H}_{\delta,\gamma}(\xi_1)/\mathcal{H}_{\delta,\gamma}(\xi_2) \le \kappa^2(\mathcal{G}_{\delta,\gamma})$ with judiciously chosen values of ξ_1 and ξ_2. Specifically, from (5.122), and (5.123), one has for $0 < \gamma < \gamma_*$,

$$\kappa^2(\mathcal{G}_{\delta,\gamma_*}) = \frac{(\delta+\gamma_*)^2}{(2\eta)^2} < \frac{\gamma_*[\delta(\gamma-\gamma_*)^2 + \gamma_*(\delta+\gamma)^2]}{(2\eta\gamma)^2} = \frac{\mathcal{H}_{\delta,\gamma}(0)}{\mathcal{H}_{\delta,\gamma}(\pm\sqrt{\delta\gamma_*})} \le \kappa^2(\mathcal{G}_{\delta,\gamma}). \quad (5.124)$$

To see why the first inequality in (5.124) holds, note that $\eta, \gamma > 0$ and multiply both sides by $(2\eta\gamma)^2$ and then subtract the term on the left hand side to get the equivalent inequality

$$\gamma_*\delta(\gamma-\gamma_*)^2 + \Big[\big(\gamma_*(\delta+\gamma)\big)^2 - \big(\gamma(\delta+\gamma_*)\big)^2\Big] = \gamma_*\delta(\gamma-\gamma_*)^2 + \Big[\delta(\gamma_*-\gamma)[2\gamma_*\gamma + \delta(\gamma_*+\gamma)\Big] > 0. \quad (5.125)$$

Clearly the latter form of the inequality is satisfied when $\gamma \in (0,\gamma_*)$ because because both terms in the sum are positive since $\gamma_*, \delta > 0$.

Now let us reconsider the lower bound $\mathcal{H}_{\delta,\gamma}(\xi_1)/\mathcal{H}_{\delta,\gamma}(\xi_2) \le \kappa^2(\mathcal{G}_{\delta,\gamma})$ with a different choice of ξ_1 than above. Specifically, suppose that $\widehat{\mathcal{L}}$ has eigenvalues $\mathrm{i}\xi \to \pm\mathrm{i}\infty$, which, when substituted into (5.121), yields $\lim_{\xi\to\pm\infty} \mathcal{H}_{\delta,\gamma}(\xi) = 1$. Combining with (5.122) and (5.123), one has for $\gamma_* < \gamma < \infty$,

$$\kappa^2(\mathcal{G}_{\delta,\gamma_*}) = \frac{(\delta+\gamma_*)^2}{(2\eta)^2} < \frac{\delta(\gamma-\gamma_*)^2 + \gamma_*(\delta+\gamma)^2}{(2\eta)^2\gamma_*} = \frac{\mathcal{H}_{\delta,\gamma}(\pm\infty)}{\mathcal{H}_{\delta,\gamma}(\pm\sqrt{\delta\gamma_*})} \le \kappa^2(\mathcal{G}_{\delta,\gamma}). \quad (5.126)$$

To see why the first inequality in (5.126) holds, note that $\eta, \gamma_* > 0$ and multiply both sides by $(2\eta)^2\gamma_*$ then subtract the term on the left hand side to get the equivalent inequality

$$\delta(\gamma-\gamma_*)^2 + \gamma_*\Big[(\delta+\gamma)^2 - (\delta+\gamma_*)^2\Big] = \delta(\gamma-\gamma_*)^2 + \gamma_*\Big[(\gamma-\gamma_*)(2\delta+\gamma+\gamma_*)\Big] > 0. \quad (5.127)$$

Clearly the latter form of this inequality is satisfied when $\gamma \in (\gamma_*, \infty)$ because both terms in the sum are positive since $\delta, \gamma_* > 0$.

By construction in (5.124) and (5.126), it has been shown that $\exists\ \widehat{\mathcal{L}}$ satisfying the assumptions of Theorem 5.11 (namely those with eigenvalues $\{0, \pm \mathrm{i}\sqrt{\eta^2+\beta^2}, \pm \mathrm{i}\infty\}$) for which $\kappa(\mathcal{G}_{\delta,\gamma}) > \kappa(\mathcal{G}_{\delta,\gamma_*}) = (\delta+\gamma_*)/(2\eta)$ for all $\gamma \in (0,\infty) \setminus \gamma_*$. Since $\kappa(\mathcal{G}_{\delta,\gamma_*}) \leq (\delta+\gamma_*)/(2\eta)$ for general $\widehat{\mathcal{L}}$ satisfying the assumptions of Theorem 5.11, it holds that $\gamma = \gamma_*$ is the minimizer over $\gamma \in (0,\infty)$ of a tight upper bound on $\kappa(\mathcal{G}_{\delta,\gamma})$, $\gamma_* = \underset{\gamma\in(0,\infty)}{\arg\min} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{G}_{\delta,\gamma})$ as in (5.113). ■

Having proved Theorem 5.11, let us now utilize this result to identify the optimal value of $\delta \in (0,\infty)$ that leads to the minimization of the maximum condition number of $\mathcal{G}_{\delta,\gamma_*(\delta)}$ when considered over all $\widehat{\mathcal{L}}$ satisfying Assumption 5.2.

Corollary 5.15 (Optimal preconditioning of $\mathcal{G}_{\delta,\gamma}$ with $\delta = \gamma = \gamma_*$)**.** *Suppose that the assumptions of Theorem 5.11 hold, then $\mathcal{G}_{\gamma_*,\gamma_*}$ has the minimum condition number of all $\mathcal{G}_{\delta,\gamma}$ for $\delta, \gamma \in (0,\infty)$ when considered over all $\widehat{\mathcal{L}} \in \mathbb{R}^{N\times N}$ satisfying Assumption 5.2,*

$$\kappa(\mathcal{G}_{\gamma_*,\gamma_*}) = \min_{\delta,\gamma\in(0,\infty)} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{G}_{\delta,\gamma}), \tag{5.128}$$

where

$$\gamma_* = \sqrt{\eta^2+\beta^2}. \tag{5.129}$$

Moreover, the minimum condition number is tightly bounded by

$$\kappa(\mathcal{G}_{\gamma_*,\gamma_*}) \leq \sqrt{1+\frac{\beta^2}{\eta^2}}. \tag{5.130}$$

Proof. In Theorem 5.11 it was shown that for any $\delta \in (0,\infty)$ a tight upper bound on the condition number of $\kappa(\mathcal{G}_{\delta,\gamma})$ over all $\widehat{\mathcal{L}}$ is minimized with respect to $\gamma \in (0,\infty)$ when $\gamma = \gamma_*(\delta)$, with its minimum value given by (5.112). Now our task is to minimize this tight upper bound with respect to $\delta \in (0,\infty)$, and thus minimize the the maximum of $\kappa(\mathcal{G}_{\delta,\gamma})$ for $\delta, \gamma \in (0,\infty)$ when considered over all $\widehat{\mathcal{L}}$. To this end, consider the derivative of the bound (5.112) on $\kappa(\mathcal{G}_{\delta,\gamma_*(\delta)})$ with respect to δ:

$$\frac{\mathrm{d}}{\mathrm{d}\delta}\left[\frac{1}{2\eta}\left(\delta+\frac{\eta^2+\beta^2}{\delta}\right)\right] = \frac{1}{2\eta}\left(1-\frac{\eta^2+\beta^2}{\delta^2}\right). \tag{5.131}$$

For $\delta > 0$ there is only one stationary point, which occurs at $\delta = \sqrt{\eta^2+\beta^2}$. Since the bound (5.112) is increasing as $\delta \to 0^+$ and $\delta \to \infty$, the stationary point $\delta = \sqrt{\eta^2+\beta^2}$

must be a local minimum. The associated optimal value of $\gamma_*(\delta)$ is calculated from (5.111) by substituting $\delta = \sqrt{\eta^2 + \beta^2}$, which gives $\gamma_*(\delta) = \sqrt{\eta^2 + \beta^2}$.

Finally, the associated upper bound of (5.130) follows from (5.112) after substituting $\delta = \sqrt{\eta^2 + \beta^2}$. ■

Finally, let us utilize the results of Theorem 5.11 to determine the optimal value of γ for use in the preconditioned Schur complement operator (5.107) that arises in the nonlinear FIRK algorithm.

Corollary 5.16 (Optimal preconditioning with $\mathcal{S}_\gamma$ of (5.107))**.** *Suppose that the assumptions of Theorem 5.11 hold. Then the preconditioned Schur complement $\mathcal{S}_{\gamma_*}$ has the minimum condition number of all $\mathcal{S}_\gamma$ for $\gamma \in (0, \infty)$ when considered over all $\widehat{\mathcal{L}} \in \mathbb{R}^{N\times N}$ satisfying Assumption 5.2,*

$$\kappa(\mathcal{S}_{\gamma_*}) = \min_{\gamma\in(0,\infty)} \max_{\widehat{\mathcal{L}}} \kappa(\mathcal{S}_\gamma), \tag{5.132}$$

where

$$\gamma_* = \eta + \frac{\beta^2}{\eta}. \tag{5.133}$$

Moreover, the minimum condition number is tightly bounded by

$$\kappa(\mathcal{S}_{\gamma_*}) \leq 1 + \frac{\beta^2}{2\eta^2}. \tag{5.134}$$

Proof. The preconditioned Schur complement $\mathcal{S}_\gamma$ (5.107) is equivalent to the more general preconditioned operator $\mathcal{G}_{\delta,\gamma}$ (5.108) analyzed in Theorem 5.11 when $\delta = \eta$: $\mathcal{S}_\gamma \equiv \mathcal{G}_{\eta,\gamma}$. The value of $\gamma = \gamma_*(\delta)$ that minimizes a tight upper bound on $\kappa(\mathcal{G}_{\delta,\gamma})$ over $\gamma \in (0, \infty)$ is given by (5.133), and the associated upper bound on $\kappa(\mathcal{G}_{\delta,\gamma_*(\delta)})$ is given by (5.112). Upon substituting $\delta = \eta$ into these expressions, results (5.133) and (5.134) follow immediately. ■

5.5 Numerical results

In this section, numerical tests are conducted for the linear and nonlinear FIRK algorithms that were presented and analyzed in Sections 5.2 and 5.3, respectively. Experiments are conducted on PDEs with known solutions so that the accuracy of the software implementation and the FIRK methods can be verified. Numerical tests are also conducted to verify the linear preconditioning theory of the previous sections. Tests for linear problems are given in Section 5.5.1, and then nonlinear problems in Section 5.5.2.

It should be noted that the test problems considered here are not particularly challenging in the sense that they are not highly non-symmetric, nor do they suffer from order reduction, for example. This section is meant to serve more as a testing ground to complement the linear preconditioning theory, and to confirm that our implementation of the methods provided in [89] functions as expected (these test problems were used in the development of package [89]).[9]

The numerical results shown here have used our implementation of the FIRK algorithms provided in the software package [89]. Package [89] is a `C++` code developed for this work that derives several classes from the MFEM library [1], with the primary ones being `TimeDependentOperator` and `ODESolver`. It also uses explicitly several of MFEM's classes such as its Krylov and Newton solvers, for example. It is straightforward to use [89] if one uses a spatial discretization from the MFEM library (it provides numerous finite-element discretizations), or if one has a discretization that is already interfaced or coupled with MFEM. Specifically, the user has to provide several key computational kernels associated with the ODEs $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \mathcal{N}(\boldsymbol{u}, t)$ from (5.2). These are: the action of $\mathcal{N}$; the action of $\mathcal{L}_i$, which is a linearization of $\mathcal{N}$; and preconditioners for linear operators of the form $\alpha I - \delta t\, \mathcal{L}_i$ for some constant α.

Roughly speaking, provided with these basic computational kernels, [89] is able to apply arbitrarily high-order FIRK methods to the ODEs $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \mathcal{N}(\boldsymbol{u}, t)$ using the algorithms described in this chapter, including more sophisticated Newton methods than the simple Newton method described in Section 5.3.1. In order to apply the algorithms for a specific FIRK scheme, information associated with its Butcher tableau (5.3) is required, as described earlier in the chapter. For both the linear and nonlinear algorithms, the Butcher tableau itself does not provide enough information since additional quantities are needed such as the eigenvalues of A_0^{-1}, the adjugate of the matrix $A_0^{-1} - xI_s$, or the real Schur decomposition of A_0^{-1}, for example. The required Butcher tableau data is implemented in [89] for a collection of DIRK and FIRK schemes, including 1–5 stage Gauss, Radau IIA, and Lobatto IIIC methods. In principle, the code can be used to apply other FIRK schemes provided the user specifies the necessary quantities associated with the Butcher tableau.

5.5.1 The linear setting

This section considers numerical tests for the linear FIRK algorithm from Section 5.2. The test problem is chosen as the following constant-coefficient advection-diffusion equation in

[9] Additional test problems may be found in our articles [91, 90], some of which are more challenging than those presented here.

two spatial dimensions:

$$u_t + 0.85u_x + u_y = 0.3u_{xx} + 0.25u_{yy} + s(x, y, t), \quad (x, y, t) \in (-1, 1)^2 \times (0, 2]. \tag{5.135}$$

The PDE is posed on a periodic spatial domain, and the source term $s(x, y, t)$ is chosen such that the solution of the PDE is $u(x, y, t) = \sin^4(\pi/2[x - 1 - 0.85t]) \sin^4(\pi/2[y - 1 - t]) \exp(-[0.3 + 0.25]t)$. Numerical tests consider FIRK time integration methods with orders of accuracy equal to three and four, which are paired with 4th-order central finite differences in space, and FIRK methods with orders of accuracy equal to seven and eight, which are paired with 8th-order central finite differences in space. The spatial mesh is discretized with $n_x \times n_y$ nodes using a constant mesh spacing of h, and a time-step of $\delta t = 2h$ is used to balance discretization errors in space and time. Specifically, problems will be considered for $n_x \times n_y = (2^3 \times 2^3, 2^4 \times 2^4, 2^5 \times 2^5, 2^6 \times 2^6, 2^7 \times 2^7)$ corresponding to $\delta t = (2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5})$. Note that a central-finite-difference discretization of $-(0.85u_x + u_y)$ is skew symmetric, while that of $0.3u_{xx} + 0.25u_{yy}$ is symmetric negative semi-definite. Therefore, $\mathcal{L}$ which is the sum of these two discretizations is neither symmetric definite nor skew symmetric, but it most certainly fulfils Assumption 5.2, since it is normal and its eigenvalues lie in the closed left half plane.[10] Note that ERK integration, for example, is unsuitable for this problem because it leads to a highly restrictive CFL condition of the form $\delta t \lesssim \mathcal{O}(h^2)$ in order to contain the largest eigenvalues of the diffusion discretization—which grow as $\mathcal{O}(-\delta t h^{-2})$—inside its finite-sized stability region. Furthermore, stability regions of ERK schemes typically provide very little (if any) coverage of the imaginary axis, and, so, it is likely that under no time-step size restriction would the stability region of an ERK scheme be capable of containing the purely imaginary eigenvalues of the high-order central advection discretization.

