Multimethods for the Efficient Solution of Multiscale Differential Equations

Steven B. Roberts

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Adrian Sandu, Chair
Calvin J. Ribbens
Carol S. Woodward
Timothy C. Warburton
Young Cao

August 11, 2021
Blacksburg, Virginia

Keywords: Time integration, Multirate, Implicit-Explicit, Runge–Kutta, General Linear Method

Multimethods for the Efficient Solution of Multiscale Differential Equations

Steven B. Roberts

(ABSTRACT)

Mathematical models involving ordinary differential equations (ODEs) play a critical role in scientific and engineering applications. Advances in computing hardware and numerical methods have allowed these models to become larger and more sophisticated. Increasingly, problems can be described as multiphysics and multiscale as they combine several different physical processes with different characteristics. If just one part of an ODE is stiff, nonlinear, chaotic, or rapidly-evolving, this can force an expensive method or a small timestep to be used. A method which applies a discretization and timestep uniformly across a multiphysics problem poorly utilizes computational resources and can be prohibitively expensive.

The focus of this dissertation is on "multimethods" which apply different methods to different partitions of an ODE. Well-designed multimethods can drastically reduce the computation costs by matching methods to the individual characteristics of each partition while making minimal concessions to stability and accuracy. However, they are not without their limitations. High order methods are difficult to derive and may suffer from order reduction. Also, the stability of multimethods is difficult to characterize and analyze.

The goals of this work are to develop new, practical multimethods and to address these issues. First, new implicit multirate Runge–Kutta methods are analyzed with a special focus on stability. This is extended into implicit multirate infinitesimal methods. We introduce approaches for constructing implicit-explicit methods based on Runge–Kutta and general linear methods. Finally, some unique applications of multimethods are considered including using surrogate models to accelerate Runge–Kutta methods and eliminating order reduction on linear ODEs with time-dependent forcing.

Multimethods for the Efficient Solution of Multiscale Differential Equations

Steven B. Roberts

(GENERAL AUDIENCE ABSTRACT)

Almost all time-dependent physical phenomena can be effectively described via ordinary differential equations. This includes chemical reactions, the motion of a pendulum, the propagation of an electric signal through a circuit, and fluid dynamics. In general, it is not possible to find closed-form solutions to differential equations. Instead, time integration methods can be employed to numerically approximate the solution through an iterative procedure. Time integration methods are of great practical interest to scientific and engineering applications because computational modeling is often much cheaper and more flexible than constructing physical models for testing.

Large-scale, complex systems frequently combine several coupled processes with vastly different characteristics. Consider a car where the tires spin at several hundred revolutions per minute, while the suspension has oscillatory dynamics that is orders of magnitude slower. The brake pads undergo periods of slow cooling, then sudden, rapid heating. When using a time integration scheme for such a simulation, the fastest dynamics require an expensive and small timestep that is applied globally across all aspects of the simulation. In turn, an unnecessarily large amount of work is done to resolve the slow dynamics.

The goal of this dissertation is to explore new "multimethods" for solving differential equations where a single time integration method using a single, global timestep is inadequate. Multimethods combine together existing time integration schemes in a way that is better tailored to the properties of the problem while maintaining desirable accuracy and stability properties. This work seeks to overcome limitations on current multimethods, further the understanding of their stability, present new applications, and most importantly, develop methods with improved efficiency.

Dedication

To my parents.

Acknowledgments

This dissertation would not be possible without the support of countless people. First, I would like to thank my advisor, Dr. Adrian Sandu, for inviting me to join the Computational Science Laboratory and for the time we have worked together. I am deeply grateful for his guidance and vested interest in my success. Also from the Computational Science Laboratory, I would like to thank labmates Ross Glandon, Mahesh Narayanamurthi, and Paul Tranquilli for their mentorship. Special thanks goes to Arash Sarshar for his constant support and all of the opportunities we had to work together. Andrey Popov has provided many contributions to software projects, and I am thankful for his help.

I am greatly appreciative of the collaboration and summer internship I had with Dr. John Loffeld and Dr. Carol Woodward of Lawrence Livermore National Laboratory.

I acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: https://arc.vt.edu/

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Mathematical models are a powerful tool for making scientific predictions, furthering our understanding of physical phenomena, developing new technologies, and making critical engineering decisions. These models are frequently described by a system of ordinary differential equations (ODEs)

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t \in [t_0, t_f], \tag{1.1}$$

where $y(t) \in \mathbb{C}^d$. ODEs arise in a multitude of domains including chemistry, atmospheric science, economics, astrophysics, and epidemiology.

Time integration methods, which numerically approximate the solution to (1.1), come in many forms. One of the primary ways to characterize a method is as explicit or implicit. Explicit methods produce a solution at the next timestep only using information from previous timesteps, while implicit methods depend on future, unknown states of the system. Implicit methods must use a nonlinear solver like Newton's method to solve for the implicitly-defined quantities. This requires the Jacobian matrix $\frac{\partial f}{\partial y}$ which is difficult to compute and store for large problems. Linear solves involving the Jacobian usually dominate the cost of implicit integrators. This presents a classic and well-known trade-off. On one hand, explicit methods are cheap, but their stability puts strong limitations on the stepsize. On the other hand, implicit methods are expensive but can have excellent stability properties.

Runge–Kutta methods and linear multistep methods are some of the most popular methods used to solve (1.1). Again, this presents a dichotomy in that Runge–Kutta methods are one-step and multi-stage, while linear multistep methods are multistep and single-stage. Similar to the implicit versus explicit trade-off, linear multistep methods tend to be relatively inexpensive but with stability limitations, while Runge–Kutta methods are more expensive but have improved stability. General linear methods (GLMs) are a class of multi-stage, multistep methods designed to achieve the best of both worlds. As the name suggests, it encompasses Runge–Kutta and linear multistep methods, while also providing the foundation for other schemes such as one-leg, Peer, and two-step Runge–Kutta methods.

These are just a few integration frameworks out of countless more, each offering hundreds if not thousands of practical methods. There are so many options because there are so many special ODE properties that must be considered: stiffness, nonlinearity, availability

and sparsity of the Jacobian, conservation properties, positivity, and asymptotic behavior. For the best performance and accuracy, an integrator must have properties tailored to the ODE it is solving and the computing system on which it is solved. These method properties include monotonicity, implicitness, storage requirements, B-convergence, and parallelism. For a given method, there is typically a narrow range of problems for which it is optimal.

As the availability, speed, and parallelism of computational resources have continued to increase, simulations have scaled up in size and complexity. Instead of modeling a single physical process, modern simulations often combine several processes that are coupled together in complex, nonlinear ways. Each partition of the problem can have vastly different properties. Consider a coupled ocean-atmosphere model where the ocean temperature and salinity evolve significantly slower than the temperature, wind speed, and stiff chemical reactions of the atmosphere. Advection-diffusion-reaction partial differential equations (PDEs) are another example where advection is generally nonstiff, diffusion is stiff, and reactions introduce nonlinearity. This presents a significant challenge for the aforementioned time integration methods. While there may be optimal integrators for particular partitions, it is rare to have a single method that is satisfactory for the entire problem. It only takes a single partition to impose an expensive method or to impose a small timestep globally across all partitions. Global implicitness is a major concern because solving coupled, nonlinear equations involving the entire system can be prohibitively expensive.

This motivates the research of "multimethods." Instead of treating the right-hand side of (1.1) as a black box, multimethods solve ODEs of the form

$$y'(t) = f^{\{1\}}(t, y(t)) + f^{\{2\}}(t, y(t)) + \cdots + f^{\{N\}}(t, y(t)). \tag{1.2}$$

As the name suggests, multimethods treat each partition $f^{\{i\}}(t, y(t))$ with a different method or uses special coupling structures among partitions to provide a more targeted and efficient solution. Implicit-explicit (IMEX) methods [13, 14, 164] are examples where stiff terms of (1.2) are treated implicitly, and nonstiff terms are treated explicitly. Alternating direction implicit (ADI) methods [49, 62, 107, 136] typically treat all partitions with the same implicit method but are decoupled so that nonlinear solves are performed within a single partition at a time. Multirate methods [61, 118, 135] use different timesteps and possibly different methods for each partition.

1.2 Dissertation Objectives and Overview

Despite offering wonderful flexibility, multimethods present several challenges that have limited their performance and applicability. The focus of this work is to develop new, practical multimethods that address the following issues.

1. Many multimethods are low order. At orders one and two, simple linear interpolation and extrapolation of solutions provide sufficiently accurate coupling information. At

higher orders, more sophisticated error analysis is required. Historically, this analysis has been done on a method-by-method basis leading to complex and fragmented theoretical results.

2. There are multimethods of orders three and higher available in the literature, but when applied stiff problems or differential algebraic equations (DAEs), many are susceptible to the order reduction phenomenon.