To make interfacing with the MFEM-based implementation [89] of the linear FIRK algorithm as straightforward as possible, the finite-difference spatial discretizations have been implemented in a `C++` class that is derived from MFEM's `Operator` class. The discretizations are assembled as `HypreParMatrix` objects, such that all parallel communication that takes place during matrix-vector products is handled by the hypre library [36]. All numerical tests use four cores in space.

The specific FIRK schemes considered are: Gauss(4) and Gauss(8), which are of 4th- and 8th-order, and have 2 and 4 stages, respectively; Lobatto IIIC(4) and Lobatto IIIC(8), which are of 4th- and 8th-order, and have 3 and 5 stages, respectively; and, Radau IIA(3)

[10] $\mathcal{L}$ is easily seen to be negative semi-definite since it is the sum of two negative semi-definite operators that are simultaneously diagonalized (they are block circulant with circulant blocks, and, thus, are diagonalized by the two-dimensional DFT). Moreover, since the individual discretizations are simultaneously diagonalized, they commute, and it is trivial to show that their sum $\mathcal{L}$ satisfies $\mathcal{L}^\top \mathcal{L} = \mathcal{L}\mathcal{L}^\top$, and is therefore normal.

and Radau IIA(7), which are of 3rd- and 7th-order, and have 2 and 4 stages, respectively. Also considered for reference purposes are: L–SDIRK(4), a 5-stage, 4th-order L-stable SDIRK method (see [48, Tab. 6.5]); and, A–SDIRK(4), a 3-stage, 4th-order, A-stable SDIRK method (see [48, (6.18)]). Recall from earlier chapters of this thesis that an SDIRK (or singly DIRK) method is a DIRK method in which the entries along the diagonal of A_0 are constant.

Before discussing further the numerical results, it is useful to recall some details of the linear FIRK algorithm from Section 5.2.1. Specifically, recall that a sequence of linear systems needs to be solved to find the FIRK stage vectors at every time step, and that these linear systems come in two different varieties. The first type is of the form $(\zeta_i I - \delta t\, \mathcal{L})\boldsymbol{x}_i = \boldsymbol{b}_i$, where ζ_i is a real eigenvalue of the inverse Butcher matrix $A_0^{-1} \in \mathbb{R}^{s\times s}$. The second type is of the form $[(\eta_i I - \delta t\, \mathcal{L})^2 + \beta_i^2 I]\boldsymbol{x}_i = \boldsymbol{b}_i$, where $\eta_i \pm \mathrm{i}\beta_i$ is a complex-conjugate eigenvalue pair of A_0^{-1}. An s-stage FIRK method will have $\frac{s}{2}$ complex-conjugate eigenvalue pairs of eigenvalues if s is even, and $\frac{s-1}{2}$ complex-conjugate eigenvalue pairs and a single real eigenvalue if s is odd. Furthermore, an s-stage DIRK scheme simply has s real eigenvalues since its A_0 is triangular. Both varieties of linear system are solved with a preconditioned Krylov method. The preconditioner for systems of the first type is simply an inexpensive preconditioner for $(\zeta_i I - \delta t\, \mathcal{L})$, for example, a single algebraic multigrid (AMG) iteration. The preconditioner for the second type is chosen as *two* applications of an inexpensive preconditioner for $(\gamma_i I - \delta t\, \mathcal{L})$, where γ_i is some constant (e.g., the optimal constant $\gamma_i = \gamma_{i*}$, or the naive choice $\gamma_i = \eta_i$).

Due to the diffusive, but non-symmetric nature of the spatial discretization $\mathcal{L}$ of (5.135), GMRES(30) is chosen as the Krylov method to solve the aforementioned linear systems. The solver uses an absolute and relative stopping tolerance of 10^{-13} with a zero initial guess. The inexpensive preconditioner for operators of the form $(\alpha I - \delta t\, \mathcal{L})$ is taken as a single iteration of a classical AMG method from the hypre library [36], as interfaced through MFEM [1]. Specifically, classical interpolation (type 0) is used, Falgout coarsening (type 6) with a strength tolerance $\theta_C = 0.25$, zero levels of aggressive coarsening, and L_1-Gauss–Seidel relaxation (type 8).

In the top row of Figure 5.4, discretization errors are shown for different FIRK methods. Expected asymptotic convergence rates (black dashed lines in the top row) are observed for all discretizations, except for A–SDIRK(4). A–SDIRK(4) appears to be converging with a rate closer to three than four; however, further decreasing δt (not shown here) confirms 4th-order convergence is achieved eventually. This verifies that our implementation as provided in [89] is indeed accurately solving the underlying FIRK stage equations at each time step. These results also demonstrate the very high accuracy that may be achieved with FIRK methods, which again, is a key motivation for their use in practice.

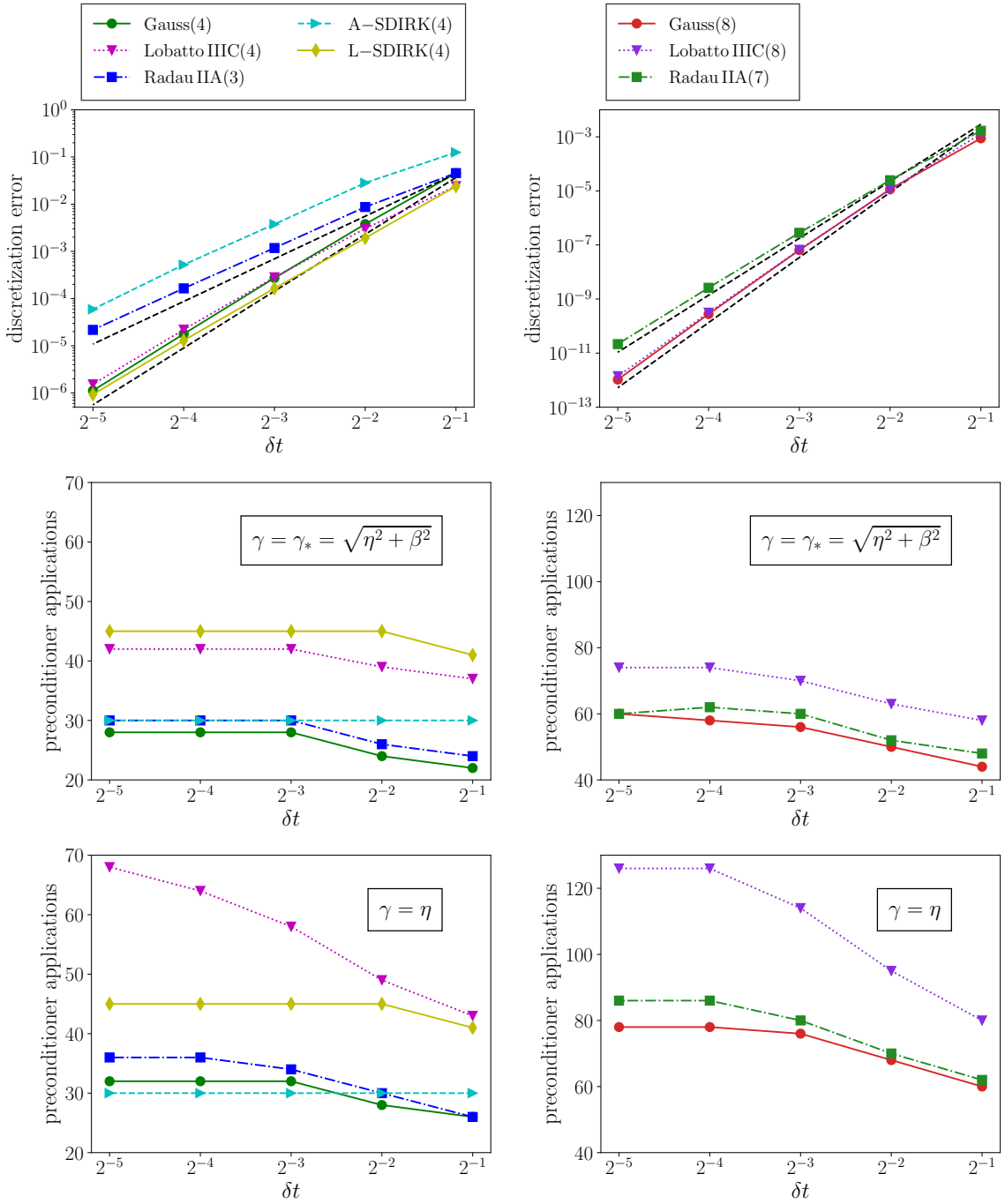

FIGURE 5.4: Advection-diffusion problem (5.135) for discretizations of 3rd and 4th order (**left column**), and 7th and 8th order (**right column**). **Top row:** L_∞-discretization errors at $t = 2$ as a function of time-step δt. Black, dashed lines with slopes of three and four are shown (left), as are those with slopes of seven and eight (right) indicating expected asymptotic convergence rates. **Middle/bottom rows:** Total number of preconditioner applications (i.e., number of AMG iterations) per time step to solve the FIRK stage equations averaged across all time steps. Results using the optimal preconditioning constants $\gamma_i = \gamma_{i*} = \sqrt{\eta_i^2 + \beta_i^2}$ are shown in the **middle row**, and the naive choice of preconditioning with $\gamma_i = \eta_i$ is shown on the **bottom row**.

Shown also in Figure 5.4 is the average number of total preconditioner applications needed

TABLE 5.3: Average number of GMRES iterations per time step for the linear system corresponding to the single complex-conjugate eigenvalue pair $\eta \pm \mathrm{i}\beta$ for Gauss(4), Radau IIA(3), and Lobatto IIIC(4). The number of iterations is shown for the naive preconditioning constant $\gamma = \eta$ and the optimal choice $\gamma = \gamma_* = \sqrt{\eta^2 + \beta^2}$. This data is associated with the $\delta t = 2^{-5}$ test problem shown in Figure 5.4. Recall each GMRES iteration for a complex-conjugate pair system requires two applications of the AMG preconditoner.

	Gauss(4)	Radau IIA(3)	Lobatto IIIC(4)
β^2/η^2	0.33	0.50	2.21
$\gamma = \eta$	16	18	29
$\gamma = \gamma_*$	14	15	16

per time step to solve the FIRK stage equations (i.e., the total number of AMG iterations). Overall, the solver appears robust with respect to mesh and problem size, since the average number of preconditioner applications per time step remains more-or-less constant as the the mesh is refined in most cases. The optimal preconditioning coefficient of $\gamma = \gamma_* = \sqrt{\eta^2 + \beta^2}$ (middle row) typically results in less total preconditioner applications per time step than the naive choice of $\gamma = \eta$ (bottom row). The difference in preconditioner applications is more pronounced for the higher-order schemes in the right column. This behaviour is not surprising, however, since the theory of Sections 5.2 and 5.4 showed that the condition number of the preconditioned operator grows with higher-order FIRK methods due to their having increased values of the ratio β_i/η_i (see also Table 5.1). There is no difference in the number of preconditioner applications between γ_* and η for the SDIRK schemes which is to be expected because all of the eigenvalues of A_0^{-1} are real (i.e., they do not occur in complex-conjugate pairs).

Considering the methods in the middle panel of the left column of Figure 5.4, the L–SDIRK(4) method requires the most preconditioner applications of all methods. A–SDIRK(4) requires far fewer preconditioner applications than L–SDIRK(4), but has a significantly larger discretization error than the other 4th-order schemes, and takes much longer to reach its asymptotic convergence rate. Thus, in terms of work done per accuracy, the 4th-order Gauss FIRK method is the clear winner for this particular test problem.

TABLE 5.4: Similar to Table 5.3, except data is shown for the high-order methods of Gauss(8), Radau IIA(7), and Lobatto IIIC(8), of which each have two complex-conjugate linear systems corresponding to $\eta_1 \pm \mathrm{i}\beta_1$ and $\eta_2 \pm \mathrm{i}\beta_2$. For each method, GMRES iterations are shown for both systems using the naive choices of $\gamma_1 = \eta_1, \gamma_2 = \eta_2$ and the optimal choices of $\gamma_1 = \gamma_{1*} = \sqrt{\eta_1^2 + \beta_1^2}, \gamma_2 = \gamma_{2*} = \sqrt{\eta_2^2 + \beta_2^2}$.

		Gauss(8)		Radau IIA(7)		Lobatto IIIC(8)	
β_1^2/η_1^2	β_2^2/η_2^2	0.09	1.60	0.13	3.51	0.38	4.88
$\gamma_1 = \eta_1$	$\gamma_2 = \eta_2$	14	25	15	28	17	41
$\gamma_1 = \gamma_{1*}$	$\gamma_2 = \gamma_{2*}$	14	16	14	16	15	17

To better illuminate the advantages of using γ_* vs. η, the number of GMRES iterations for only complex-conjugate pair linear systems is shown in Tables 5.3 and 5.4 for the largest problem size of $\delta t = 2^{-5}$. In all cases using γ_* over η results in fewer iterations (except for Gauss(8) when $\beta^2/\eta^2 = 0.09$ for which iterations remain constant). The most extreme example is Lobatto IIIC(8) with $\beta^2/\eta^2 = 4.88$ which results in $\approx 2.5\times$ fewer iterations. In general it seems as though systems with larger ratios of β^2/η^2 require more iterations to converge, but also yield larger savings when using γ_* vs. η. Both of these results are consistent with the theoretical analyses, which predicted that the condition numbers increase as a function of β^2/η^2. In particular, the eigenvalue analyses showed that when using γ_* over η, the reduction one sees in the condition number increases with β^2/η^2 (e.g., see Figures 5.1 and 5.2).

Recall that the preconditioning theory of Section 5.4 showed that over the space of all spatial discretizations satisfying Assumption 5.2, it is skew-symmetric matrices that yield the poorest conditioning (in the worst case sense). Furthermore, the eigenvalue analysis in Section 5.2 showed that using γ_* over η leads to greater reductions in condition number for skew-symmetric matrices than it does for symmetric definite ones. Therefore the test problem here is not particularly challenging in the sense that the linear systems become SPD in the limit that $h \to 0$ because the symmetric negative semi-definite diffusion discretization scales as $\mathcal{O}(h^{-2})$ and therefore dominates the skew-symmetric advection discretization that scales as $\mathcal{O}(h^{-1})$.[11]

5.5.2 The nonlinear setting

This section considers numerical tests for the nonlinear FIRK algorithm from Section 5.3. Much of the discussion from the previous section on numerical results in the linear setting carries over to the current one. The nonlinear test problem is chosen as the following viscous Burgers equation in two spatial dimensions:

$$u_t + (0.85u^2)_x + (u^2)_y = 0.3u_{xx} + 0.25u_{yy} + s(x,y,t), \quad (x,y,t) \in (-1,1)^2 \times (0,2]. \tag{5.136}$$

The PDE is posed on a periodic spatial domain, and the source term $s(x,y,t)$ is chosen such that the solution of the PDE is $u(x,y,t) = \sin^4(\pi/2[x-1-0.85t])\sin^4(\pi/2[y-1-t])\exp(-[0.3+0.25]t)$. Numerical tests will consider the same combinations of FIRK methods and finite-difference spatial discretizations as were used in the previous linear test problems. Again, the time-step is taken to be $\delta t = 2h$. The same problem sizes of $n_x \times n_y$

[11] In fact, an example in our article [91, Tab. 3] for a highly non-symmetric problem arising from the discretization of a pure advection problem showed an almost $6\times$ reduction in iteration count for the hardest Lobatto IIIC(8) system (i.e., the one with $\beta^2/\eta^2 = 4.88$ in Table 5.4).

will be considered as in Section 5.5.1, with the exception of the smallest $2^3 \times 2^3$ problems being omitted. The finite-difference spatial discretizations are again implemented in a `C++` class that is derived from MFEM's `Operator` class. The implementation used for the diffusion terms in the linear PDE (5.135) is again used for those in the nonlinear PDE (5.136). However, the implementation for the advection terms is different from the linear setting since they are now nonlinear. Specifically, when the action of the nonlinear advection discretization needs to be computed (which is required to compute the nonlinear residual in Newton's method), or it needs to be linearized, the communication of data at processor boundaries is done using additional MPI send and receive calls. Once the operator has been linearized, it is again stored as a `HypreParMatrix`, such that any MPI communication done during matrix-vector products is handled by hypre [36]. As previously, all tests use four cores in space.