3. The linear stability analysis of multimethods is difficult to characterize and is not well understood. Stability depends heavily on the choice of the linear test problem, its dimension, and many other factors. The coupling of a multimethod must be chosen judiciously to avoid significant degradation of the stability.

4. Not all problems have the right properties for multimethods to be effective. Examples include problems that cannot be partitioned easily or have tight, nonlinear coupling among partitions. Even when a problem has the correct properties, multimethods introduce additional storage, nonuniform data access, and more complex implementations, and these overheads can detract from speedups. Can multimethods be applied more broadly and in nontraditional settings?

Issues 1 and 3 are first addressed in chapter 2. It presents an investigation into implicit multirate methods based on the generalized-structure additively partitioned Runge–Kutta (GARK) framework. With implicit methods used across all partitions, methods must find a balance between stability and the cost of solving nonlinear equations for the stages. In order to characterize this important trade-off, we explore multirate coupling strategies, problems for assessing linear stability, and techniques to efficiently implement Newton iterations for stage equations. New implicit multirate methods up to order four are derived, and their accuracy and efficiency properties are verified with numerical tests.

Chapter 3 also considers implicit multirate Runge–Kutta methods and relates to issues 1 and 3. The methods of this chapter are based on the multirate infinitesimal general-structure additive Runge-Kutta (MRI-GARK) framework, though. The slow components of (1.2) are discretized by a Runge-Kutta method, and the fast components are resolved by solving modified fast differential equations. Two MRI-GARK extensions are proposed that introduce coupled implicit stages. This considerably improves the overall stability of the scheme at the price of requiring implicit stage calculations over the entire system. Again, methods as high as order four are derived.

GLMs offer an attractive foundation to build IMEX methods, but deriving high order schemes with acceptable stability properties is quite challenging. In chapter 4, we develop two systematic approaches for the construction IMEX GLMs with stages that can be computed in parallel. The primary novelty in this research is the parallel ensemble IMEX Euler family of methods. It is a generalization of IMEX Euler to arbitrarily high orders while

maintaining the same linear stability region and roughly the same runtime in a parallel setting. This address issues 1 and 3 but also issue 2 because IMEX GLMs are capable of solving stiff problems and DAEs.

For large-scale ODEs, surrogate models such as machine learning models or reduced-order models offer another way to reduce the computational cost but introduce an additional source of approximation error. In order to overcome the expense of a full model on one hand and the limitations of a surrogate models on the other, chapter 5 proposes a new accelerated time-stepping strategy that combines information from both. This approach is based on the MRI-GARK framework and relates to issue 4 above. The inexpensive surrogate model is integrated with a small timestep to guide the solution trajectory, and the full model is treated with a large timestep to occasionally correct for the surrogate model error and ensure convergence.

On stiff, linear ODEs with time-dependent forcing, Runge–Kutta methods can exhibit convergence rates lower than predicted by classical order condition theory. In chapter 6, we demonstrate how GARK-based multimethods can eliminate this order reduction, and thus, address issues 2 and 4. Instead of resorting to an expensive, fully implicit Runge-Kutta method for the entire problem, we propose a more flexible approach. An arbitrary Runge–Kutta method can be augmented with a fully implicit method to treat the forcing in such a way that it maintains the classical order in the presence of stiffness.

Finally, Chapter 7 investigates IMEX methods based on the GARK framework. Historically, IMEX methods have been built almost exclusively in the additive Runge–Kutta (ARK) and additive semi-implicit Runge–Kutta (ASIRK) frameworks. GARK allows us to combine implicit and explicit methods with a different number of stages, offers more control over the coupling, and provides more flexibility with simplifying assumptions. We use this to improve upon existing, highly-optimized IMEX methods in the literature. Order conditions for index-1 DAEs are considered as well to target issue 2.

Chapter 2

Implicit Multirate GARK Methods

Material from: Steven Roberts, John Loffeld, Arash Sarshar, Carol S. Woodward, and Adrian Sandu. Implicit multirate GARK methods. *Journal of Scientific Computing*, 87(1):4, 2021. doi:10.1007/s10915-020-01400-z

2.1 Introduction

In many real-world dynamical systems, there are parts of the system that evolve at significantly faster rates than other parts of the system. Time integration methods in which a single timestep is applied to all parts of the system can be inefficient and unsatisfactory for these multiscale problems. The fastest dynamics impose a relatively small, global timestep to ensure stability, to meet accuracy requirements, and in the case of an implicit method, to ensure convergence of the nonlinear solves for stage equations. This forces the slowest dynamics to be evaluated more frequently than necessary, leading to a costly integration. Instead of treating such a system as a black box, many numerical methods consider the fast and slow processes independently:

$$y' = f(y) = f^{\{\mathfrak{f}\}}(y) + f^{\{\mathfrak{s}\}}(y), \quad y(t_0) = y_0, \quad y(t) \in \mathbb{R}^d. \tag{2.1}$$

An important special case of this additively partitioned system is the component partitioned problem

$$\begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}' = \begin{bmatrix} f^{\{\mathfrak{f}\}}\left(y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right) \\ f^{\{\mathfrak{s}\}}\left(y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right) \end{bmatrix}, \tag{2.2}$$

with $y^{\{\mathfrak{f}\}} \in \mathbb{R}^{d^{\{\mathfrak{f}\}}}$, $y^{\{\mathfrak{s}\}} \in \mathbb{R}^{d^{\{\mathfrak{s}\}}}$, and $d = d^{\{\mathfrak{f}\}} + d^{\{\mathfrak{s}\}}$.

Multirate methods efficiently solve the system of ordinary differential equations (ODEs) given in (2.1) by integrating the fast dynamics $f^{\{\mathfrak{f}\}}$ with a smaller timestep than the slow dynamics $f^{\{\mathfrak{s}\}}$. The choice of how to partition f into $f^{\{\mathfrak{f}\}}$ and $f^{\{\mathfrak{s}\}}$ can depend on many factors including stiffness, accuracy requirements, evaluation cost, linearity, and memory requirements. In the case where an implicit method is needed, which will be the focus of this paper, the cost and convergence of the nonlinear solver also comes into consideration. There may be a small number of components of an ODE that causes slow convergence of Newton's method (e.g., a boundary layer). Such components can be grouped into $f^{\{\mathfrak{f}\}}$. In some cases, the Jacobian of f is an unstructured matrix leading to expensive linear solves,

but the problem can be decomposed such that linear solves with the Jacobians of $f^{\{\mathtt{f}\}}$ and $f^{\{\mathtt{s}\}}$ are inexpensive. Alternating directions implicit (ADI) methods [107] and approximate matrix factorization (AMF) methods [18], for example, exploit this property.

Implicit methods require excellent stability to offset the cost of solving potentially nonlinear equations in each step. For this reason, an understanding of the stability of multirate methods is crucial. One of the first works studying multirate stability was that of Gear [60]. Subsequent authors have examined multirate stability in the context of backward Euler [128, 148, 158], Runge–Kutta methods [10, 80, 93], linear multistep methods [61, 148, 159], and Rosenbrock methods [64, 125, 137].

Much of the development and implementation of multirate schemes for stiff systems has focused on multirate Rosenbrock methods [16, 64, 68, 138], but methods based on implicit Runge–Kutta methods have been explored as well. In [80], a multirate θ-method is presented and analyzed. Recently, multirate methods based on TR-BDF2 were proposed in [21, 46]. In [121, 129], new strategies for creating implicit multirate infinitesimal methods were introduced.

In [132], Sandu and Günther propose the generalized-structure additively partitioned Runge–Kutta (GARK) family of methods. GARK provides a unifying framework that includes traditional, implicit-explicit (IMEX), and multirate Runge–Kutta methods. Order conditions as well as the linear and nonlinear stability analysis are developed for this large class of methods. Günther and Sandu continue in [66] where many variants of multirate Runge–Kutta methods are cast as GARK methods. Multirate GARK (MrGARK) methods up to order four are derived in [135]. These include methods that are explicit in both partitions and methods that combine explicit and implicit methods.

In this work, we develop new MrGARK methods that are implicit in both the fast and slow partitions. The development is guided by new theoretical results regarding the stability of multirate methods. Necessary and sufficient conditions for achieving A-stability are presented, as well as some fundamental stability limitations on certain types of multirate methods. Many of these results extend past multirate methods to the entire GARK framework. Numerical experiments verify the order of convergence and the efficiency of the new schemes.

The structure of this paper is as follows. Section 2.2 introduces multirate methods using the GARK framework. The linear stability of multirate methods is explored in section 2.3. Section 2.4 discusses techniques to efficiently implement the Newton iterations. Section 2.5 contains the newly derived implicit MrGARK methods, and section 2.6 presents the numerical experiments used to test the methods. Finally, we summarize the results of the paper in section 2.7.

2.2 Multirate GARK Methods

The GARK framework [132] is used as the foundation for representing and analyzing multirate Runge–Kutta methods. In the most general form for a two-partitioned system (2.1), one step reads

$$Y_i^{\{\mathfrak{f}\}} = y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}} f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f}\}}\right) + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\left(Y_j^{\{\mathfrak{s}\}}\right), \quad \text{for } i = 1, \ldots, s^{\{\mathfrak{f}\}}, \tag{2.3a}$$

$$Y_i^{\{\mathfrak{s}\}} = y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s},\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\left(Y_j^{\{\mathfrak{s}\}}\right) + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{s},\mathfrak{f}\}} f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f}\}}\right), \quad \text{for } i = 1, \ldots, s^{\{\mathfrak{s}\}}, \tag{2.3b}$$

$$y_{n+1} = y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f}\}}\right) + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} b_j^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\left(Y_j^{\{\mathfrak{s}\}}\right). \tag{2.3c}$$

The coefficients of these methods can be organized into the following Butcher tableau:

$$\begin{array}{c|c} \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ \hline \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline \mathbf{b}^{\{\mathfrak{f}\}T} & \mathbf{b}^{\{\mathfrak{s}\}T} \end{array}. \tag{2.4}$$

The fast method $\left(\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}, \mathbf{b}^{\{\mathfrak{f}\}}, \mathbf{c}^{\{\mathfrak{f}\}}\right)$ has $\mathbf{s}^{\{\mathfrak{f}\}}$ stages, and the slow method $\left(\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}, \mathbf{b}^{\{\mathfrak{s}\}}, \mathbf{c}^{\{\mathfrak{s}\}}\right)$, has $\mathbf{s}^{\{\mathfrak{s}\}}$ stages. Also, we use the notation

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^{\{\mathfrak{f}\}} \\ \mathbf{b}^{\{\mathfrak{s}\}} \end{bmatrix}, \quad \mathbf{s} = \mathbf{s}^{\{\mathfrak{f}\}} + \mathbf{s}^{\{\mathfrak{s}\}}.$$

A common simplifying assumption, which ensures the fast and slow functions in (2.3) are computed at consistent times, is internal consistency:

$$\mathbf{c}^{\{\mathfrak{f}\}} := \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} \mathbb{1}_{s^{\{\mathfrak{f}\}}} = \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \mathbb{1}_{s^{\{\mathfrak{s}\}}} \quad \text{and} \quad \mathbf{c}^{\{\mathfrak{s}\}} := \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \mathbb{1}_{s^{\{\mathfrak{s}\}}} = \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} \mathbb{1}_{s^{\{\mathfrak{f}\}}}. \tag{2.5}$$

In [66], it was shown how several types of multirate Runge–Kutta methods can be described as GARK methods. In one step of a multirate method, the slow dynamics $f^{\{\mathfrak{s}\}}$ are integrated with a macro-step of H, and the fast dynamics $f^{\{\mathfrak{f}\}}$ are integrated with a micro-step of $h = H/M$. The multirate ratio M is a positive integer. Information between the two partitions is shared via the coupling matrices $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$ and $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$. In this section, we present two families of multirate Runge–Kutta methods viewed as special cases of the GARK framework.

2.2.1 Standard MrGARK

A standard MrGARK method is built on an $s^{\{\mathfrak{f}\}}$-stage fast base method $\left(A^{\{\mathfrak{f},\mathfrak{f}\}}, b^{\{\mathfrak{f}\}}, c^{\{\mathfrak{f}\}}\right)$ and an $s^{\{\mathfrak{s}\}}$-stage slow base method $\left(A^{\{\mathfrak{s},\mathfrak{s}\}}, b^{\{\mathfrak{s}\}}, c^{\{\mathfrak{s}\}}\right)$. From [66], one step proceeds as

$$\begin{aligned} Y_i^{\{\mathfrak{s}\}} &= y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s},\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\left(Y_j^{\{\mathfrak{s}\}}\right) + h \sum_{\lambda=1}^{M} \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{s},\mathfrak{f},\lambda\}} f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f},\lambda\}}\right), \\ &\text{for } i = 1, \dots, s^{\{\mathfrak{s}\}}, \end{aligned} \tag{2.6a}$$

$$\begin{aligned} Y_i^{\{\mathfrak{f},\lambda\}} &= \widetilde{y}_{n+(\lambda-1)/M} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} f^{\{\mathfrak{s}\}}\left(Y_j^{\{\mathfrak{s}\}}\right) + h \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}} f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f},\lambda\}}\right), \\ &\text{for } i = 1, \dots, s^{\{\mathfrak{f}\}}, \\ \widetilde{y}_{n+\lambda/M} &= \widetilde{y}_{n+(\lambda-1)/M} + h \sum_{i=1}^{s^{\{\mathfrak{f}\}}} b_i^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}}\left(Y_i^{\{\mathfrak{f},\lambda\}}\right), \\ &\text{for } \lambda = 1, \dots, M, \end{aligned} \tag{2.6b}$$

$$y_{n+1} = \widetilde{y}_{n+M/M} + H \sum_{i=1}^{s^{\{\mathfrak{s}\}}} b_i^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\left(Y_i^{\{\mathfrak{s}\}}\right), \tag{2.6c}$$

where the micro-steps start with $\widetilde{y}_n = y_n$. The corresponding Butcher tableau for (2.6) is

$$\begin{array}{c|c} \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ \hline \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline \mathbf{b}^{\{\mathfrak{f}\}T} & \mathbf{b}^{\{\mathfrak{s}\}T} \end{array} := \begin{array}{ccc|c} \frac{1}{M}A^{\{\mathfrak{f},\mathfrak{f}\}} & \cdots & 0 & A^{\{\mathfrak{f},\mathfrak{s},1\}} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{M}\mathbb{1}_{s^{\{\mathfrak{f}\}}} b^{\{\mathfrak{f}\}T} & \cdots & \frac{1}{M}A^{\{\mathfrak{f},\mathfrak{f}\}} & A^{\{\mathfrak{f},\mathfrak{s},M\}} \\ \hline \frac{1}{M}A^{\{\mathfrak{s},\mathfrak{f},1\}} & \cdots & \frac{1}{M}A^{\{\mathfrak{s},\mathfrak{f},M\}} & A^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline \frac{1}{M}b^{\{\mathfrak{f}\}T} & \cdots & \frac{1}{M}b^{\{\mathfrak{f}\}T} & b^{\{\mathfrak{s}\}T} \end{array}. \tag{2.7}$$

Note that $\mathbf{s}^{\{\mathfrak{f}\}} = Ms^{\{\mathfrak{f}\}}$ and $\mathbf{s}^{\{\mathfrak{s}\}} = s^{\{\mathfrak{s}\}}$.

If the fast and slow base methods are identical, the method is called *telescopic* as it can be applied in a nested fashion to more than two partitions [66]. Further, MrGARK methods can be classified as coupled or decoupled [135]. Decoupled methods only have implicitness in the base methods; the stages used in coupling can always be computed before they are needed. Coupled methods, on the other hand, have fast and slow stages that are implicitly defined in terms of each other and that must be computed together. Decoupled methods can be implemented more efficiently, but can sacrifice stability as we will see in section 2.3.

Order conditions for this family of methods comes from applying the particular multirate structure of (2.7) into the GARK order conditions. The conditions up to order four are provided in [135]. A similar approach has been used in [121, 129] to derive order conditions for MRI-GARK methods and in [132, 146] for multirate infinitesimal step methods [92, 141, 161].

2.2.2 Compound-Fast MrGARK

Another multirate strategy, based on the early work of Rice [118] and the later developments in [139, 158], is the compound-fast approach. The idea is to first take a macro-step of the full system (2.1) called the compound step. Over the large timestep, the fast integration is inaccurate and discarded. The fast partition is then reintegrated using a smaller timestep. Slow coupling information is required at the intermediate micro-steps and can come from an interpolant of the compound step solution. Note the fast partition is integrated twice for each timestep, but no extrapolation is required for the coupling. Moreover, an error estimate from the compound step, say from an embedded method, can be used to dynamically determine at each step which variables exceed accuracy tolerances and should form the fast components [139].