The simple Newton method from Section 5.3.1 is applied to solve the nonlinear FIRK stage equations that arise at each time step. The Newton method is iterated until the norm of the relative residual of the nonlinear system falls below 10^{-10}, and a zero initial guess is used for the stage vectors. Recall that at every Newton iteration, a block upper triangular Jacobian system is solved via block backward substitution, which requires inversion of its diagonal blocks. The individual diagonal blocks are of the form $\zeta_i I - \delta t\, \mathcal{L}$ if they correspond to a real eigenvalue ζ_i of A_0^{-1}, or of the form $\begin{bmatrix} \eta_i I - \delta t\, \mathcal{L} & \phi_i I \\ -\beta_i^2/\phi_i I & \eta_i I - \delta t\, \mathcal{L} \end{bmatrix}$ if they correspond to a complex-conjugate eigenvalue pair $\eta_i \pm \mathrm{i}\beta_i$ of A_0^{-1}. In both cases, a preconditioned Krylov method is used to solve the 1×1 or 2×2 block system. Here GMRES(30) is used with a relative-residual stopping tolerance of 10^{-4} and a zero initial guess. Such a loose stopping tolerance is applied to avoid drastically over-solving the Jacobian systems since this typically does not accelerate convergence of the (outer) Newton iteration.

The preconditioner for systems of the first kind is again a single iteration of a classical AMG method from hypre [36] applied to the operator $(\zeta_i I - \delta t\, \mathcal{L})$. The preconditioner for the second type is the lower-triangular, Schur complement-based approximation $\begin{bmatrix} \eta_i I - \delta t\, \mathcal{L} & 0 \\ -\beta_i^2/\phi_i I & \gamma_i I - \delta t\, \mathcal{L} \end{bmatrix}^{-1}$, where γ_i is some constant (e.g., the optimal constant $\gamma_i = \gamma_{i*} = \eta_i + \beta_i^2/\eta_i$, or the naive constant $\gamma_i = \eta_i$). Furthermore, when the action of this preconditioner is applied, a single iteration of a classical AMG method from hypre is used to approximate the inverses of the diagonal blocks. The same parameters used to construct the AMG preconditioners in Section 5.5.1 are also used here.

Numerical results are shown in Figure 5.5. The left panel shows the discretization errors for the 3rd- and 4th-order methods, while the right shows the discretization errors for the 7th- and 8th-order methods. All of the FIRK methods appear to be converging with the correct orders of accuracy. The 4th-order SDIRK methods appear to be converging

TABLE 5.5: Average number of GMRES iterations per time step (i.e., per Newton solve) for linearized systems corresponding to the complex-conjugate pair $\eta \pm \mathrm{i}\beta$ for Gauss(4), Radau IIA(3), and Lobatto IIIC(4). The number of iterations is shown for the naive preconditioning constant $\gamma = \eta$ and the optimal choice $\gamma = \gamma_* = \eta + \beta^2/\eta$. This data is associated with the $\delta t = 2^{-5}$ nonlinear Burgers test problem shown in Figure 5.5.

	Gauss(4)	Radau IIA(3)	Lobatto IIIC(4)
β^2/η^2	0.33	0.50	2.21
$\gamma = \eta$	20	25	41
$\gamma = \gamma_*$	20	21	29

not quite with order four, with A–SDIRK(4) converging with a rate much closer to three (recall A–SDIRK(4) behaved similarly in the linear tests; see Figure 5.4). This verifies that our implementation of the nonlinear FIRK algorithm provided in [89] is functioning as expected, as are the finite-difference spatial discretizations.

Considering the scalability of the solver with respect to problem size is now more complicated than in the linear FIRK setting because it involves both the scalability of the Newton solver and the scalability of the linear solver used to solve the Jacobian systems. Since no analysis was done for the Newton method itself, let us just look at the total number of GMRES iterations for one particular problem size to assess the correctness of the linear preconditioning theory from Section 5.3.3 and Section 5.4. In particular, let us consider the total number of 2×2 block GMRES iterations per time step or Newton solve (i.e., the cumulative number of 2×2 GMRES iterations over all iterations of a Newton solve)

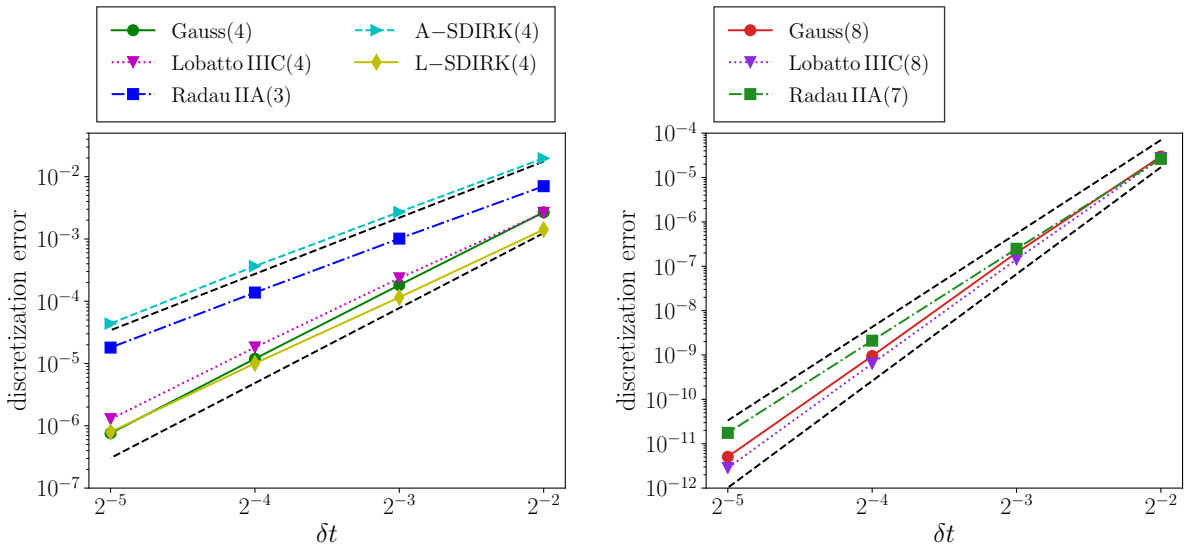

FIGURE 5.5: L_∞-discretization errors at $t = 2$ as a function of time-step δt for the nonlinear viscous Burgers problem (5.136). **Left:** 3rd- and 4th-order methods, with the black, dashed lines having slopes three and four to indicate expected asymptotic convergence rates. **Right:** 7th- and 8th-order methods, with the black, dashed lines having slopes seven and eight to indicate expected asymptotic convergence rates.

TABLE 5.6: Similar to Table 5.3, except data is shown for the high-order methods of Gauss(8), Radau IIA(7), and Lobatto IIIC(8), of which each have two 2×2 complex-conjugate linear systems corresponding to $\eta_1 \pm \mathrm{i}\beta_1$ and $\eta_2 \pm \mathrm{i}\beta_2$. For each method, GMRES iterations are shown for both systems using the naive choices of $\gamma_1 = \eta_1, \gamma_2 = \eta_2$ and the optimal choices of $\gamma_1 = \gamma_{1*} = \eta_1 + \beta_1^2/\eta_1, \gamma_2 = \gamma_{2*} = \eta_2 + \beta_2^2/\eta_2$.

		Gauss(8)		Radau IIA(7)		Lobatto IIIC(8)	
β_1^2/η_1^2	β_2^2/η_2^2	0.09	1.60	0.13	3.51	0.38	4.88
$\gamma_1 = \eta_1$	$\gamma_2 = \eta_2$	18	34	17	38	24	52
$\gamma_1 = \gamma_{1*}$	$\gamma_2 = \gamma_{2*}$	17	25	17	27	22	31

averaged across all time steps within a given simulation. Tables 5.5 and 5.6 shows this data for the $\delta t = 2^{-5}$ solves used to produce the results in Figure 5.5. Total GMRES iterations are considered for the optimal choice of constant $\gamma = \eta + \beta^2/\eta$ and the naive choice of $\gamma = \eta$. Note that this is a fair comparison because when averaged across all time steps, these simulations each required five iterations per Newton solve. In all cases using the optimal constant results in the same or fewer iterations than when using $\gamma = \eta$. Moreover, more iterations are required for increasing β^2/η^2, which is consistent with the theory of Section 5.3.3 and Section 5.4 since it showed the condition number of the preconditioned Schur complement for these 2×2 systems increasing as a function of β^2/η^2.

As for the example in the linear setting, this problem is not particularly challenging from the point of view that the linearized spatial discretization becomes symmetric as $h \to 0$. It is likely that larger reductions in iteration count would be achieved using γ_* over η on a more non-symmetric problem.[12]

5.6 Conclusions

During each discrete time step, an s-stage implicit Runge-Kutta method requires the solution of a system of $Ns \times Ns$ non-symmetric, block-coupled, nonlinear algebraic equations. In the context of approximating the solution of a PDE through the method of lines, N is the number of spatial degrees-of-freedom, which is typically large, and therefore this system of equations is large. For DIRK methods, this system is only block coupled in a triangular sense, so that its solution can be obtained through a block backward/forward substitution requiring only the solution of $N \times N$ systems. Conversely, FIRK methods are fully block coupled and thus present a much greater challenge in practice, so much so that they are seldom used compared to DIRK methods.

[12]For example, a result in our article [90, Tab. 4] shows for a compressible fluid dynamics problem a total reduction greater than three for the combined iteration counts of the two Lobatto IIIC(8) systems when using γ_* over η (for reference, the combined reduction in Table 5.6 is $(24 + 52)/(22 + 31) \approx 1.4$).

Despite their difficulty to implement efficiently, FIRK methods are attractive for the numerical solution of PDEs since they may be of arbitrarily high order (e.g., $\mathcal{O}(2s)$), including high stage order, which is important for many stiff PDE problems. DIRK methods in contrast typically have much lower order for a comparable number of stages (e.g., $\mathcal{O}(s)$), and are limited to a stage order of one, which means they may only in practice achieve 1st-order accuracy for some stiff problems independent of their number of stages.

This chapter considered in detail the new algorithms for the efficient application of FIRK methods to both linear and nonlinear PDEs, respectively. Section 5.1 provides relevant information on Runge-Kutta methods as they are applied in the method of lines, including the associated systems of linear and nonlinear algebraic equations, and various algorithms that have been proposed in the literature for solving such equations. The key components of our linear and nonlinear algorithms proposed in are then given in Sections 5.2.1 and 5.3.1, respectively.

Detailed theoretical analysis has been conducted to assess the efficacy of the linear preconditioners used by these new algorithms. This analysis first considers using eigenvalues under the assumption that the (linearized) spatial discretization of the PDE is either symmetric definite (Sections 5.2.3 and 5.3.3), or skew symmetric (Section 5.2.4). Tight upper bounds on the condition numbers are established, and are then minimized through the optimal choice of a free constant appearing in the preconditioners. Condition numbers associated with the optimized preconditioners are found (in the worst cases) to grow weakly with the order/number of stages of the FIRK method at hand. The condition numbers remain $\mathcal{O}(1)$, however, for many high-order FIRK methods, such as the 10th-order Gauss method.

Following this, Section 5.4 develops more general theory that does not require the (linearized) spatial discretization to be symmetric definite or skew symmetric, but instead only assumes that its field of values is contained in the closed left half plane. Such an assumption can be seen as an effectively minimal requirement to ensure that the numerical integration of the associated (linearized) ODE system remains stable with an A-stable FIRK method. Optimized preconditioners are determined by minimizing the worst-case condition number that can occur for (linearized) spatial discretizations satisfying this assumption. The associated optimized condition numbers are again found to grow weakly with the order/number of stages of the FIRK method at hand. In fact, this theory recovers the optimized constants that were derived using eigenvalue analysis, which shows they were optimal in a much more general sense. Furthermore, this analysis is used to show that over the space of (linearized) spatial discretizations whose field of values lie in the closed left half plane, it is skew-symmetric matrices which result in the poorest conditioning of the preconditioned operators.

Finally, Section 5.5 provides detailed numerical examples for both linear and nonlinear PDEs that confirm the linear preconditioning theory developed earlier in the chapter. Some details of our `C++` software package [89] that implements these algorithms is also discussed.

Chapter 6

Conclusions and future work

Parallel-in-time integration has seen a significant increase in both interest and relevance with the development of massively parallel computers. For many problems, parallel-in-time methods have been shown capable of significantly reducing the (wall-clock) time to solution compared to the traditional approach of sequential time-stepping. Most often, this success has been for problems dominated by diffusive processes. Unfortunately, however, there has been little success for the parallel-in-time integration of hyperbolic problems, and advection-dominated problems more broadly. In this thesis, we have focused on the multigrid reduction-in-time method known as MGRIT, of which the two-grid variant is equivalent to Parareal. For this solver, coarse-grid operators defined through the standard approach of rediscretizing the fine-grid problem typically result in rapid convergence for diffusion-dominated problems, but divergence for those that are advection dominated. The overarching goal of this thesis has been to develop a better understanding of why this behaviour occurs for advection-dominated problems, and moreover, to develop new and efficient coarse-grid operators for their solution.

In Chapter 2, we investigated the convergence of MGRIT for the constant-coefficient linear advection problem. We identified two primary reasons as to why MGRIT diverges on this problem when rediscretizing on coarse grids. First, the coarse-grid operator must track characteristics, which it cannot if it uses a fixed stencil with respect to the time-step size, as explicit Eulerian discretizations do. Second, for fast convergence, coarse-grid operators must propagate spatial modes that slowly decay in time in a very similar way as the ideal coarse-grid operator does; however, rediscretized coarse-grid operators do not. Using these heuristics, we developed an optimization-based approach for defining coarse-grid operators for this problem. For explicit and implicit discretizations, of low- and high-orders of accuracy, we showed numerically that the optimized coarse-grid operators yield fast MGRIT convergence in just a handful of iterations. Moreover, for explicit discretizations,

parallel tests showed that the optimized coarse-grid operators lead to significant speed-ups over sequential time-stepping. These results represent the largest speed-ups recorded to date for a multigrid-in-time method applied to an advection-dominated problem.

In Chapter 3, using the tools of LFA, we derived a closed-form convergence theory for two-level MGRIT. By comparison with existing literature, we found that the approximations derived under the convergence theory are tight with respect to the number of time points. In addition, we applied our convergence theory to provide an alternative explanation for the poor convergence of MGRIT on advection-dominated problems. Specifically, we identified that a rediscretized coarse-grid operator provides an inadequate coarse-grid correction to characteristic components.

The work in Chapter 3 leaves open many possible areas for future research regarding generalizations of our LFA theory. For example, numerical tests for some diffusive problems have shown that employing spatial coarsening within MGRIT does not significantly slow convergence compared to coarsening in time only [35, 57]. However, the same behaviour does not appear to hold for advection problems, where convergence can significantly deteriorate [57, 58]. There does not yet exist an explanation for why this is the case. Therefore, it would be interesting if our LFA theory could be extended to account for spatial coarsening within the MGRIT algorithm, since this would undoubtedly shed light on how spatial coarsening could (or could not) be used effectively for advection problems. Our LFA theory was derived only for two-level MGRIT methods; in practice, however, one often wants to use a multilevel method. More broadly in the context of the multigrid solution of advection-dominated problems, it is common that convergence degrades substantially in moving from two to multiple levels [10, 108, 109, 58]. For such cases, three-level LFA can be applied to develop a better understanding of convergence [104]. It remains an open question whether or not our closed-form LFA can be extended to three levels. Undoubtedly, such an extension would provide significant insight into developing multilevel solvers.

In Chapter 4, based on our findings from the previous chapters, we developed effective coarse-grid operators for semi-Lagrangian discretizations of advection-dominated problems. Our coarse-grid operator is designed to ensure that it tracks characteristics, and that it closely approximates the ideal coarse-grid operator on smooth spatial modes. For the special case of constant-coefficient advection, we theoretically proved that the coarse-grid operator is stable for all problem parameters. Moreover, for a wide range of problems, including those with variable wave-speeds, and those using high-order discretizations, we showed numerically that the coarse-grid operator yields fast MGRIT convergence. More broadly, our coarse-grid operator represents a significant advancement in the parallel-in-time solution of advection-dominated problems.

The work in Chapter 4 also leaves open many topics for future research. A key next step involves the implementation of the coarse-grid operators using the XBraid package [107] so that parallel scaling studies can be run. Further investigation into the efficiency of iterative solvers for approximately inverting the backward Euler matrix that forms part of the coarse-grid operator is also warranted. We also found empirically that our approach did not work for discretizations whose dominant error is dispersive rather than dissipative. Future studies should attempt to develop a better understanding of why this is the case and try to rectify it. A key focus of our future work will be extending the coarse-grid operator so that it can be used in conjunction with fine-grid discretizations that are Eulerian in nature, rather than semi-Lagrangian. It will also be interesting to investigate the extension of these ideas to nonlinear PDEs, including those with both smooth and discontinuous solutions.

Rather than focusing on parallel-in-time integration, Chapter 5 instead studied sequential time integration. Specifically, we presented new algorithms for the FIRK solution of both linear and nonlinear ODEs that arise from the method-of-lines solution of PDEs. FIRK methods may have excellent stability and accuracy properties, and are particularly well suited to the solution of stiff problems. However, they do not see wide-spread use in the context of PDEs due to the difficultly of numerically solving the associated fully coupled algebraic stage equations. We presented new algorithms for solving these algebraic equations, with our approach centred on strategically preconditioning linear systems associated with complex-conjugate eigenvalues of the Butcher tableau matrix. We presented detailed theoretical results showing the optimality of our proposed preconditioning strategy. The first of such results were based on eigenvalue analysis, under assumptions of symmetry or skew symmetry on the spatial discretization. The second set of results generalized the earlier theory to the case of all spatial discretizations with field of values contained in the closed left-half plane. Numerical results confirmed the accuracy of our theoretical developments, and showed that our algorithms can outperform other implicit integration methods that are widely used, such as DIRK schemes, for example. One possible direction for future research that closely relates to the rest of this thesis is the parallelization-in-time of our algorithm. That is, computing all FIRK stages, or complex-conjugate pairs of stages, in parallel.

Our ever expanding desire to solve large-scale problems continues to drive the development of scalable, fast, and highly parallelizable algorithms. While limited success has been previously obtained for parallel-in-time solutions of advection-dominated problems, the work in this thesis has led to significant improvements in our understanding of the shortcomings of these algorithms. Importantly, these insights provide new and promising directions for successful parallel-in-time solvers, and provide optimism for the future of this field.

Appendix A

Additional materials from Chapter 2

A.1 Runge-Kutta Butcher tableaux

For completeness, here we provide Butcher tableaux for some Runge-Kutta methods that are commonly used throughout the thesis. Explicit Runge-Kutta (ERK) methods are given in Tables A.1 and A.2, and L-stable singly diagonally implicit Runge-Kutta (SDIRK) methods are given in Tables A.2 and A.3.

Table A.1: Butcher tableaux for ERK methods of orders 1–4.

ERK1

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

ERK2 [52, (9.7)]

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

ERK3 [52, (9.8)]

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array}$$

ERK4 [14, p. 180]

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

TABLE A.2: Butcher tableaux for order 5 ERK (**left**), and order 4 L-stable SDIRK (**right**).

ERK5 [14, (236a)]

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
\frac{3}{4} & \frac{3}{16} & -\frac{3}{8} & \frac{3}{8} & \frac{9}{16} & 0 & 0 \\
1 & -\frac{3}{7} & \frac{8}{7} & \frac{6}{7} & -\frac{12}{7} & \frac{8}{7} & 0 \\
\hline
 & \frac{7}{90} & 0 & \frac{32}{90} & \frac{12}{90} & \frac{32}{90} & \frac{7}{90}
\end{array}
$$

SDIRK4 [48, (6.16)]

$$
\begin{array}{c|ccccc}
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\
\frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{11}{20} & \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & 0 & 0 \\
\frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} & 0 \\
1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\
\hline
 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4}
\end{array}
$$

TABLE A.3: Butcher tableaux for L-stable SDIRK methods of orders 1–3. Constants used in SDIRK3 are: $\zeta = 0.43586652150845899942\ldots$, $\alpha = \frac{1+\zeta}{2}$, $\beta = \frac{1-\zeta}{2}$, $\gamma = -\frac{3}{2}\zeta^2 + 4\zeta - \frac{1}{4}$, $\epsilon = \frac{3}{2}\zeta^2 - 5\zeta + \frac{5}{4}$.

SDIRK1

$$
\begin{array}{c|c}
1 & 1 \\
\hline
 & 1
\end{array}
$$

SDIRK2 [14, p. 261]

$$
\begin{array}{c|cc}
1-\frac{\sqrt{2}}{2} & 1-\frac{\sqrt{2}}{2} & 0 \\
1 & \frac{\sqrt{2}}{2} & 1-\frac{\sqrt{2}}{2} \\
\hline
 & \frac{\sqrt{2}}{2} & 1-\frac{\sqrt{2}}{2}
\end{array}
$$

SDIRK3 [14, p. 262]

$$
\begin{array}{c|ccc}
\zeta & \zeta & 0 & 0 \\
\alpha & \beta & \zeta & 0 \\
1 & \gamma & \epsilon & \zeta \\
\hline
 & \gamma & \epsilon & \zeta
\end{array}
$$

A.2 A nonlinear approximation of Ψ_{ideal}

In Section 2.5, coarse-grid time-stepping operators were sought through a linear least squares procedure that used heuristics based on convergence theory (see Section 2.4.1). To better understand how accurate this heuristic-driven approach was, in this section, we formulate and solve a nonlinear optimization problem that approximately minimizes error estimates (2.6) in a more direct way.

We seek a coarse-grid time-stepping operator Ψ that minimizes estimates (2.6) for the coarse-grid MGRIT error propagation matrix $\mathcal{E}$:

$$
\Psi := \underset{\widehat{\Psi}\in\mathbb{R}^{n_x\times n_x}}{\arg\min} \left\|\mathcal{E}\left(\widehat{\Psi}\right)\right\|_2^2 = \underset{\widehat{\Psi}\in\mathbb{R}^{n_x\times n_x}}{\arg\min} \max_{1\le k\le n_x} \left\|\mathcal{E}_k\left(\widehat{\Psi}\right)\right\|_2^2. \tag{A.1}
$$

Such minimax problems are, in general, difficult to solve given their non-smoothness. For this reason, we approximate (A.1) with a smoother problem, in which the max-norm is

replaced with the two-norm. This yields the following nonlinear least squares problem

$$\Psi := \underset{\widehat{\Psi}\in\mathbb{R}^{n_x\times n_x}}{\arg\min} \frac{1}{n_x}\sum_{k=1}^{n_x} \left\|\mathcal{E}_k\left(\widehat{\Psi}\right)\right\|_2^2. \tag{A.2}$$

To solve this problem, we use MATLAB's nonlinear least squares routine, `lsqnonlin`, which employs the well-known Levenberg–Marquardt algorithm. Again, we only focus on discretizations of (2.1) with periodic boundary conditions, such that the underlying convergence theory of [28] applies. As previously, this also means that the time-stepping operators' eigenvalues—which are required for evaluation of the objective function in (A.2)—are inexpensive to compute, and are linearly related to the entries in the underlying circulant matrices via the DFT.

Here, we consider the solution of (A.2) for ERK discretizations of (2.1). The default settings are used for MATLAB's `lsqnonlin`, except that a maximum of 30 nonlinear iterations is permitted.[1] The sparsity patterns for Ψ used in Section 2.5.3 are also applied here since they were successful previously. Similarly, since the solution of weighted linear least squares problem (2.8) was successful in Section 2.5, it is passed to `lsqnonlin` as the initial iterate in all instances. MGRIT iteration counts obtained using the resulting coarse-grid operators are given in Table A.4.

[1] One exception here is for ERK2+U2 with $m = 64$ where the solutions generated resulted in an MGRIT solver whose convergence stalled. Permitting `lsqnonlin` to use only 10 iterations in this instance appears to resolve this issue.

Table A.4: Two-level iteration counts for ERK discretizations with Ψ as nonlinear least squares solution (A.2).

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

The iteration counts from Table A.4 are similar to those in Table 2.3, where Ψ was the solution of the simpler, heuristic-based linear least squares problem (2.8). In most cases, the iteration counts in the two tables are almost identical. While the iteration counts for the two optimization formulations result in the same or similar iteration counts, upon closer inspection (not shown here for brevity) they appear to not be converging to the same solution, in general. This suggests that MGRIT convergence is perhaps not very sensitive to the precise optimization problem solved. In any event, that the iteration counts are similar for the two approaches indicates that the heuristics developed in Section 2.4.1 are accurate and that they are properly captured by the approach pursued in Section 2.5.

A.3 Parallel results: Two-level solvers

In the setting of two time grids, two effects have to be balanced for optimizing parallel performance. On the one hand, aggressive coarsening with $m \gg 2$ reduces the number of coarse-grid points and, thus, the cost of the sequential coarse-grid solve. On the other hand, when using a large coarsening factor, relaxation on the fine grid is performed sequentially over a larger time interval, that is, for more time points. Typically, the fastest runtimes on a given number of processors have been obtained when using a coarsening factor such that the number of coarse-grid points is equal to the number of processors.

Figure A.1 shows compute times of two-level MGRIT for $m = 64$ coarsening, with dashed lines representing runtimes of sequential time-stepping for reference purposes. On smaller numbers of processors, time-stepping is faster, demonstrating the computational overhead of the MGRIT approach. This extra work, however, can be effectively parallelized at

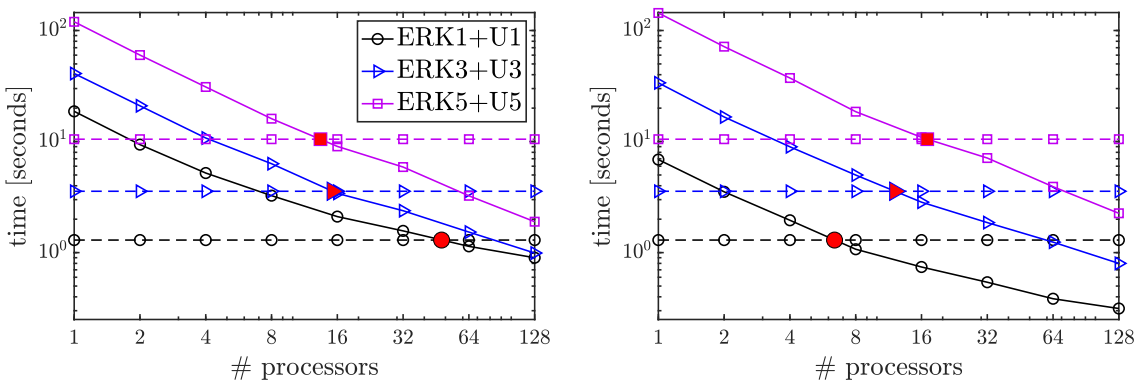

FIGURE A.1: Strong parallel scaling: Runtimes of two-level MGRIT with $m = 64$ coarsening and using time-only parallelism for ERKp+Up discretizations on space-time grids of size $n_x \times n_t = 2^{12} \times (2^{14}, 2^{13}, 2^{13})$ for $p = (1, 3, 5)$. **Left:** Fixed residual stopping tolerance 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.

higher processor counts with good parallel scalability. The crossover point at which it becomes beneficial to use MGRIT over sequential time-stepping is between eight and 64 processors, depending on the discretization and on the stopping criterion. For ERK1+U1, for example, when solving to high accuracy, the crossover point is at 64 processors, while using only eight processors results in a faster compute time than sequential time-stepping for achieving discretization error accuracy. In both settings, the largest speed-up achieved over sequential time-stepping is at 128 processors, where two-level MGRIT is faster by a factor of about 1.4, 3.6, and 5.5 when solving to a residual tolerance of 10^{-10}, and speed-ups of about 4.1, 4.5, and 4.7 when solving up to discretization error (for the discretizations in the order of increasing accuracy). Note that, considering $m = 64$ coarsening and 2^{13} time steps on the fine grid as for the discretizations of orders three and five, the coarse grid consists of 128 time points corresponding to the number of processors for which the largest speed-up is achieved.

Appendix B

Some theoretical results from Chapter 4

B.1 An alternative coarse-grid operator

In Section 3.7, we identified that the culprit for poor MGRIT convergence on advection-dominated problems is that the coarse-grid correction fails to adequately treat characteristic error components. This analysis was inspired by [108], in which it was shown that an analogous problem occurs in the multigrid solution for steady state advection-dominated problems. The solution to obtaining a better coarse-grid correction for characteristic components is to have the truncation error of the coarse-grid operator match more accurately that of the ideal coarse-grid operator. Indeed, this is the reasoning behind the coarse-grid operators proposed in Section 4.2.2. An alternative to the coarse-grid operators we proposed earlier is a coarse-grid operator that itself is a linear combination of two coarse-grid operators, as proposed in [108, Sec. 5.1] for steady state advection problems. We consider this idea here (as briefly discussed in Remark 4.3) in the context of semi Lagrangian discretizations.