Traditionally, compound-fast methods have been posed for component partitioned systems (2.2), however, they easily extend to additively partitioned systems (2.1). One step of a compound-fast MrGARK scheme is given by

$$Y_i = y_n + H \sum_{j=1}^{s} a_{i,j}\, f(Y_j), \qquad i = 1, \ldots, s, \tag{2.8a}$$

$$\begin{aligned} Y_i^{\{\mathfrak{f},\lambda\}} &= \widetilde{y}_{n+(\lambda-1)/M} + h \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}}\, f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f},\lambda\}}\right) + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, f^{\{\mathfrak{s}\}}(Y_j), \\ &\quad \text{for } i = 1, \ldots, s^{\{\mathfrak{f}\}}, \\ \widetilde{y}_{n+\lambda/M} &= \widetilde{y}_{n+(\lambda-1)/M} + h \sum_{i=1}^{s^{\{\mathfrak{f}\}}} b_i^{\{\mathfrak{f}\}}\, f^{\{\mathfrak{f}\}}\left(Y_i^{\{\mathfrak{f},\lambda\}}\right), \\ &\quad \text{for } \lambda = 1, \ldots, M, \end{aligned} \tag{2.8b}$$

$$y_{n+1} = \widetilde{y}_{n+M/M} + H \sum_{i=1}^{s^{\{\mathfrak{s}\}}} b_i^{\{\mathfrak{s}\}}\, f^{\{\mathfrak{s}\}}(Y_i), \tag{2.8c}$$

where the micro-steps start with $\widetilde{y}_n = y_n$. The corresponding tableau is

$$\begin{array}{c|c} \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ \hline \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline \mathbf{b}^{\{\mathfrak{f}\}T} & \mathbf{b}^{\{\mathfrak{s}\}T} \end{array} := \begin{array}{cccc|c} A & 0 & \cdots & 0 & A \\ 0 & \frac{1}{M}A & \cdots & 0 & A^{\{\mathfrak{f},\mathfrak{s},1\}} \\ 0 & \vdots & \ddots & \vdots & \vdots \\ 0 & \frac{1}{M}\mathbb{1}_s b^T & \cdots & \frac{1}{M}A & A^{\{\mathfrak{f},\mathfrak{s},M\}} \\ \hline A & 0 & \cdots & 0 & A \\ \hline 0 & \frac{1}{M}b^T & \cdots & \frac{1}{M}b^T & b^T \end{array}.$$

This family of methods is telescopic, coupled in the macro-step (2.8a), and decoupled in the remaining fast micro-steps (2.8b). The coupling matrix $A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}$ can be interpreted as the

interpolation weights for the slow tendencies. These multirate methods will preserve the order of the base method if the interpolant is sufficiently accurate. We note that this is a sufficient condition but not always necessary. The GARK order conditions can be used to derive precise conditions to achieve a particular order.

Theorem 2.1 (Compound-fast MrGARK order conditions)**.** *An internally consistent compound-fast MrGARK method has order four if and only if the base method* (A, b, c) *has order four and the following coupling conditions hold:*

$$M\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, \mathbb{1}_{s^{\{\mathfrak{s}\}}} = (\lambda - 1)\, \mathbb{1}_{s^{\{\mathfrak{f}\}}} + c, \qquad \textit{(int. consistency)} \tag{2.9a}$$

$$\frac{M}{6} = \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c, \qquad \textit{(order 3)} \tag{2.9b}$$

$$\frac{M^2}{8} = \sum_{\lambda=1}^{M} (\lambda - 1)\, b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c + \sum_{\lambda=1}^{M} (b \times c)^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c, \qquad \textit{(order 4)} \tag{2.9c}$$

$$\frac{M}{12} = \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c^2, \qquad \textit{(order 4)} \tag{2.9d}$$

$$\frac{M^2}{24} = \sum_{\lambda=1}^{M} b^T\, A\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c + \sum_{\lambda=1}^{M} (M - \lambda)\, b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c, \qquad \textit{(order 4)} \tag{2.9e}$$

$$\frac{M}{24} = \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, A\, c. \qquad \textit{(order 4)} \tag{2.9f}$$

Proof. From [132], an internally consistent GARK method has order four if and only the base methods have order four and the following coupling conditions hold.

Condition 3a:

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} = \frac{1}{M} \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c.$$

Condition 3b:

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} = b^T\, A\, c.$$

Condition 4a:

$$\begin{aligned}
\frac{1}{8} &= \left(\mathbf{b}^{\{\mathfrak{f}\}} \times \mathbf{c}^{\{\mathfrak{f}\}}\right)^T \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} \\
&= \frac{1}{M^2} \sum_{\lambda=1}^{M} (b \times (c + (\lambda - 1)\, \mathbb{1}_s))^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c^{\{\mathfrak{s}\}}. \\
&= \frac{1}{M^2} \sum_{\lambda=1}^{M} (\lambda - 1)\, b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c + \frac{1}{M^2} \sum_{\lambda=1}^{M} (b \times c)^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c
\end{aligned}$$

Condition 4b:

$$\frac{1}{8} = \left(\mathbf{b}^{\{\mathfrak{s}\}} \times \mathbf{c}^{\{\mathfrak{s}\}}\right)^T \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} = (b \times c)^T\, A\, c.$$

Condition 4c:

$$\frac{1}{12} = \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}\times 2} = \frac{1}{M} \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c^2.$$

Condition 4d:

$$\frac{1}{12} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}\times 2} = b^T\, A\, c^2.$$

Condition 4e:

$$\begin{aligned}
\frac{1}{24} &= \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} \\
&= \frac{1}{M^2} \sum_{\lambda=1}^{M} \left(b^T\, A + \sum_{k=1}^{M-\lambda} b^T\, \mathbb{1}_s\, b^T \right) A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c \\
&= \frac{1}{M^2} \sum_{\lambda=1}^{M} b^T\, A\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c + \frac{1}{M^2} \sum_{\lambda=1}^{M} (M-\lambda)\, b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, c.
\end{aligned}$$

Condition 4f:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} = \frac{1}{M} \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, A\, c.$$

Condition 4g:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} = \frac{1}{M} \sum_{\lambda=1}^{M} b^T\, A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, A\, c.$$

Condition 4h:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} = b^T\, A\, A\, c.$$

Condition 4i:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} = b^T\, A\, A\, c.$$

Condition 4j:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} = b^T\, A\, A\, c.$$

Note that conditions 3b, 4b, 4d, and 4h–j resolve to order conditions of the base method, and thus, are satisfied if and only if the base method has order four. Further, condition 4g is identical to 4f. The remaining order conditions give (2.9). □

2.3 Multirate Linear Stability Analysis

In the analysis of single rate Runge–Kutta methods, it is common to apply methods to the Dahlquist test problem

$$y' = \lambda\, y, \tag{2.10}$$

with $\lambda \in \mathbb{C}^- = \{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$. This yields the well-known linear stability function

$$R(z) = 1 + z\, b^T\, (I - z\, A)^{-1}\, \mathbb{1}_s, \tag{2.11}$$

where $z = \lambda\, h$. It suffices to only examine a scalar problem (2.10) because in the case λ and z are matrices, the behavior of $R(z)^m$ as $m \to \infty$ only depends on the scalar eigenvalues of z. That is, the choice of basis for a system of linear ODEs does not affect a Runge–Kutta method's stability.

As noted by Gear [61], this property does not hold for multirate and other partitioned schemes. For this reason, the linear stability analysis becomes significantly more complex. In this section, we analyze and compare linear stability for both scalar and two-dimensional (2D) test problems. The scalar test problem is a simple model of an additively partitioned system (2.1) where the Jacobians of the two processes triangularize simultaneously. The 2D problem is a simple model for a component partitioned system (2.2), where each component's dynamics as well as the interaction between components are linear.

In this section, we will focus on two-partitioned GARK methods for simplicity. Nearly all of the stability analysis, however, has straightforward generalizations to the full N-partitioned GARK framework.

2.3.1 Scalar Test Problem

The simplest generalization of (2.10) for two-partitioned multirate methods is the scalar test problem

$$y' = \lambda^{\{\mathfrak{f}\}}\, y + \lambda^{\{\mathfrak{s}\}}\, y, \tag{2.12}$$

where, $\lambda^{\{\mathfrak{f}\}}, \lambda^{\{\mathfrak{s}\}} \in \mathbb{C}^-$. As shown in [132], when (2.6) is applied to (2.12), we arrive at the stability function

$$R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) = 1 + \mathbf{b}^T\, Z\, (I_{\mathbf{s}\times\mathbf{s}} - \mathbf{A}\, Z)^{-1}\, \mathbb{1}_{\mathbf{s}}, \tag{2.13}$$

where $z^{\{\mathfrak{f}\}} = H\, \lambda^{\{\mathfrak{f}\}}$, $z^{\{\mathfrak{s}\}} = H\, \lambda^{\{\mathfrak{s}\}}$, and

$$Z = \begin{bmatrix} z^{\{\mathfrak{f}\}}\, I_{\mathbf{s}^{\{\mathfrak{f}\}}\times\mathbf{s}^{\{\mathfrak{f}\}}} & 0 \\ 0 & z^{\{\mathfrak{s}\}}\, I_{\mathbf{s}^{\{\mathfrak{s}\}}\times\mathbf{s}^{\{\mathfrak{s}\}}} \end{bmatrix}.$$

Definition 2.2 (Scalar region of absolute stability)**.** The set

$$S_1 = \big\{(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}) \in \mathbb{C}\times\mathbb{C} : \big|R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big)\big| \leq 1\big\}$$

is the region of absolute stability for the test problem (2.12). A GARK method is called scalar A-stable if $S_1 \supseteq \mathbb{C}^- \times \mathbb{C}^-$. Further, a GARK method is called scalar L-stable if it is scalar A-stable,

$$\lim_{z^{\{\mathfrak{f}\}} \to \infty} R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) = 0, \quad \text{and} \quad \lim_{z^{\{\mathfrak{s}\}} \to \infty} R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) = 0. \tag{2.14}$$

Definition 2.3 (Scalar A(α)- and L(α)-stability)**.** A GARK method is scalar A(α)-stable if $S_1 \supseteq W(\alpha) \times W(\alpha)$, where $W(\alpha)$ is the wedge $\{z \in \mathbb{C} : |\arg(-z)| < \alpha, z \neq 0\}$. A scalar A($\alpha$)-stable GARK method that additionally satisfies (2.14) is called scalar L(α)-stable.