Recall from Lemma 4.1 that, for constant-wave-speed advection, a semi Lagrangian scheme using degree (at most) p interpolating polynomials has a truncation error given by

$$u(\boldsymbol{x}, t_{n+1}) - \mathcal{S}_p^{(\delta t)} u(\boldsymbol{x}, t_n) = (-h)^{p+1} f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+1}) + \mathcal{O}(h^{p+2}), \tag{B.1}$$

where, f_{p+1} is the degree $p+1$ polynomial defined by

$$f_{p+1}(z) := \frac{1}{(p+1)!} \prod_{q=-\ell(p)}^{r(p)} (q+z). \tag{B.2}$$

Further recall from Lemma 4.1 that the truncation error of the associated ideal coarse-grid operator is

$$u(\boldsymbol{x}, t_{n+m}) - \left[\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}\right] u(\boldsymbol{x}, t_n) = (-h)^{p+1} m f_{p+1}\big(\varepsilon^{(\delta t)}\big) \frac{\partial^{p+1}}{\partial x^{p+1}} u(\boldsymbol{x}, t_{n+m}) + \mathcal{O}(h^{p+2}). \tag{B.3}$$

Now we consider a coarse-grid operator of the form

$$\Phi^{(m\delta t)} = \beta_p \mathcal{S}_p^{(m\delta t)} + \beta_q \mathcal{S}_q^{(m\delta t)}, \quad q > p. \tag{B.4}$$

The idea is to choose the linear weights β_p and β_q so that the truncation error of (B.4) provides a closer approximation to that in (B.3) than either $\mathcal{S}_p^{(m\delta t)}$ or $\mathcal{S}_q^{(m\delta t)}$ can provide on their own. Using the truncation error formula (B.1), it is not very difficult to show that this leads to weights given by

$$\beta_p = m \frac{f_{p+1}\big(\varepsilon^{(\delta t)}\big)}{f_{p+1}\big(\varepsilon^{(m\delta t)}\big)}, \quad \beta_q = 1 - \beta_p, \tag{B.5}$$

providing that $f_{p+1}\big(\varepsilon^{(m\delta t)}\big) \neq 0$. Recall that $\varepsilon^{(m\delta t)} \in [0, 1)$ is the distance from the coarse-grid departure point to its east neighbour, and that $f_{p+1}\big(\varepsilon^{(m\delta t)}\big) = 0$ only when the departure point coincides with a mesh point, $\varepsilon^{(m\delta t)} = 0$ (see Lemma B.1).

We have now encountered the first issue with the operator (B.4): If coarse-grid departure points coincide with mesh points, then (B.4) has zero truncation error. Therefore, if fine-grid departure points do not coincide with mesh points, while the coarse-grid departure points do, then (B.4) cannot possibly capture the non-zero truncation error of $\prod_{k=0}^{m-1} \mathcal{S}_p^{(\delta t)}$. Note that this situation can certainly arise in practice; recall from (4.35) that $\varepsilon^{(m\delta t)} = m\varepsilon^{(\delta t)} - \lfloor m\varepsilon^{(\delta t)} \rfloor$, and thus $\varepsilon^{(m\delta t)} = 0$ whenever m is even and $\varepsilon^{(\delta t)} = \frac{1}{2}$, for example. The fact that the truncation error of a coarse-grid semi Lagrangian scheme can vanish when that of the ideal coarse-grid operator does not is what motivated the alternative operators we proposed in Section 4.2.2.

A second problem for the coarse-grid operator (B.4) is that of stability. For example, consider the following bound on its norm

$$\begin{aligned} \big\|\Phi^{(m\delta t)}\big\|_2 &= \big\|\beta_p \mathcal{S}_p^{(m\delta t)} + (1-\beta_p)\mathcal{S}_q^{(m\delta t)}\big\|_2, \\ &\leq \big\|\beta_p \mathcal{S}_p^{(m\delta t)}\big\|_2 + \big\|(1-\beta_p)\mathcal{S}_q^{(m\delta t)}\big\|_2 = |\beta_p| + |1-\beta_p|, \end{aligned} \tag{B.6}$$

with the last expression following from the stability of $\mathcal{S}_p^{(m\delta t)}$ and $\mathcal{S}_q^{(m\delta t)}$ (see Assumption 4.1). While the upper bound of $|\beta_p| + |1-\beta_p|$ is not necessarily tight, it is instructive

to consider. It states that the operator is definitely stable if the weights are convex, that is, $\beta_p \in [0,1]$, and it hints at the idea that the operator may be unstable if $|\beta_p| \gg 1$. Indeed, numerical tests (not shown here for the sake of brevity) reveal that $|\beta_p| \gg 1$ certainly can occur, and that when it does it often leads to $\left\|\Phi^{(m\delta t)}\right\|_2 > 1$. Considering the form of β_q in (B.5), there are two ways in which $|\beta_p| \gg 1$ can occur. First, if $\varepsilon^{(\delta t)}$ is bounded away from 0 and 1, and $\varepsilon^{(m\delta t)} \to 0$ or $\to 1$, then the weight blows up. However, this is the same problem we discussed previously, just with coarse-grid departure points now becoming arbitrarily close to mesh points rather than exactly coinciding with them. A second and more interesting scenario occurs when both $\varepsilon^{(\delta t)}$ and $\varepsilon^{(m\delta t)}$ are bounded away from 0 and 1, then, from (B.5), $|\beta_p|$ scales as $\mathcal{O}(m)$. In other words, the difference between the truncation error (B.3) of the ideal operator and the linear combination operator (B.4) grows as $\mathcal{O}(m)$. Given our discussion in Section 4.2.4.1 regarding that the stencil of the coarse-grid operator needs to grow with m, and the fact that that of (B.4) does not, it is not surprising that the operator becomes unstable with increasing m. Note that the stability problem cannot be resolved by either taking a linear combination of more than two coarse-grid operators, or by using a multilevel solver that employs slow temporal coarsening.

B.2 Important properties of the polynomial $f_{p+1}(z)$

The following lemma describes the properties of the polynomial $f_{p+1}(z)$. The results are used in the proof of Lemma 4.5, which is given in Appendix B.3.

Lemma B.1 (Important properties of $f_{p+1}(z)$)**.** *Suppose that Assumption 4.1 holds, that is, p is odd, and let $f_{p+1}(z)$ be the degree $p+1$ polynomial as in (4.43):*

$$f_{p+1}(z) = \frac{1}{(p+1)!} \prod_{q=-\frac{p+1}{2}}^{\frac{p-1}{2}} (q+z). \tag{B.7}$$

Then,

1. $f_{p+1}(0) = f_{p+1}(1) = 0$,
2. $f_{p+1}(z)$ *is symmetric on* $[0,1]$ *about* $z = \frac{1}{2}$,
3. *when* $\frac{p+1}{2}$ *is odd, on* $(0,1)$*:* $f_{p+1}(z)$ *is negative, and has only a single critical point, which is a global minimum at* $z = \frac{1}{2}$,
4. *when* $\frac{p+1}{2}$ *is even, on* $(0,1)$*:* $f_{p+1}(z)$ *is positive, and has only a single critical point, which is a global maximum at* $z = \frac{1}{2}$.

Proof. To begin, it is useful to make the change of variables $z = \frac{1}{2}(1+\Delta)$ or $\Delta = 2z - 1$ in (B.7) and instead consider the function

$$f_{p+1}(z) = f_{p+1}\big(\tfrac{1}{2}(1+\Delta)\big) = \frac{1}{(p+1)!}\left(\frac{1}{2}\right)^{p+1}\mathcal{M}_p(\Delta), \quad \Delta \in [-1,1]. \tag{B.8}$$

where $\mathcal{M}_p(\Delta)$ is the product of monomials

$$\mathcal{M}_p(\Delta) := \prod_{q=-p,-p+2,\ldots}^{p} (q+\Delta), \tag{B.9}$$

$$= (-1)^{\frac{p+1}{2}}\Big[(p-\Delta)(p-2-\Delta)(\cdots)(1-\Delta)\Big]\Big[(1+\Delta)(\cdots)(p-2+\Delta)(p+\Delta)\Big]. \tag{B.10}$$

Observe from (B.10) that $\mathcal{M}_p(\Delta)$ has roots at both $\Delta = -1$ and $\Delta = 1$, and that $\mathcal{M}_p(\Delta)$ is symmetric about $\Delta = 0$. From (B.8), $f_{p+1}(z)$ therefore has roots at $z = 0$ and $z = 1$, and is symmetric on $[0,1]$ about $z = \frac{1}{2}$.

It is easy to see that the products of $\frac{p+1}{2}$ monomials contained in the closed parentheses in (B.10) are strictly positive when $\Delta \in (-1,1)$. Therefore, the sign of $\mathcal{M}_p(\Delta)$ is equal to that of $(-1)^{\frac{p+1}{2}}$ when $\Delta \in (-1,1)$. From (B.8), it follows that the sign of $f_{p+1}(z)$ on $(0,1)$ is equal to that of $(-1)^{\frac{p+1}{2}}$.

Since $\mathcal{M}_p(\Delta)$ is symmetric about $\Delta = 0$, it follows that $\Delta = 0$ is a local extremum. The $p+1$ roots of $\mathcal{M}_p(\Delta)$ are $\Delta = -p, -p+2, \ldots, 0, \ldots, p-2, p$, and since $\mathcal{M}_p(\Delta)$ is not constant, it must have at least one turning point between neighbouring roots. Since $\mathcal{M}_p'(\Delta)$ is a polynomial of degree p, it has at most p roots, and thus $\mathcal{M}_p(\Delta)$ can have only a single turning point between each pair of neighbouring roots. Thus, $\Delta = 0$ can be the only local extremum of $\mathcal{M}_p(\Delta)$ in the interval $\Delta \in (-1,1)$. From (B.8), it follows that $z = \frac{1}{2}$ is the location of the only local extrema of $f_{p+1}(z)$ on $(0,1)$. ■

B.3 Proof of Lemma 4.5: Important properties of γ_{p+1}

Proof. Throughout this proof, let us write $\varepsilon \equiv \varepsilon^{(\delta t)}$ for notational simplicity. For a particular problem, ε is a constant, but we want to establish properties of (4.36) that hold for all possible values of ε, and, so, in this analysis we will often treat ε as a continuous parameter in $[0,1)$.

First note that the arguments of f_{p+1} in (4.36) are in the interval $[0,1)$, and thus we need only examine the behaviour of f_{p+1} on this interval (as was done in Lemma B.1). Recall from Lemma B.1 that $f_{p+1}(0) = 0$, and thus it follows immediately that

$\gamma_{p+1}(0) = f_{p+1}(0) - mf_{p+1}(0) = 0$. Recall from Lemma B.1 that on $(0,1)$, the only dependence of f_{p+1} on the parity of $\frac{p+1}{2}$ is its sign. Moving forward through this proof, we assume $\frac{p+1}{2}$ is odd, since the properties for even $\frac{p+1}{2}$ follow immediately from the sign reversal of f_{p+1}, and thus the sign reversal of (4.36).

The argument $m\varepsilon - \lfloor m\varepsilon \rfloor$ in the first term of (4.36) is periodic on $\varepsilon \in [0,1)$ with m periods. Specifically, in each of the m subintervals $\varepsilon \in \left[\frac{2j}{2m}, \frac{2j+1}{2m}\right]$, $j \in \{0,\ldots,m-1\}$, it linearly increases from $0 \to \frac{1}{2}$. Furthermore, in each of the remaining m subintervals $\varepsilon \in \left[\frac{2j+1}{2m}, \frac{2j+2}{2m}\right)$, $j \in \{0,\ldots,m-1\}$, it linearly increases from $\frac{1}{2} \to 1^-$. Therefore, since f_{p+1} is symmetric over $[0,1)$ about $\frac{1}{2}$, it follows that $f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$ is periodic over $\varepsilon \in [0,1)$, with m periods. Combining this with the fact that $f_{p+1}(\varepsilon)$ is symmetric on $\varepsilon \in [0,1)$ about $\varepsilon = \frac{1}{2}$, (4.36) must be symmetric about $\varepsilon = \frac{1}{2}$. For this reason, we need only analyze $\gamma_{p+1}(\varepsilon)$ for $\varepsilon \in \left[0, \frac{1}{2}\right]$. See the examples in Figure B.1.

The periodic function $f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$ is non-positive on $\varepsilon \in (0, \frac{1}{2})$. More specifically, in the interval $\varepsilon \in (0,1)$, it has $m-2$ roots at $\varepsilon = \frac{j}{m}, j \in \{1,\ldots,m-1\}$, corresponding to $m\varepsilon - \lfloor m\varepsilon \rfloor = 0$. Furthermore, it is strictly decreasing on the m subintervals $\varepsilon \in \left(\frac{2j}{2m}, \frac{2j+1}{2m}\right)$, $j \in \{0,\ldots,m-1\}$, corresponding to where $m\varepsilon - \lfloor m\varepsilon \rfloor$ linearly increases from $0^+ \to \frac{1}{2}^-$. Finally, it is strictly increasing on the m subintervals $\varepsilon \in \left(\frac{2j+1}{2m}, \frac{2j+2}{2m}\right)$, $j \in \{0,\ldots,m-1\}$, corresponding to where $m\varepsilon - \lfloor m\varepsilon \rfloor$ linearly increases from $\frac{1}{2}^+ \to 1^-$. See the examples in Figure B.1.

Recall from (4.36) that $\gamma_{p+1}(\varepsilon) = -mf_{p+1}(\varepsilon) + f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$, and from Lemma B.1, that on $\varepsilon \in (0, \frac{1}{2})$, the function $-mf_{p+1}(\varepsilon)$ is positive and strictly increasing. Thus, for $\varepsilon \in (0, \frac{1}{2})$, $\gamma_{p+1}(\varepsilon)$ is the sum of the positive, strictly increasing function $-mf_{p+1}(\varepsilon)$, and the non-positive, periodic function $f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$. Therefore, it follows that the minimum of $\gamma_{p+1}(\varepsilon)$ over $\varepsilon \in (0, \frac{1}{2})$ must occur during the first subinterval in which $f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$ is decreasing, which, as described in the previous paragraph, is $\varepsilon \in \left(0, \frac{1}{2m}\right)$.