One way to determine if a single rate Runge–Kutta method is stable in the entire left half-plane is by ensuring stability on the imaginary axis and that the poles of $R(z)$ are in the right half-plane [72, Section IV.3]. Further, stability on the imaginary axis is equivalent to the E-polynomial

$$E(y) = Q(i\,y)\,Q(-i\,y) - P(i\,y)\,P(-i\,y)$$

being nonnegative for all $y \in \mathbb{R}$. Here, P and Q are the numerator and denominator of (2.11), respectively. As we will now show, these practical techniques for determining linear stability have simple and direct generalizations for GARK methods applied to (2.12).

Theorem 2.4 (Necessary and sufficient condition for scalar A-stability)**.** *The GARK method* (2.3) *is scalar A-stable if and only if*

$$\big|R_1\big(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\big)\big| \leq 1 \qquad \textit{for all } y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}} \in \mathbb{R} \tag{2.15}$$

and R_1 is analytic over $\mathbb{C}^- \times \mathbb{C}^-$.

Proof. This follows from the multivariate maximum principle (see for example [140]).

□

Remark 2.5 (Finding $A(\alpha)$-stability regions)**.** The maximum principle can also be used to efficiently determine the angle for scalar A(α)-stability. Instead of ensuring stability for all points inside a 4D wedge $W(\alpha) \times W(\alpha)$, one can limit the analysis to the boundary points $\partial W(\alpha) \times \partial W(\alpha)$.

Notably, Proposition 2.4 reduces the space on which we have to check for A-stability from four to two dimensions. For multirate methods, however, R_1 is different for each value of M, thus adding another dimension to consider.

Theorem 2.6 (E-polynomial)**.** *The E-polynomial for GARK methods is*

$$\begin{aligned} E_1\big(y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\big) = {} & Q_1\big(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\big)\,Q_1\big(-i\,y^{\{\mathfrak{f}\}}, -i\,y^{\{\mathfrak{s}\}}\big) \\ & - P_1\big(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\big)\,P_1\big(-i\,y^{\{\mathfrak{f}\}}, -i\,y^{\{\mathfrak{s}\}}\big), \end{aligned}$$

where P_1 and Q_1 are the numerator and denominator of (2.13)*, respectively. The scalar stability region of a method contains the imaginary axes if and only if the E-polynomial is nonnegative for all $y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}} \in \mathbb{R}$.*

Proof. Following the single rate approach presented in [72, Section IV.3], we have that

$$\begin{aligned}
&1 \geq \left|R_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)\right|^2 \\
&0 \leq \left|Q_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)\right|^2 - \left|P_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)\right|^2 \\
&0 \leq Q_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)\overline{Q_1(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}})} - P_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)\overline{P_1(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}})} \\
&0 \leq Q_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)Q_1\left(-i\,y^{\{\mathfrak{f}\}}, -i\,y^{\{\mathfrak{s}\}}\right) - P_1\left(i\,y^{\{\mathfrak{f}\}}, i\,y^{\{\mathfrak{s}\}}\right)P_1\left(-i\,y^{\{\mathfrak{f}\}}, -i\,y^{\{\mathfrak{s}\}}\right) \\
&0 \leq E_1\left(y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right).
\end{aligned}$$

Since each of these inequalities is equivalent, the statement is proven. □

2.3.2 2D Test Problem

Another test problem, first proposed in [60], and later used in [43, 80, 93, 137], is the 2D linear test problem

$$\begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}' = \underbrace{\begin{bmatrix} \lambda^{\{\mathfrak{f}\}} & \eta^{\{\mathfrak{s}\}} \\ \eta^{\{\mathfrak{f}\}} & \lambda^{\{\mathfrak{s}\}} \end{bmatrix}}_{\Lambda} \begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}. \tag{2.16}$$

Here, the exact solution must be bounded. That is, the eigenvalues of Λ have nonpositive real parts and eigenvalues on the imaginary axis are regular. Further, we enforce that $\lambda^{\{\mathfrak{f}\}}, \lambda^{\{\mathfrak{s}\}} \in \mathbb{C}^-$ so the individual partitions have bounded dynamics. We will denote the set of these special exponentially bounded matrices by $\mathbb{M}$, and this test problem will be referred to as the complex 2D test problem. Many authors have considered simplifying assumptions including restricting Λ to real entries. In this case, the constraints on Λ simplify to $\lambda^{\{\mathfrak{f}\}}, \lambda^{\{\mathfrak{s}\}} \leq 0$, $\eta^{\{\mathfrak{f}\}}\eta^{\{\mathfrak{s}\}} \leq \lambda^{\{\mathfrak{f}\}}\lambda^{\{\mathfrak{s}\}}$, and a zero eigenvalue must be regular. We will refer to this problem as the real 2D test problem.

When (2.3) is applied to (2.16), we arrive at the stability matrix

$$\begin{aligned}
&R_2\left(\begin{bmatrix} z^{\{\mathfrak{f}\}} & w^{\{\mathfrak{s}\}} \\ w^{\{\mathfrak{f}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix}\right) \\
&= I_{2\times 2} + \begin{bmatrix} \mathbf{b}^{\{\mathfrak{f}\}T} & 0 \\ 0 & \mathbf{b}^{\{\mathfrak{s}\}T} \end{bmatrix} \begin{bmatrix} I_{\mathbf{s}^{\{\mathfrak{f}\}}\times\mathbf{s}^{\{\mathfrak{f}\}}} - z^{\{\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} & -w^{\{\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ -w^{\{\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & I_{\mathbf{s}^{\{\mathfrak{s}\}}\times\mathbf{s}^{\{\mathfrak{s}\}}} - z^{\{\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \end{bmatrix}^{-1} \\
&\quad \begin{bmatrix} z^{\{\mathfrak{f}\}}\,\mathbb{1}_{\mathbf{s}^{\{\mathfrak{f}\}}} & w^{\{\mathfrak{s}\}}\,\mathbb{1}_{\mathbf{s}^{\{\mathfrak{f}\}}} \\ w^{\{\mathfrak{f}\}}\,\mathbb{1}_{\mathbf{s}^{\{\mathfrak{s}\}}} & z^{\{\mathfrak{s}\}}\,\mathbb{1}_{\mathbf{s}^{\{\mathfrak{s}\}}} \end{bmatrix},
\end{aligned} \tag{2.17}$$

where

$$\begin{bmatrix} z^{\{\mathfrak{f}\}} & w^{\{\mathfrak{s}\}} \\ w^{\{\mathfrak{f}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix} = H \begin{bmatrix} \lambda^{\{\mathfrak{f}\}} & \eta^{\{\mathfrak{s}\}} \\ \eta^{\{\mathfrak{f}\}} & \lambda^{\{\mathfrak{s}\}} \end{bmatrix}.$$

Definition 2.7 (Complex 2D region of absolute stability)**.** The set

$$S_2 = \left\{Z \in \mathbb{C}^{2\times 2} : R_2(Z) \text{ power bounded}\right\}$$

is the complex 2D region of absolute stability for the test problem (2.16). A GARK method is called complex 2D A-stable if $S_2 \supseteq \mathbb{M}$.

Definition 2.8 (Real 2D region of absolute stability)**.** The set

$$\widehat{S}_2 = \left\{ Z \in \mathbb{R}^{2\times 2} : R_2(Z) \text{ power bounded} \right\}$$

is the real 2D region of absolute stability for the test problem (2.16). A GARK method is called real 2D A-stable if $\widehat{S}_2 \supseteq (\mathbb{M} \cap \mathbb{R}^{2\times 2})$.