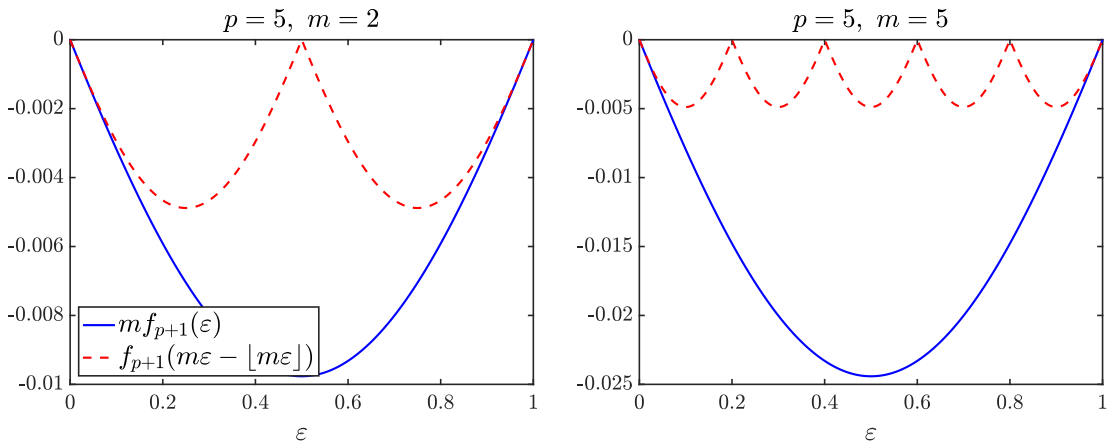

FIGURE B.1: Examples of the two functions whose difference is the function in question: $\gamma_{p+1}(\varepsilon) = -mf_{p+1}(\varepsilon) + f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$.

We now show that $\gamma_{p+1}(\varepsilon)$ is positive for $\varepsilon \in \left(0, \frac{1}{2m}\right)$. To begin, it is useful to re-express $f_{p+1}(z)$ from (4.43) by

$$f_{p+1}(z) = \frac{-z\left(\frac{p+1}{2} - z\right)}{(p+1)!} \prod_{\substack{q=-\frac{p-1}{2} \\ q\neq 0}}^{\frac{p-1}{2}} (q+z) = \frac{-z\left(\frac{p+1}{2} - z\right)}{(p+1)!} \prod_{q=1}^{\frac{p-1}{2}} (z^2 - q^2). \tag{B.11}$$

Substituting this into $\gamma_{p+1}(\varepsilon) = -mf_{p+1}(\varepsilon) + f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$, we have the following expressions and lower bounds when $\varepsilon \in \left(0, \frac{1}{2m}\right)$

$$\gamma_{p+1}(\varepsilon) = -mf_{p+1}(\varepsilon) + f_{p+1}(m\varepsilon), \tag{B.12}$$

$$= \frac{m\varepsilon}{(p+1)!}\left[\left(\tfrac{p+1}{2} - \varepsilon\right)\prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - \varepsilon^2\right) - \left(\tfrac{p+1}{2} - m\varepsilon\right)\prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - (m\varepsilon)^2\right)\right], \tag{B.13}$$

$$\begin{aligned} &= \frac{m\varepsilon}{(p+1)!}\left(\tfrac{p+1}{2} - \varepsilon\right)\left[\prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - \varepsilon^2\right) - \prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - (m\varepsilon)^2\right)\right] \\ &\quad + \frac{m\varepsilon}{(p+1)!}(m-1)\varepsilon\prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - (m\varepsilon)^2\right), \end{aligned} \tag{B.14}$$

$$> \frac{m\varepsilon}{(p+1)!}\left(\tfrac{p+1}{2} - \varepsilon\right)\left[\prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - \varepsilon^2\right) - \prod_{q=1}^{\frac{p-1}{2}}\left(q^2 - (m\varepsilon)^2\right)\right], \tag{B.15}$$

$$> 0. \tag{B.16}$$

To arrive at the inequality (B.15), note that the term dropped from (B.14) is positive because $m > 1$, and $q > m\varepsilon \in (0, \frac{1}{2})$. To arrive at the inequality (B.16), notice that $r^2 - \varepsilon^2 > r^2 - (m\varepsilon)^2 > 0$ when $\varepsilon \in \left(0, \frac{1}{2m}\right)$ for any $m > 1$.

By our previous arguments that the minimum of $\gamma_{p+1}(\varepsilon)$ over $\varepsilon \in \left(0, \frac{1}{2}\right)$ occurs on the subinterval $\varepsilon \in \left(0, \frac{1}{2m}\right)$, and that $\gamma_{p+1}(\varepsilon)$ is symmetric on $\varepsilon \in [0, 1)$, it therefore follows from the above result that the minimum of $\gamma_{p+1}(\varepsilon)$ on $\varepsilon \in (0, 1)$ is positive, and thus $\gamma_{p+1}(\varepsilon) > 0$, $\forall \varepsilon \in (0, 1)$.

Now let us move on to the second part of the proof, which is the bound (4.45) on the magnitude of $\gamma_{p+1}(\varepsilon)$ over $\varepsilon \in [0, 1)$ when the coarsening factor m is even. Recall from our arguments earlier in this proof that $-mf_{p+1}(\varepsilon)$ is positive on $\left(0, 1\right)$, and recall from Lemma B.1 that this function attains its maximum in this interval at $\varepsilon = \frac{1}{2}$. Recall also that $f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$ is non-positive for $\varepsilon \in \left(0, 1\right)$, and that when m is even it has a zero at $\varepsilon = \frac{1}{2}$. Since $\gamma_{p+1}(\varepsilon) = -mf_{p+1}(\varepsilon) + f_{p+1}(m\varepsilon - \lfloor m\varepsilon \rfloor)$, it follows that $\gamma_{p+1}(\varepsilon)$ attains

its global maximum on $[0,1)$ at the point $\varepsilon = \frac{1}{2}$:

$$\max_{\varepsilon\in[0,1)} \gamma_{p+1}(\varepsilon) = \gamma_{p+1}\big(\tfrac{1}{2}\big) = -m f_{p+1}\big(\tfrac{1}{2}\big), \quad \frac{m}{2} \in \mathbb{N}. \tag{B.17}$$

To evaluate $f_{p+1}\big(\frac{1}{2}\big)$, it is perhaps easiest to evaluate the function as it is given in (B.8) at $\Delta = 0$. This gives

$$\big|f_{p+1}\big(\tfrac{1}{2}\big)\big| = \frac{1}{(p+1)!}\left|\prod_{q=-p,-p+2,\dots}^{p} q\right|\left(\frac{1}{2}\right)^{p+1} = \frac{p!!p!!}{(p+1)!}\left(\frac{1}{2}\right)^{p+1} = \frac{p!!}{(p+1)!!}\left(\frac{1}{2}\right)^{p+1}, \tag{B.18}$$

in which $a!! = a\cdot(a-2)\cdot(a-4)\cdots 3\cdot 1$ for odd a, and $a!! = a\cdot(a-2)\cdot(a-4)\cdots 4\cdot 2$ for even a, is commonly known as the *double factorial* of $a \in \mathbb{N}$. Finally, to bound this fraction of double factorials as in the claimed bound of (4.45), observe for any $j \in \mathbb{N}$ the inequality $\frac{j}{j+1} < \sqrt{\frac{j}{j+2}}$ that follows from $j^2(j+2) < j(j+1)^2$. Applying this inequality to all of the following fractions except the first one gives

$$\frac{p!!}{(p+1)!!} = \frac{1}{2}\cdot\frac{3}{4}\cdot\frac{5}{6}\cdots\frac{p-2}{p-1}\cdot\frac{p}{p+1} \le \frac{1}{2}\cdot\sqrt{\frac{3}{5}}\cdot\sqrt{\frac{5}{7}}\cdots\sqrt{\frac{p-2}{p}}\cdot\sqrt{\frac{p}{p+2}} = \frac{1}{2}\sqrt{\frac{3}{p+2}}, \tag{B.19}$$

with equality holding only for the $p = 1$ case.

This completes the proof. ■

B.4 The exact solution of $\frac{\partial u}{\partial t} + \cos(2\pi x)\frac{\mathrm{d}\tau(t)}{\mathrm{d}t}\frac{\partial u}{\partial x} = 0$

We now derive the exact solution to a class of variable-wave-speed advection problems, which is used in Section 4.3 to measure the accuracy of certain semi Lagrangian discretizations.

Consider the variable-wave-speed advection problem

$$\frac{\partial u}{\partial t} + \cos(2\pi x)\frac{\mathrm{d}\tau(t)}{\mathrm{d}t}\frac{\partial u}{\partial x} = 0, \quad u(x,0) = u_0(x), \tag{B.20}$$

subject to periodic boundary conditions in space, and with $\tau(t)$ (or its derivative) some prescribed function. The Lagrangian formulation of this PDE is simply

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = \cos(2\pi\xi(t))\frac{\mathrm{d}\tau(t)}{\mathrm{d}t}, \tag{B.21}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t),t) = 0. \tag{B.22}$$

ODE (B.21) defines the characteristics $x(t) = \xi(t)$ of the PDE, and (B.22) simply states that the solution along characteristics is constant. Thus, *all* that needs to be done to calculate the solution to (B.20) is to solve (B.21) for the characteristics.

Considering (B.21), observe that the right-hand side vanishes whenever $\xi(t) = \frac{1}{4} + \frac{k}{2}$ for $k \in \mathbb{Z}$, which has the consequence that if $\xi_0 := \xi(0) \in \left[-\frac{1}{4} + \frac{k}{2}, \frac{1}{4} + \frac{k}{2}\right]$, then $\xi(t)$ will remain in this interval for all remaining time. Furthermore, if $\xi_0 = \frac{1}{4} + \frac{k}{2}$, then $\xi(t) = \xi_0$ (which follows from all other derivatives of $\xi(t)$ also vanishing at ξ_0). To this end, for $\xi_0 \neq \frac{1}{4} + \frac{k}{2}$, define the new dependent variable

$$y(t) := \xi(t) - \frac{k_0}{2} \in \left(-\frac{1}{4}, \frac{1}{4}\right), \quad \text{where} \quad k_0 = \left\lfloor 2\xi_0 + \frac{1}{2} \right\rfloor \in \mathbb{Z}. \tag{B.23}$$

Under this change of variable, ODE (B.21) becomes

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \cos(2\pi y)(-1)^{k_0} \frac{\mathrm{d}\tau(t)}{\mathrm{d}t}, \quad y(t) \in \left(-\frac{1}{4}, \frac{1}{4}\right). \tag{B.24}$$

Integrating both sides of this ODE with respect to t yields

$$\int \frac{1}{\cos(2\pi y)} \frac{\mathrm{d}y}{\mathrm{d}t}\, \mathrm{d}t = \int \frac{\mathrm{d}y}{\cos(2\pi y)} = (-1)^{k_0} \int \frac{\mathrm{d}f(t)}{\mathrm{d}t}\, \mathrm{d}t = (-1)^{k_0} f(t) + c, \tag{B.25}$$

for some arbitrary constant of integration c.

To integrate the left-hand side of (B.25), first invoke the following trigonometric identities

$$\frac{1}{\cos(2\pi y)} = \frac{1 + \tan^2(\pi y)}{1 - \tan^2(\pi y)} = \frac{\sec^2(\pi y)}{1 - \tan^2(\pi y)}. \tag{B.26}$$

Now let $z = \tan(\pi y) \in (-1, 1)$, then since $\frac{\mathrm{d}z}{\mathrm{d}y} = \pi \sec^2(\pi y)$, the integral in (B.25) simplifies and can be calculated as follows

$$\int \frac{\sec^2(\pi y)}{1 - \tan^2(\pi y)}\, \mathrm{d}y = \frac{1}{\pi} \int \frac{\mathrm{d}z}{1 - z^2} = \frac{1}{\pi} \ln\left(\frac{1+z}{1-z}\right) = \frac{1}{\pi} \tanh^{-1}(z) = \frac{1}{\pi} \tanh^{-1}(\tan(\pi y)). \tag{B.27}$$

Substituting into (B.25) and rearranging, the general solution of ODE (B.24) is

$$y(t) = \frac{1}{\pi} \tan^{-1}\left[\tanh\left((-1)^{k_0} \pi \tau(t) + c\right)\right], \tag{B.28}$$

for c an arbitrary constant. Transforming back to the original variable ξ using (B.23) and applying the initial condition $\xi(0) = \xi_0$ yields the solution of the ODE in (B.21) as

$$\xi(t) = \frac{1}{\pi} \tan^{-1}\left\{\tanh\left[(-1)^{k_0} \pi [\tau(t) - \tau(0)] + \tanh^{-1}\left(\tan\left(\pi\left[\xi_0 - \tfrac{k_0}{2}\right]\right)\right)\right]\right\} + \frac{k_0}{2}. \tag{B.29}$$

Notice that despite the method for obtaining the solution not holding for $\xi_0 = \frac{1}{4} + \frac{k_0}{2}$, (B.29) does yield the correct solution of $\xi(t) = \xi_0$ for such ξ_0, and is thus a valid solution of the ODE for any $\xi_0 \in \mathbb{R}$.

From ODE (B.22), the solution of PDE (B.20) is constant along characteristics given by (B.29). Thus, the solution of (B.20) at some arbitrary point in space-time $(x, t) = (x_*, t_*)$ is simply equal to the initial condition of (B.20) at the $(t = 0)$ departure point of the characteristic passing through the point (x_*, t_*). The departure point is obtained by evaluating the solution of ODE (B.21) at $t = 0$ when it has been subject to the final-time condition $\xi(t_*) = x_*$. The solution of the final-value problem may be obtained from solution (B.29) for the initial-value problem by making the substitutions $\tau(0) \mapsto \tau(t_*)$, $k_0 \mapsto k_*$, and $\xi_0 \mapsto x_*$. Locating the departure point by setting $t = 0$ in the resulting equation and then substituting it into the initial condition of (B.20) gives the solution of the PDE at any point (x, t) in space-time as

$$u(x,t) = u_0 \left(\frac{1}{\pi} \tan^{-1} \left\{ \tanh \left[(-1)^\ell \pi [\tau(0) - \tau(t)] + \tanh^{-1} \left(\tan \left(\pi \left[x - \tfrac{\ell}{2} \right] \right) \right) \right] \right\} + \frac{\ell}{2} \right), \tag{B.30}$$

where

$$\ell = \left\lfloor 2x + \frac{1}{2} \right\rfloor. \tag{B.31}$$

B.5 Proof of Lemma 4.10: Constant-coefficient multilevel operator

Proof. Recall that the coefficients (4.91) are defined in terms of f_{p+1} applied to the quantities $\varepsilon^{(m^q \delta t)}$, $q \in \{0, \ldots, \ell\}$, which do not depended on the time index t_n for constant-coefficient problems. Therefore, we can immediately define a simplified version of (4.91) that applies to the constant-coefficient problem:

$$\nu_{p+1}^{(m^\ell \delta t)} = \begin{cases} \varphi_{p+1}^{(m^\ell \delta t)}, & \ell = 1, \\ m \nu_{p+1}^{(m^{\ell-1} \delta t)} + \varphi_{p+1}^{(m^\ell \delta t)}, & \ell > 1. \end{cases} \tag{B.32}$$

Furthermore, the coarse-grid operator (4.90) itself may be written in the simplified form,

$$\Phi^{(m^\ell \delta t)} = \left[I - \nu_{p+1}^{(m^\ell \delta t)} \mathcal{D}_{p+1} \right]^{-1} \mathcal{S}_p^{(m^\ell \delta t)}, \quad \ell \in \mathbb{N}. \tag{B.33}$$