For both cases of the 2D test problem, the power boundedness condition makes finding necessary and sufficient conditions for stability significantly more challenging. Considering test problems on the boundary of $\mathbb{M}$ does provide important necessary conditions. Consider the particular test problem

$$y' = \begin{bmatrix} 0 & \eta \\ -\eta & 0 \end{bmatrix} y, \tag{2.18}$$

which has purely imaginary eigenvalues for $\eta \in \mathbb{R}$. Note that

$$\begin{aligned} R_2\left(\begin{bmatrix} 0 & w \\ -w & 0 \end{bmatrix}\right) &= I_{2\times 2} + w \begin{bmatrix} \mathbf{b}^{\{\mathfrak{f}\}T} & 0 \\ 0 & \mathbf{b}^{\{\mathfrak{s}\}T} \end{bmatrix} \begin{bmatrix} I_{\mathbf{s}^{\{\mathfrak{f}\}}\times\mathbf{s}^{\{\mathfrak{f}\}}} & -w\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ w\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & I_{\mathbf{s}^{\{\mathfrak{s}\}}\times\mathbf{s}^{\{\mathfrak{s}\}}} \end{bmatrix}^{-1} \begin{bmatrix} 0 & \mathbb{1}_{\mathbf{s}^{\{\mathfrak{f}\}}} \\ -\mathbb{1}_{\mathbf{s}^{\{\mathfrak{s}\}}} & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 - w^2\,\mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{d}^{\{\mathfrak{s}\}} & w\,\mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{d}^{\{\mathfrak{f}\}} \\ -w\,\mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{d}^{\{\mathfrak{s}\}} & 1 - w^2\,\mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{d}^{\{\mathfrak{f}\}} \end{bmatrix}, \end{aligned} \tag{2.19}$$

where $w = H\,\eta$ and

$$\begin{aligned} \mathbf{d}^{\{\mathfrak{f}\}} &= \left(I_{\mathbf{s}^{\{\mathfrak{f}\}}\times\mathbf{s}^{\{\mathfrak{f}\}}} + w^2\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\right)^{-1} \mathbb{1}_{\mathbf{s}^{\{\mathfrak{f}\}}}, \\ \mathbf{d}^{\{\mathfrak{s}\}} &= \left(I_{\mathbf{s}^{\{\mathfrak{s}\}}\times\mathbf{s}^{\{\mathfrak{s}\}}} + w^2\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\right)^{-1} \mathbb{1}_{\mathbf{s}^{\{\mathfrak{s}\}}}. \end{aligned}$$

An important property of this stability function, which will be used later for proposition 2.12, is that it depends on the coupling coefficients but not the base method coefficients $A^{\{\mathfrak{f},\mathfrak{f}\}}$ and $A^{\{\mathfrak{s},\mathfrak{s}\}}$.

Remark 2.9 (Other test problems)**.** The 2D problem can be generalized to the linear block system

$$\begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}' = \begin{bmatrix} \Lambda^{\{\mathfrak{f}\}} & E^{\{\mathfrak{s}\}} \\ E^{\{\mathfrak{f}\}} & \Lambda^{\{\mathfrak{s}\}} \end{bmatrix} \begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}. \tag{2.20}$$

This problem has been considered in [10, 61]. An even more general block system was used by Skelboe in [148]. We do not consider these block generalizations further as we find that the 2D problem already poses a surprisingly challenging test problem.

2.3.3 Comparison of Stability Test Problems

When designing an implicit method, unconditional stability is a highly desirable property. A natural question is which test problem should be used to determine stability. In this section, we explore the relationships among the different stability criteria in order to address this question. Consider, for example, the GARK method given by the tableau below:

$$\begin{array}{c|c} 1 & 0 \\ \hline 1 & 1 \\ \hline 1 & 1 \end{array}.$$

This method is scalar L-stable and even algebraically stable [132], but

$$\rho\left(R_2\left(\begin{bmatrix} -1 & 1 \\ -10 & -1 \end{bmatrix}\right)\right) = \frac{\sqrt{5}+3}{4} > 1,$$

with ρ the spectral radius operator. Thus, it is only conditionally stable for the real and complex 2D test problems.

Conversely, consider the GARK method

$$\begin{array}{cc|cc} \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \hline \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \hline \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{array}.$$

The base method is only A(45°) stable, and thus, it is easy to show the GARK method is conditionally stable with respect to the scalar test problem:

$$|R_1(-4+8i, 0)| = \frac{\sqrt{17}}{4} > 1. \tag{2.21}$$

For the real 2D test problem, this GARK method is A-stable. This result reveals a shortcoming of the real 2D test problem: the individual partitions have purely real eigenvalues. Ideally, a test problem should reveal instabilities of the base methods off the real axis. Despite the apparent independence of the stability functions (2.13) and (2.17), we do note that

$$R_1\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) = \begin{bmatrix} z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix} R_2\left(\begin{bmatrix} z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{s}\}} \\ z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix}\right) \begin{bmatrix} \frac{\alpha}{z^{\{\mathfrak{f}\}}} \\ \frac{1-\alpha}{z^{\{\mathfrak{s}\}}} \end{bmatrix}, \tag{2.22a}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} R_2\left(\begin{bmatrix} z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{f}\}} \\ z^{\{\mathfrak{s}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix}\right) \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix}, \tag{2.22b}$$

for any $\alpha \in \mathbb{C}$.

When (2.16) is taken to have complex entries, however, there is a meaningful connection to the scalar test problem.

Theorem 2.10. *If a GARK method is A-stable with respect to the complex 2D test problem, then it is A-stable with respect to the scalar test problem.*

Proof. First, we define

$$R_2\left(\begin{bmatrix} z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{f}\}} \\ z^{\{\mathfrak{s}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix}\right) = \begin{bmatrix} r_{1,1} & r_{1,2} \\ r_{2,1} & r_{2,2} \end{bmatrix}. \tag{2.23}$$

Since (2.22b) must hold for all α,

$$\text{const} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} r_{1,1} & r_{1,2} \\ r_{2,1} & r_{2,2} \end{bmatrix} \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix} = \alpha\left(r_{1,1} + r_{2,1} - r_{1,2} - r_{2,2}\right) + r_{1,2} + r_{2,2}.$$

Thus, $r_{1,1} + r_{2,1} - r_{1,2} - r_{2,2} = 0$ and

$$R_2\left(\begin{bmatrix} z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{f}\}} \\ z^{\{\mathfrak{s}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix}\right) = \begin{bmatrix} r_{1,1} & r_{1,2} \\ r_{2,1} & r_{1,1} + r_{2,1} - r_{1,2} \end{bmatrix}. \tag{2.24}$$

Due to this structure, $r_{1,1} + r_{2,1}$ is an eigenvalue, and if a GARK method is A-stable for the 2D test problem, then $|r_{1,1} + r_{2,1}| \leq 1$. Using (2.22b) with $\alpha = 1$, we have that

$$\begin{aligned} R_1\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) &= \begin{bmatrix} 1 & 1 \end{bmatrix} R_2\left(\begin{bmatrix} z^{\{\mathfrak{f}\}} & z^{\{\mathfrak{f}\}} \\ z^{\{\mathfrak{s}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix}\right) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= r_{1,1} + r_{2,1} \\ \left|R_1\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right)\right| &\leq 1. \end{aligned}$$

Thus, the method is A-stable for the scalar test problem. □

While the 2D test problem may be a more thorough, reliable, and informative method of assessing stability, it is also more difficult to analyze and visualize due to the high-dimensional space of test problems. We summarize the hierarchy of linear stability properties in Figure 2.1.

Lemma 2.11. *For a decoupled GARK method, the following matrix is nilpotent:*

$$\begin{bmatrix} 0 & \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & 0 \end{bmatrix}.$$

Proof. The full matrix $\mathbf{A}$ can be viewed as the adjacency matrix of a weighted directed graph. Cycles indicate the method is implicit, and by the definition of a decoupled method, implicitness only comes from the base methods. With the base method coefficients set to zero, the directed graph becomes acyclic: a property equivalent to nilpotency of the adjacency matrix. □

$$
\begin{tikzcd}
\text{Complex 2D A-stability} \arrow[r, Rightarrow] \arrow[dr, Rightarrow] & \text{Real 2D A-stability} & \\
\text{Algebraic stability} \arrow[r, Rightarrow] & \text{Scalar A-stability} \arrow[r, Rightarrow] & \text{Scalar A}(\alpha)\text{-stability} \\
\text{Scalar L-stability} \arrow[r, Rightarrow] \arrow[ur, Rightarrow] & \text{Scalar L}(\alpha)\text{-stability} \arrow[ur, Rightarrow] &
\end{tikzcd}
$$

Figure 2.1: Stability implications for the various linear test problems. In general, no implication arrows are reversible.

Theorem 2.12. *A decoupled GARK method consistent with (2.1) (first order accurate) cannot be A-stable for the real 2D test problem.*

Proof. Consider the particular test problem given in (2.18). Note that in (2.19), the matrix being inverted is the sum of an identity matrix and a nilpotent matrix by the decoupled assumption and proposition 2.11. Expanding the inverse in a Neumann series reveals $\mathbf{d}^{\{\mathfrak{f}\}}$ and $\mathbf{d}^{\{\mathfrak{s}\}}$ must be even polynomials in w of finite degree. Moreover, the off-diagonal terms of the stability matrix satisfy

$$
\begin{aligned}
w\,\mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{d}^{\{\mathfrak{f}\}} &= w\,\mathbf{b}^{\{\mathfrak{f}\}T}\left(I + w^2\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} + \ldots\right)\mathbb{1}_{\mathbf{s}^{\{\mathfrak{f}\}}} &&= w + w^3\,p_{1,2}(w^2),\\
-w\,\mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{d}^{\{\mathfrak{s}\}} &= -w\,\mathbf{b}^{\{\mathfrak{s}\}T}\left(I + w^2\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} + \ldots\right)\mathbb{1}_{\mathbf{s}^{\{\mathfrak{s}\}}} &&= -w - w^3\,p_{2,1}(w^2),
\end{aligned}
$$

where $p_{1,2}$ and $p_{2,1}$ are polynomials. Note the consistency assumption implies $\mathbf{b}^{\{\mathfrak{f}\}T}\mathbb{1}_{\mathbf{s}^{\{\mathfrak{f}\}}} = \mathbf{b}^{\{\mathfrak{s}\}T}\mathbb{1}_{\mathbf{s}^{\{\mathfrak{s}\}}} = 1$ and is used to determine the coefficient multiplying the w terms. Now the stability matrix can be written in the form

$$
R_2(w) = \begin{bmatrix} 1 - w^2\,p_{1,1}(w^2) & w + w^3\,p_{1,2}(w^2) \\ -w - w^3\,p_{2,1}(w^2) & 1 - w^2\,p_{2,2}(w^2) \end{bmatrix},
$$

where $p_{1,1}$, and $p_{2,2}$ are also polynomials.

Suppose by means of contradiction that the method is A-stable. Consider the trace of the stability matrix:

$$
\operatorname{tr}(R_2(w)) = 2 - w^2\,(p_{1,1}(w^2) + p_{2,2}(w^2)).
$$

In order to avoid an eigenvalue of $R_2(w)$ being unbounded in w, we must have that $p_{2,2}(w^2) = -p_{1,1}(w^2)$. Using this necessary condition, the determinant is

$$
\begin{aligned}
\det(R_2(w)) &= (1 - w^2\,p_{1,1}(w^2))(1 + w^2\,p_{1,1}(w^2)) \\
&\quad - (w + w^3\,p_{1,2}(w^2))(-w - w^3\,p_{2,1}(w^2)) \\
&= 1 + w^2 + \mathcal{O}\big(w^4\big).
\end{aligned}
$$

Since the determinant grows unbounded in w, the spectral radius can be made arbitrarily large. This is a contradiction. Therefore, the method cannot be A-stable for the real 2D test problem. □

2.3.4 Compound-Fast Scalar Stability

Directly using the general stability formula (2.13) on an MrGARK method requires inverting a matrix of size $\mathbf{s}\times\mathbf{s}$. When trying to analyze or visualize the linear stability for large M, this becomes very expensive. Fortunately, the particular structure of compound-fast MrGARK methods allows for an explicit derivation of the scalar stability function using only matrices of size $s \times s$.

For the base method (A, b, c), let $R_{\text{int}}(z)$ be the internal stability function:

$$R_{\text{int}}(z) = (I_{s\times s} - z\,A)^{-1}\,\mathbb{1}_s.$$

We now seek to find the scalar internal stability of a compound-fast MrGARK method. Let $z = z^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}$. Then the first macro-step (2.8a) is composed of traditional Runge–Kutta stages and is simply

$$Y = y_n\,R_{\text{int}}(z) \tag{2.25}$$

for the scalar linear test problem. The λ-th fast micro-step (2.8b) has stages defined by the recurrence relation

$$Y^{\{\mathfrak{f},\lambda\}} = y_n\,\mathbb{1}_s + \frac{z^{\{\mathfrak{f}\}}}{M}\sum_{k=1}^{\lambda-1}\mathbb{1}_s\,b^T\,Y^{\{\mathfrak{f},k\}} + \frac{z^{\{\mathfrak{f}\}}}{M}\,A\,Y^{\{\mathfrak{f},\lambda\}} + z^{\{\mathfrak{s}\}}\,A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\,Y.$$

Solving for $Y^{\{\mathfrak{f},\lambda\}}$ explicitly is equivalent to solving the following linear system via block forward substitution:

$$\begin{bmatrix} I - \frac{z^{\{\mathfrak{f}\}}}{M}\,A & \dots & 0 \\ \vdots & \ddots & \vdots \\ -\frac{z^{\{\mathfrak{f}\}}}{M}\,\mathbb{1}_s\,b^T & \dots & I - \frac{z^{\{\mathfrak{f}\}}}{M}\,A \end{bmatrix} \begin{bmatrix} Y^{\{\mathfrak{f},1\}} \\ \vdots \\ Y^{\{\mathfrak{f},M\}} \end{bmatrix} = \begin{bmatrix} y_n\mathbb{1}_s + z^{\{\mathfrak{s}\}}\,A^{\{\mathfrak{f},\mathfrak{s},1\}}\,Y \\ \vdots \\ y_n\mathbb{1}_s + z^{\{\mathfrak{s}\}}\,A^{\{\mathfrak{f},\mathfrak{s},M\}}\,Y \end{bmatrix}.$$

This yields

$$\begin{aligned} & Y^{\{\mathfrak{f},\lambda\}} \\ &= \frac{z^{\{\mathfrak{f}\}}}{M}\,R_{\text{int}}\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right) b^T \left(I - \frac{z^{\{\mathfrak{f}\}}}{M}A\right)^{-1} \sum_{k=1}^{\lambda-1} R\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right)^{\lambda-1-k} (y_n\,\mathbb{1}_s \\ &\quad + z^{\{\mathfrak{s}\}}\,A^{\{\mathfrak{f},\mathfrak{s},k\}}\,Y) + \left(I - \frac{z^{\{\mathfrak{f}\}}}{M}\,A\right)^{-1} \left(y_n\,\mathbb{1}_s + z^{\{\mathfrak{s}\}}\,A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\,Y\right) \\ &= \left(\frac{z^{\{\mathfrak{f}\}}\,z^{\{\mathfrak{s}\}}}{M}\,R_{\text{int}}\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right) b^T \left(I - \frac{z^{\{\mathfrak{f}\}}}{M}\,A\right)^{-1} \sum_{k=1}^{\lambda-1} R\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right)^{\lambda-1-k} A^{\{\mathfrak{f},\mathfrak{s},k\}}\,R_{\text{int}}(z) \right. \\ &\quad \left. + z^{\{\mathfrak{s}\}}\left(I - \frac{z^{\{\mathfrak{f}\}}}{M}\,A\right)^{-1} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\,R_{\text{int}}(z) + R\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right)^{\lambda-1} R_{\text{int}}\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right)\right) y_n. \end{aligned} \tag{2.26}$$

Together, (2.25) and (2.26) form the internal stability for a compound-fast MrGARK method. With this in hand, the scalar linear stability function (2.13) can be derived:

$$\begin{aligned}
&R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) \\
&= 1 + \frac{z^{\{\mathfrak{f}\}}}{M} \sum_{\lambda=1}^{M} b^T\, Y^{\{\mathfrak{f},\mathfrak{s},\lambda\}} + z^{\{\mathfrak{s}\}}\, b^T\, Y \\
&= R\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right)^M + z^{\{\mathfrak{s}\}} b^T R_{\text{int}}(z) \\
&\quad + \frac{z^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{f}\}}}{M}\, b^T \left(I_{s\times s} - \frac{z^{\{\mathfrak{f}\}}}{M}\, A\right)^{-1} \sum_{\lambda=1}^{M} R\left(\frac{z^{\{\mathfrak{f}\}}}{M}\right)^{M-\lambda} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, R_{\text{int}}(z).
\end{aligned}$$

If A is invertible, then $R_{\text{int}}(-\infty) = 0_s$ and

$$\begin{aligned}
\lim_{z^{\{\mathfrak{f}\}}\to-\infty} R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) &= R(-\infty)^M + z^{\{\mathfrak{s}\}} \left(b^T - b^T\, A^{-1}\, A^{\{\mathfrak{f},\mathfrak{s},M\}}\right) R_{\text{int}}(-\infty) \\
&= R(-\infty)^M.
\end{aligned}$$