From the shorthand introduced in (4.82) for $\boldsymbol{\varphi}_{p+1}^{(t_n,m^\ell\delta t)}$, we have that $\varphi_{p+1}^{(m^\ell\delta t)} = \varphi_{p+1}\big(\varepsilon^{(m^{\ell-1}\delta t)},\dots,\varepsilon^{(m^{\ell-1}\delta t)},\varepsilon^{(m^\ell\delta t)}\big)$. Further recalling from (4.71) that $\varphi_{p+1}(y_0,\dots,y_0,y_m) = f_{p+1}(y_m) - mf_{p+1}(y_0)$, the simplified coefficients (B.32) can be simplified even further as

$$\nu_{p+1}^{(m^\ell\delta t)} = \begin{cases} \Big[f_{p+1}\big(\varepsilon^{(m^\ell\delta t)}\big) - mf_{p+1}\big(\varepsilon^{(m^{\ell-1}\delta t)}\big)\Big], & \ell = 1, \\ m\nu_{p+1}^{(m^{\ell-1}\delta t)} + \Big[f_{p+1}\big(\varepsilon^{(m^\ell\delta t)}\big) - mf_{p+1}\big(\varepsilon^{(m^{\ell-1}\delta t)}\big)\Big], & \ell > 1. \end{cases} \tag{B.34}$$

Consider the coefficient (B.34) for some general $\ell \geq 4$, say, and let us drop the $p+1$ subscript and use the superscript ℓ rather than $m^\ell\delta t$ to improve readability, then,

$$\begin{aligned} \nu^{(\ell)} &= m\nu^{(\ell-1)} + \Big[f\big(\varepsilon^{(\ell)}\big) - mf\big(\varepsilon^{(\ell-1)}\big)\Big], && \text{(B.35)}\\ &= m^2\nu^{(\ell-2)} + m\Big[f\big(\varepsilon^{(\ell-1)}\big) - mf\big(\varepsilon^{(\ell-2)}\big)\Big] + \Big[f\big(\varepsilon^{(\ell)}\big) - mf\big(\varepsilon^{(\ell-1)}\big)\Big], && \text{(B.36)}\\ &= m^2\nu^{(\ell-2)} - m^2 f\big(\varepsilon^{(\ell-2)}\big) + f\big(\varepsilon^{(\ell)}\big), && \text{(B.37)}\\ &= m^3\nu^{(\ell-3)} + m^2\Big[f\big(\varepsilon^{(\ell-2)}\big) - mf\big(\varepsilon^{(\ell-3)}\big)\Big] - m^2 f\big(\varepsilon^{(\ell-2)}\big) + f\big(\varepsilon^{(\ell)}\big), && \text{(B.38)}\\ &= m^3\nu^{(\ell-3)} - m^3 f\big(\varepsilon^{(\ell-3)}\big) + f\big(\varepsilon^{(\ell)}\big). && \text{(B.39)} \end{aligned}$$

By an inductive argument, it is clear the telescoping nature of $\nu^{(\ell)}$ leads to it satisfying the following recurrence for general $\ell \in \mathbb{N}$

$$\nu^{(\ell)} = m^q\nu^{(\ell-q)} - m^q f\big(\varepsilon^{(\ell-q)}\big) + f\big(\varepsilon^{(\ell)}\big), \quad q \in \{0,\dots,\ell-1\}. \tag{B.40}$$

Evaluating this recurrence at $q = \ell - 1$ yields

$$\begin{aligned} \nu^{(\ell)} &= m^{\ell-1}\nu^{(1)} - m^{\ell-1} f\big(\varepsilon^{(1)}\big) + f\big(\varepsilon^{(\ell)}\big), && \text{(B.41)}\\ &= m^{\ell-1}\Big[f\big(\varepsilon^{(1)}\big) - mf\big(\varepsilon^{(0)}\big)\Big] - m^{\ell-1} f\big(\varepsilon^{(1)}\big) + f\big(\varepsilon^{(\ell)}\big), && \text{(B.42)}\\ &= f\big(\varepsilon^{(\ell)}\big) - m^\ell f\big(\varepsilon^{(0)}\big). && \text{(B.43)} \end{aligned}$$

Or, without the shorthand, we have

$$\nu_{p+1}^{(m^\ell\delta t)} = f_{p+1}\big(\varepsilon^{(m^\ell\delta t)}\big) - m^\ell f_{p+1}\big(\varepsilon^{(\delta t)}\big), \quad \ell \in \mathbb{N}. \tag{B.44}$$

Substituting this coefficient into the coarse-grid operator (B.33) gives the claimed result of (4.92). ■

B.6 Proof of Lemma 4.11: Polynomial interpolation error in two dimensions

Proof. The two-dimensional interpolating polynomial may be written as

$$v_p(x,y) = \sum_{q=-\ell(p)}^{r(p)} L_q(y) \sum_{s=-\ell(p)}^{r(p)} L_s(x) v(x_s; y_q) = \sum_{q=-\ell(p)}^{r(p)} L_q(y) \mathcal{X}_p(x; y_q), \tag{B.45}$$

in which L_q is the qth, degree p Lagrange basis polynomial. Now evaluate $v_p(x,y)$ at $(x_0 - h\varepsilon, y_0 - h\eta)$, and replace the interpolating polynomials in the x direction, i.e., $\mathcal{X}_p(x_0 - h\varepsilon; y_q)$, with their error estimate that was used, for example, in the proof of Lemma 4.1, to give

$$\begin{aligned}
&v_p(x_0 - h\varepsilon, y_0 - h\eta) \\
&= \sum_{q=-\ell(p)}^{r(p)} L_q(y_0 - h\eta) \mathcal{X}_p(x_0 - h\varepsilon; y_q), && \text{(B.46)}\\
&= \sum_{q=-\ell(p)}^{r(p)} L_q(y_0 - h\eta) \left(\left[1 - (-h)^{p+1} f_{p+1}(\varepsilon) \frac{\partial^{p+1}}{\partial x^{p+1}} \right] v(x_0 - h\varepsilon; y_q) + \mathcal{O}(h^{p+2}) \right), && \text{(B.47)}\\
&= \left[1 - (-h)^{p+1} f_{p+1}(\varepsilon) \frac{\partial^{p+1}}{\partial x^{p+1}} \right] \sum_{q=-\ell(p)}^{r(p)} L_q(y_0 - h\eta) v(x_0 - h\varepsilon; y_q) + \mathcal{O}(h^{p+2}), && \text{(B.48)}\\
&= \left[1 - (-h)^{p+1} f_{p+1}(\varepsilon) \frac{\partial^{p+1}}{\partial x^{p+1}} \right] \mathcal{Y}_p(y_0 - h\eta; x_0 - h\varepsilon) + \mathcal{O}(h^{p+2}). && \text{(B.49)}
\end{aligned}$$

Now replace the one-dimensional interpolating polynomials in the y-direction, that is, $\mathcal{Y}_p(y_0 - h\eta; x_0 - h\varepsilon)$, with the error estimate analogous to what was used in the x-direction to give

$$\begin{aligned}
&v_p(x_0 - h\varepsilon, y_0 - h\eta) = \left[1 - (-h)^{p+1} f_{p+1}(\varepsilon) \frac{\partial^{p+1}}{\partial x^{p+1}} \right] \\
&\left(\left[1 - (-h)^{p+1} f_{p+1}(\eta) \frac{\partial^{p+1}}{\partial y^{p+1}} \right] v(x_0 - h\varepsilon, y_0 - h\eta) + \mathcal{O}(h^{p+2}) \right) + \mathcal{O}(h^{p+2}), && \text{(B.50)}\\
&= \left(1 - (-h)^{p+1} \left[f_{p+1}(\varepsilon) \frac{\partial^{p+1}}{\partial y^{p+1}} + f_{p+1}(\eta) \frac{\partial^{p+1}}{\partial y^{p+1}} \right] \right) v(x_0 - h\varepsilon, y_0 - h\eta) + \mathcal{O}(h^{p+2}). && \text{(B.51)}
\end{aligned}$$

Rearranging (B.51) for $v(x_0 - h\varepsilon, y_0 - h\eta)$ gives the result (4.109). ■

Bibliography

[1] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, W. Pazner, M. Stowell, V. Tomov, J. Dahm, D. Medina, and S. Zampini, *MFEM: a modular finite element methods library*, Comput. Math. Appl., (2020). [Cited on pages 183 and 185.]

[2] G. Bal and Y. Maday, *A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put*, in Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2002, pp. 189–202. [Cited on page 15.]

[3] R. E. Bank, J. W. L. Wan, and Z. Qu, *Kernel preserving multigrid methods for convection-diffusion equations*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1150–1171. [Cited on pages 75, 82, and 87.]

[4] S. Basting and E. Bänsch, *Preconditioners for the Discontinuous Galerkin time-stepping method of arbitrary order*, ESAIM: Math. Model. Numer. Anal., 51 (2017), pp. 1173–1195. [Cited on pages 146, 153, 158, and 159.]

[5] M. Benzi, *Some uses of the field of values in numerical analysis*, Boll. Unione Mat. Ital., 14 (2021), pp. 159–177. [Cited on pages 147 and 148.]

[6] J.-P. Berrut and L. N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517. [Cited on page 118.]

[7] N. Besse and M. Mehrenberger, *Convergence of classes of high-order semi-Lagrangian schemes for the Vlasov–Poisson system*, Math. Comp., 77 (2008), pp. 93–123. [Cited on page 102.]

[8] T. A. Bickart, *An Efficient Solution Process for Implicit Runge–Kutta Methods*, SIAM J. Numer. Anal., 14 (1977), pp. 1022–1027. [Cited on pages 145, 146, and 166.]

[9] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–333. [Cited on pages 45 and 54.]

[10] ——, *Multigrid solvers for non-elliptic and singular-perturbation steady-state problems.* The Weizmann Institute of Science. Rehovot, Israel. (unpublished), 1981. [Cited on pages 45, 81, 82, 84, and 195.]

[11] A. Brandt and I. Yavneh, *On multigrid solution of high-reynolds incompressible entering flows*, J. Comput. Phys., 101 (1992), pp. 151–164. [Cited on page 86.]

[12] A. Brandt and I. Yavneh, *Accelerated multigrid convergence and high-reynolds recirculating flows*, SIAM J. Sci. Comput., 14 (1993), pp. 607–626. [Cited on pages 81 and 82.]

[13] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM: Society for Industrial and Applied Mathematics, 2000. [Cited on pages 7 and 45.]

[14] J. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, Ltd, 2003. [Cited on pages 18, 143, 197, and 198.]

[15] J. C. Butcher, *On the implementation of implicit Runge-Kutta methods*, BIT Numer. Math., 16 (1976), pp. 237–240. [Cited on pages 145, 146, and 166.]

[16] X. Cai, S. Boscarino, and J.-M. Qiu, *High order semi-lagrangian discontinuous galerkin method coupled with runge-kutta exponential integrators for nonlinear vlasov dynamics*, J. Comput. Phys., 427 (2021), p. 110036. [Cited on pages 4 and 112.]

[17] X. Cai, W. Guo, and J.-M. Qiu, *A high order conservative semi-Lagrangian discontinuous Galerkin method for two-dimensional transport simulations*, J. Sci. Comput., 73 (2017), pp. 514—542. [Cited on pages 4 and 112.]

[18] M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don, *The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: a study of the boundary error*, SIAM J. Sci. Comput., 16 (1995), pp. 1241–1252. [Cited on page 37.]

[19] F. Chen, J. S. Hesthaven, and X. Zhu, *On the use of reduced basis methods to accelerate and stabilize the parareal method*, in Reduced Order Methods for modeling and computational reduction, Springer, 2014, pp. 187–214. [Cited on pages 15 and 44.]

[20] J. Christopher, R. D. Falgout, J. B. Schroder, S. M. Guzik, and X. Gao, *A space-time parallel algorithm with adaptive mesh refinement for computational fluid dynamics*, Comput. Vis. Sci., 23 (2020). [Cited on page 45.]

[21] R. Courant, K. Friedrichs, and H. Lewy, *Über die partiellen differenzengleichungen der mathematischen physik*, Math. Ann., 100 (1928), pp. 32–74. [Cited on page 3.]

[22] ——, *On the partial difference equations of mathematical physics*, IBM J. Res. Dev, 11 (1967), pp. 215–234. [Cited on page 3.]

[23] X. Dai and Y. Maday, *Stable parareal in time method for first-and second-order hyperbolic systems*, SIAM J. Sci. Comput., 35 (2013), pp. A52–A78. [Cited on pages 15 and 44.]

[24] P. J. Davis, *Interpolation and approximation*, Dover Publications, INC., New York, 1975. [Cited on pages 94 and 115.]

[25] H. De Sterck, R. D. Falgout, S. Friedhoff, O. A. Krzysik, and S. P. MacLachlan, *Optimizing multigrid reduction-in-time and Parareal coarse-grid operators for linear advection*, Numer. Linear Algebra Appl., 28 (2021). [Cited on page iv.]

[26] H. De Sterck, S. Friedhoff, A. J. M. Howse, and S. P. MacLachlan, *Convergence analysis for parallel-in-time solution of hyperbolic systems*, Numer. Linear Algebra Appl., 27 (2020), p. e2271. [Cited on pages 10, 16, 44, 45, 46, 48, 53, 55, 56, 75, 78, and 79.]

[27] S. Demko, W. F. Moss, and P. W. Smith, *Decay rates for inverses of band matrices*, Math. Comp., 43 (1984), pp. 491–491. [Cited on page 109.]

[28] V. A. Dobrev, T. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM J. Sci. Comput., 39 (2017), pp. S501–S527. [Cited on pages 10, 15, 16, 18, 23, 24, 34, 37, 44, 46, 48, 53, 64, 76, 131, and 199.]

[29] D. R. Durran, *Numerical Methods for Fluid Dynamics*, Springer New York, second ed., 2010. [Cited on pages 3, 4, and 111.]

[30] M. Emmett and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Commun. Appl. Math. Comput. Sci., 7 (2012), pp. 105–132. [Cited on page 6.]

[31] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton Jacobi Equations*, CAMBRIDGE, Feb. 2014. [Cited on pages 3, 93, 102, 111, 112, and 134.]

[32] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 14 (2014), pp. 951–952. [Cited on pages 6, 7, 9, 10, 11, 15, 23, 34, 48, and 131.]

[33] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, J. B. Schroder, and S. Vandewalle, *Multigrid methods with space–time concurrency*, Comput. Vis. Sci., 18 (2017), pp. 123–143. [Cited on pages 6, 15, 42, and 131.]

[34] R. D. Falgout, M. Lecouvez, and C. S. Woodward, *A parallel-in-time algorithm for variable step multistep methods*, J. Comput. Sci., 37 (2019), p. 101029. [Cited on pages 11 and 15.]

[35] R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, *Multigrid reduction in time for nonlinear parabolic problems: A case study*, SIAM J. Sci. Comput., 39 (2017), pp. S298–S322. [Cited on pages 7, 15, 16, 131, and 195.]

[36] R. D. Falgout and U. M. Yang, *hypre: A library of high performance preconditioners*, European Conference on Parallel Processing, 2331 LNCS (2002), pp. 632–641. [Cited on pages 184, 185, and 189.]

[37] S. R. Franco, F. J. Gaspar, M. A. V. Pinto, and C. Rodrigo, *Multigrid method based on a space-time approach with standard coarsening for parabolic problems*, Appl. Math. Comput., 317 (2018), pp. 25–34. [Cited on page 79.]

[38] S. Friedhoff and S. MacLachlan, *A generalized predictive analysis tool for multigrid methods*, Numer. Linear Alg. Appl., 22 (2015), pp. 618–647. [Cited on pages 16, 44, 45, 46, 48, 75, and 78.]

[39] S. Friedhoff, S. MacLachlan, and C. Börgers, *Local fourier analysis of space-time relaxation and multigrid schemes*, SIAM J. Sci. Comput., 35 (2013), pp. S250–S276. [Cited on pages 45 and 75.]

[40] M. J. Gander, *Analysis of the parareal algorithm applied to hyperbolic problems using characteristics*, Soc. Esp. Mat. Apl., 42 (2008), pp. 21–35. [Cited on pages 15, 16, 31, and 44.]

[41] M. J. Gander, *50 years of time parallel time integration*, in Contrib. Math. Comput. Sci., Springer International Publishing, 2015, pp. 69–113. [Cited on page 5.]

[42] M. J. Gander and S. Güttel, *PARAEXP: A parallel integrator for linear initial-value problems*, SIAM J. Sci. Comput., 35 (2013), pp. C123–C142. [Cited on page 16.]

[43] M. J. Gander, Y.-L. Jiang, B. Song, and H. Zhang, *Analysis of two parareal algorithms for time-periodic problems*, SIAM J. Sci. Comput., 35 (2013), pp. A2393–A2415. [Cited on pages 45 and 72.]

[44] M. J. Gander, F. Kwok, and H. Zhang, *Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT*, Comput. Vis. Sci., 19 (2018), pp. 59–74. [Cited on pages 6, 16, 44, 48, and 49.]

[45] M. J. Gander and M. Neumuller, *Analysis of a new space-time parallel multigrid algorithm for parabolic problems*, SIAM J. Sci. Comput., 38 (2016), pp. A2173–A2208. [Cited on pages 6, 45, and 79.]

[46] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578. [Cited on pages 6, 15, 16, and 44.]

[47] M. Gasca and T. Sauer, *On the history of multivariate polynomial interpolation*, in Numerical Analysis: Historical Developments in the 20th Century, Elsevier, 2001, pp. 135–147. [Cited on page 135.]

[48] E. Hairer and G. Wanner, *Solving ordinary differential equations II: Stiff and Differential-Algebraic Problems*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1996. [Cited on pages 18, 143, 147, 185, and 198.]

[49] A. Hessenthaler, R. D. Falgout, J. B. Schroder, A. de Vecchi, D. Nordsletten, and O. Röhrle, *Time-periodic steady-state solution of fluid-structure interaction and cardiac flow problems through multigrid-reduction-in-time*. arXiv:2105.00305, 2021. [Cited on pages 45 and 72.]

[50] A. Hessenthaler, D. Nordsletten, O. Röhrle, J. B. Schroder, and R. D. Falgout, *Convergence of the multigrid reduction in time algorithm for the linear elasticity equations*, Numer. Linear Algebra Appl., 25 (2018), p. e2155. [Cited on pages 15, 44, and 128.]

[51] A. Hessenthaler, B. S. Southworth, D. Nordsletten, O. Röhrle, R. D. Falgout, and J. B. Schroder, *Multilevel convergence analysis of multigrid-reduction-in-time*, SIAM J. Sci. Comput., 42 (2020), pp. A771–A796. [Cited on pages 16, 18, 23, 24, 35, 37, 44, 46, 48, 49, 53, and 76.]

[52] J. S. Hesthaven, *Numerical methods for conservation laws: From analysis to algorithms*, SIAM, Philadelphia, PA, 2017. [Cited on pages 3, 4, 18, 37, and 197.]

[53] W. Hoffmann and J. J. B. D. Swart, *Approximating Runge-Kutta matrices by triangular matrices*, BIT Numer. Math., 37 (1997), pp. 346–354. [Cited on page 146.]

[54] G. Horton and S. Vandewalle, *A space-time multigrid method for parabolic partial differential equations*, SIAM J. Sci. Comput., 16 (1995), pp. 848–864. [Cited on page 6.]

[55] G. Horton, S. Vandewalle, and P. Worley, *An algorithm with polylog parallel complexity for solving parabolic partial differential equations*, SIAM J. Sci. Comput., 16 (1995), pp. 531–541. [Cited on page 6.]

[56] P. J. v. d. Houwen and J. J. B. d. Swart, *Triangularly Implicit Iteration Methods for ODE-IVP Solvers*, SIAM J. Sci. Comput., 18 (1997), pp. 41–55. [Cited on page 146.]

[57] A. Howse, *Nonlinear Preconditioning Methods for Optimization and Parallel-In-Time Methods for 1D Scalar Hyperbolic Partial Differential Equations*, PhD thesis, University of Waterloo, Waterloo, Canada, 2017. [Cited on pages 15, 44, and 195.]

[58] A. J. M. Howse, H. De Sterck, R. D. Falgout, S. MacLachlan, and J. Schroder, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid Burgers equations*, SIAM J. Sci. Comput., 41 (2019), pp. A538–A565. [Cited on pages 7, 15, 22, 36, 44, 48, 49, 113, 128, and 195.]

[59] C.-S. Huang, T. Arbogast, and C.-H. Hung, *A semi-Lagrangian finite difference WENO scheme for scalar nonlinear conservation laws*, J. Comput. Phys., 322 (2016), pp. 559–585. [Cited on page 112.]

[60] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228. [Cited on page 3.]

[61] C. Kennedy and M. H. Carpenter, *Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review.* NASA Technical Memorandum, TM-2016-219173, 2016. [Cited on page 143.]

[62] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge University Press, Cambridge, United Kingdom, 2004. [Cited on pages 3 and 4.]

[63] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, vol. 98, SIAM, 2007. [Cited on pages 3 and 4.]

[64] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'edp par un schéma en temps pararéel*, C. R. Acad. Sci-Series I-Mathematics, 332 (2001), pp. 661–668. [Cited on pages 6 and 15.]

[65] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212. [Cited on page 3.]

[66] C. F. Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100. [Cited on page 47.]

[67] K. A. Mardal, T. K. Nilssen, and G. A. Staff, *Order-Optimal Preconditioners for Implicit Runge–Kutta Schemes Applied to Parabolic PDEs*, SIAM J. Sci. Comput., 29 (2007), pp. 361–375. [Cited on page 146.]

[68] G. Mengaldo, A. Wyszogrodzki, M. Diamantakis, S.-J. Lock, F. X. Giraldo, and N. P. Wedi, *Current and emerging time-integration strategies in global numerical weather and climate prediction*, Arch. Comput. Methods Eng., 26 (2018), pp. 663–684. [Cited on page 4.]

[69] A. S. Nielsen, G. Brunner, and J. S. Hesthaven, *Communication-aware adaptive Parareal with application to a nonlinear hyperbolic system of partial differential equations*, J. Comput. Phys., 371 (2018), pp. 483–505. [Cited on pages 11, 15, 16, 22, and 44.]

[70] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Commun. ACM, 7 (1964), pp. 731–733. [Cited on page 5.]

[71] Y. Notay, *Rigorous convergence proof of space-time multigrid with coarsening in space*, Numer. Algorithms, (2021). [Cited on page 78.]

[72] B. W. Ong and J. B. Schroder, *Applications of time parallelization*, Comput. Vis. Sci., 23 (2020). [Cited on pages 5 and 15.]

[73] W. Pazner and P.-O. Persson, *Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations*, J. Comput. Phys., 335 (2017), pp. 700–717. [Cited on pages 144 and 146.]

[74] J.-M. Qiu and C.-W. Shu, *Conservative high order semi-lagrangian finite difference WENO methods for advection in incompressible flow*, J. Comput. Phys., 230 (2011), pp. 863–889. [Cited on pages 4 and 112.]

[75] M. M. Rana, V. E. Howle, K. Long, A. Meek, and W. Milestone, *A new block preconditioner for implicit runge–kutta methods for parabolic PDE problems*, SIAM J. Sci. Comput., (2021), pp. S475–S495. [Cited on page 146.]

[76] S. C. Reddy and L. N. Trefethen, *Stability of the method of lines*, Numer. Math., 62 (1992), pp. 235–267. [Cited on page 147.]

[77] M. Ries, U. Trottenberg, and G. Winter, *A note on MGR methods*, Linear Algebra Appl., 49 (1983), pp. 1–26. [Cited on page 75.]

[78] C. Rodrigo, F. J. Gaspar, C. W. Oosterlee, and I. Yavneh, *Accuracy measures and fourier analysis for the full multigrid algorithm*, SIAM J. Sci. Comput., 32 (2010), pp. 3108–3129. [Cited on page 57.]

[79] K. Rupp, *42 years of microprocessor trend data*, 2018. https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/. [Cited on page 5.]

[80] D. Ruprecht, *Wave propagation characteristics of Parareal*, Comput. Vis. Sci., 19 (2018), pp. 1–17. [Cited on pages 12, 15, 16, 44, and 97.]

[81] D. Ruprecht and R. Krause, *Explicit parallel-in-time integration of a linear acoustic-advection system*, Comput. & Fluids, 59 (2012), pp. 72–83. [Cited on pages 15 and 44.]

[82] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, second ed., 2003. [Cited on page 159.]

[83] A. Schmitt, M. Schreiber, P. Peixoto, and M. Schäfer, *A numerical study of a semi-Lagrangian Parareal method applied to the viscous Burgers equation*, Comput. Vis. Sci., 19 (2018), pp. 45–57. [Cited on pages 12, 15, 16, and 44.]

[84] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in Advanced numerical approximation of nonlinear hyperbolic equations, Springer, 1998, pp. 325–432. [Cited on pages 4 and 18.]

[85] A. A. Sivas, B. S. Southworth, and S. Rhebergen, *AIR algebraic multigrid for a space-time hybridizable discontinuous Galerkin discretization of advection (-diffusion)*, arXiv preprint arXiv:2010.11130, (2020). [Cited on page 6.]

[86] P. K. Smolarkiewicz and J. A. Pudykiewicz, *A class of semi-lagrangian approximations for fluids*, J. Atmospheric Sci., 49 (1992), pp. 2082–2096. [Cited on pages 112 and 119.]

[87] B. Song and Y.-L. Jiang, *Analysis of a new parareal algorithm based on waveform relaxation method for time-periodic problems*, Numer. Algorithms, 67 (2013), pp. 599–622. [Cited on page 45.]

[88] B. S. Southworth, *Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 564–608. [Cited on pages 16, 23, 24, 37, 44, 46, 48, 49, 53, 76, and 77.]

[89] B. S. Southworth, O. A. Krzysik, and W. Pazner, *IRKIntegration*. Software, 2020. https://github.com/bensworth/IRKIntegration. [Cited on pages 148, 150, 183, 184, 185, 190, and 193.]

[90] B. S. Southworth, O. A. Krzysik, and W. Pazner, *Fast solution of fully implicit Runge-Kutta and discontinuous Galerkin in time for numerical PDEs, Part II:*

nonlinearities and DAEs, SIAM J. Sci. Comput. (accepted), (2021). arXiv preprint arXiv:2101.01776. [Cited on pages iv, 147, 166, 183, and 191.]

[91] B. S. Southworth, O. A. Krzysik, W. Pazner, and H. De Sterck, *Fast solution of fully implicit Runge-Kutta and discontinuous Galerkin in time for numerical PDEs, Part I: the linear setting*, SIAM J. Sci. Comput. (accepted), (2021). arXiv preprint arXiv:2101.00512. [Cited on pages iv, 147, 149, 183, and 188.]

[92] B. S. Southworth, W. Mitchell, A. Hessenthaler, and F. Danieli, *Tight two-level convergence of Linear Parareal and MGRIT: Extensions and implications in practice.* arXiv:2010.11879, 2020. [Cited on pages 76 and 77.]

[93] B. S. Southworth, A. A. Sivas, and S. Rhebergen, *On fixed-point, Krylov, and 2×2 block preconditioners for nonsymmetric problems*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 871–900. [Cited on page 168.]

[94] G. A. Staff, K.-A. Mardal, and T. K. Nilssen, *Preconditioning of fully implicit Runge-Kutta schemes for parabolic PDEs*, Model. Identif. Control., 27 (2006), pp. 109–123. [Cited on pages 146 and 153.]

[95] A. Staniforth and J. Côté, *Semi-lagrangian integration schemes for atmospheric models—a review*, Mon. Wea. Rev., 119 (1991), pp. 2206–2223. [Cited on page 4.]

[96] O. Steinbach and H. Yang, *Comparison of algebraic multigrid methods for an adaptive space-time finite-element discretization of the heat equation in 3d and 4d*, Numer. Linear Algebra Appl., 25 (2018), p. e2143. [Cited on page 6.]

[97] J. Steiner, D. Ruprecht, R. Speck, and R. Krause, *Convergence of Parareal for the Navier-Stokes equations depending on the Reynolds number*, in Numerical Mathematics and Advanced Applications-ENUMATH 2013, Springer, 2015, pp. 195–202. [Cited on pages 15 and 44.]

[98] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1982, pp. 1–176. [Cited on pages 45 and 75.]

[99] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic press, 2001. [Cited on pages 7, 37, 45, 48, 55, 59, 61, 73, 75, 82, 84, and 86.]

[100] S. Vandewalle and G. Horton, *Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods*, Computing, 54 (1995), pp. 317–330. [Cited on pages 45, 73, 75, and 79.]

[101] S. Vandewalle and R. Piessens, *Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1330–1346. [Cited on page 6.]

[102] S. G. Vandewalle and E. F. V. D. Veldet, *Space-time concurrent multigrid waveform relaxation*, Ann. Numer. Math., 1 (1994), pp. 347–360. [Cited on page 6.]

[103] W. L. Wan and T. F. Chan, *A phase error analysis of multigrid methods for hyperbolic equations*, SIAM J. Sci. Comput., 25 (2003), pp. 857–880. [Cited on pages 75 and 82.]

[104] R. Wienands and C. W. Oosterlee, *On three-grid fourier analysis for multigrid*, SIAM J. Sci. Comput., 23 (2001), pp. 651–671. [Cited on page 195.]

[105] R. R. Wienands and W. W. Joppich, *Practical Fourier analysis for multigrid methods*, Chapman & Hall/CRC, Boca Raton, FL, 2005. [Cited on pages 45 and 55.]

[106] D. L. Williamson, *The evolution of dynamical cores for global atmospheric models*, J. Meteorol. Soc. Jpn., 85B (2007), pp. 241–269. [Cited on page 4.]

[107] *XBraid: Parallel multigrid in time.* `http://llnl.gov/casc/xbraid`. [Cited on pages 41 and 196.]

[108] I. Yavneh, *Coarse-grid correction for nonelliptic and singular perturbation problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1682–1699. [Cited on pages 26, 33, 45, 75, 81, 82, 84, 86, 87, 89, 98, 128, 195, and 202.]

[109] I. Yavneh, C. H. Venner, and A. Brandt, *Fast multigrid solution of the advection problem with closed characteristics*, SIAM J. Sci. Comput., 19 (1998), pp. 111–125. [Cited on pages 86, 128, and 195.]