The other limit is more difficult to approach directly, so we consider first the internal stability (2.26). Starting with the first micro-step, we have that

$$\lim_{z^{\{\mathfrak{s}\}}\to-\infty} Y^{\{\mathfrak{f},1\}} = \left(I - \frac{z^{\{\mathfrak{f}\}}}{M} A\right)^{-1} \left(\mathbb{1}_s - A^{\{\mathfrak{f},\mathfrak{s},1\}} A^{-1}\, \mathbb{1}_s\right) y_n.$$

This suggests the condition $A^{\{\mathfrak{f},\mathfrak{s},1\}}\, A^{-1}\, \mathbb{1}_s = \mathbb{1}_s$ to ensure the stage values go to zero in the limit. Now we can use an inductive argument to generalize this condition for the remaining micro-step stages. Assume that $\lim_{z^{\{\mathfrak{s}\}}\to-\infty} Y^{\{\mathfrak{f},\ell\}} = 0$ for $\ell = 1, \ldots, \lambda - 1$. Then

$$\lim_{z^{\{\mathfrak{s}\}}\to-\infty} Y^{\{\mathfrak{f},\lambda\}} = \left(I_{s\times s} - \frac{z^{\{\mathfrak{f}\}}}{M}\, A\right)^{-1} \left(\mathbb{1}_s - A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, A^{-1}\, \mathbb{1}_s\right)\, y_n.$$

This suggests the condition

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\, A^{-1}\, \mathbb{1}_s = \mathbb{1}_s \qquad \lambda = 1, \ldots, M \tag{2.27}$$

to ensure all stages go to zero in the limit. Further (2.27) leads to the result

$$\lim_{z^{\{\mathfrak{s}\}}\to-\infty} R_1\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) = R(-\infty) + \frac{z^{\{\mathfrak{f}\}}}{M} \sum_{\lambda=1}^{M} b^T\, Y^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = R(-\infty).$$

2.4 Numerical Solution of Implicit Stage Equations

The key to an efficient implicit GARK method is an efficient Newton iteration. Written compactly, the stage equations are

$$\widehat{Y} = \mathbb{1}_{\mathbf{s}} \otimes y_n + H\left(\mathbf{A} \otimes I_{d\times d}\right) \widehat{f}\left(\widehat{Y}\right), \tag{2.28}$$

where

$$\widehat{Y} = \begin{bmatrix} Y^{\{\mathfrak{f}\}} \\ Y^{\{\mathfrak{s}\}} \end{bmatrix}, \qquad \widehat{f}\left(\widehat{Y}\right) = \begin{bmatrix} f^{\{\mathfrak{f}\}}(Y^{\{\mathfrak{f}\}}) \\ f^{\{\mathfrak{s}\}}(Y^{\{\mathfrak{s}\}}) \end{bmatrix}. \tag{2.29}$$

Applying Newton's method to solve for the stages yields the iterative procedure

$$\left(I_{\mathbf{s}\times\mathbf{s}} \otimes I_{d\times d} - H\left(\mathbf{A} \otimes I_{d\times d}\right) \widehat{J}\right) \delta = -\widehat{Y} + \mathbb{1}_{\mathbf{s}} \otimes y_n + H\left(\mathbf{A} \otimes I_{d\times d}\right) \widehat{f}\left(\widehat{Y}\right), \tag{2.30a}$$

$$\widehat{Y} = \widehat{Y} + \delta, \tag{2.30b}$$

with

$$\widehat{J} = \operatorname{diag}\left(J_1^{\{\mathfrak{f}\}}, \ldots, J_{s^{\{\mathfrak{f}\}}}^{\{\mathfrak{f}\}}, J_1^{\{\mathfrak{s}\}}, \ldots, J_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}}\right), \tag{2.31}$$

and $J_i^{\{\sigma\}} = \frac{\partial f^{\{\sigma\}}}{\partial y}\left(Y_i^{\{\sigma\}}\right)$ for $\sigma \in \{\mathfrak{s}, \mathfrak{f}\}$.

In single rate Newton iterations, it is common to evaluate the Jacobian once at y_n and use it across all stages which yields a cheaper modified Newton's method. A similar strategy can be employed for each partition's Jacobian in a GARK Newton iteration. For multirate methods, it might be beneficial to reevaluate the fast Jacobian at each micro-step and keep the slow Jacobian across the entire macro-step.

We note that (2.30) serves mostly theoretical purposes, as it is impractically expensive and rarely necessary to simultaneously solve for all $\mathbf{s}$ stages. All methods presented in section 2.5, for example, require solving nonlinear systems with dimension no larger than d. In this section, we will explore techniques and method structures that allow for these efficient implementations of Newton iterations. In the cost analyses we present, matrix decompositions involving the Jacobians are assumed to be the dominant cost of a step.

2.4.1 Decoupled Methods

As described in section 2.2.1, decoupled methods only have implicitness in the base methods. For this subsection, we will assume both base methods are diagonally implicit which seems to be the most practical structure for decoupled implicit methods. Now, each of the $\mathbf{s}$ method stages defines a d-dimensional nonlinear equation which can be solved sequentially for a cost of $\mathcal{O}(\mathbf{s}\, d^3)$, assuming direct methods are used. If we further assume the slow matrix decomposition is reused across a multirate macro-step and the fast matrix decomposition

is reused across a micro-step, the cost is reduced to $\mathcal{O}(M\,d^3)$. It is important to note that the slow and fast Jacobians are likely to have simpler structures than the full Jacobian, and these structures can be exploited in the linear solves.

For the special case of component partitioned systems (2.2), the linear solves are of the reduced dimensions $d^{\{\mathfrak{f}\}}$ and $d^{\{\mathfrak{s}\}}$. In the most extreme case where each variable of a system forms a partition, a step would involve scalar Newton iterations for all variables and only the diagonal of the Jacobian of f would be required. We note, however, that this leads to an explosion in the number of coupling error terms and degraded stability.

2.4.2 Compound-Fast Methods

Compound-fast methods start by taking a full macro-step like a single rate Runge–Kutta method. Consequently, the nonlinear equations for the stages can be solved just as they would for a single rate method. When using Newton's method, the full, unpartitioned Jacobian is used. It may be appropriate to loosen the solver tolerances of the fast variables for the compound step as they will be recomputed later [157]. Although the remaining micro-steps are also implicitly defined, only $J_i^{\{\mathfrak{f}\}}$ is now involved in Newton iterations. Assuming a diagonally implicit structure for the base method, these Newton iterations are of the form

$$\left(I_{d\times d} - h\,a_{i,i}^{\{\mathfrak{f},\mathfrak{f}\}}\,J_i^{\{\mathfrak{f}\}}\right)\delta = -Y_i^{\{\mathfrak{f},\lambda\}} + \widetilde{y}_{n+(\lambda-1)/M} + h\sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}}\,f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f},\lambda\}}\right) + H\sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\,f^{\{\mathfrak{s}\}}(Y_j).$$

We note that an accurate stage value predictor to start the Newton iterations can come from dense output of the compound step.

In an implementation where a decomposition of the full matrix is formed once and a decomposition for the fast matrix is formed at each micro-step, the total cost for one step is $\mathcal{O}(M\,d^3)$. For component partitioned systems, this reduces to $\mathcal{O}\big(d^3 + M\,d^{\{\mathfrak{f}\}\times 3}\big)$.

2.4.3 Stage Reducibility

Consider the simple methods defined by the GARK tableaus (2.4) below:

$$\begin{array}{c|c} 1 & 1 \\ \hline 1 & 1 \\ \hline 1 & 1 \end{array} \quad \text{and} \quad \begin{array}{c|c} \frac{1}{2} & 1 \\ \hline \frac{1}{2} & 1 \\ \hline 1 & 1 \end{array}.$$

The former is backward Euler cast into the GARK framework. A direct application of (2.30) would require solving linear systems of size $2d$ when clearly solves of size d can suffice. Here, $Y_1^{\{\mathfrak{f}\}} = Y_1^{\{\mathfrak{s}\}}$, and these stages fall back onto the traditional backward Euler stage $Y_1 = y_n + Hf(Y_1)$. The latter method, which is an additive Runge–Kutta (ARK) method cast into the GARK framework, also has $Y_1^{\{\mathfrak{f}\}} = Y_1^{\{\mathfrak{s}\}}$. Equation (2.30a) can be simplified to

$$\left(I_{d\times d} - \frac{H}{2}\, J_1^{\{\mathfrak{f}\}} + H\, J_1^{\{\mathfrak{s}\}}\right)\delta = -Y_1 + y_n + \frac{H}{2}\, f^{\{\mathfrak{f}\}}(Y_1) + H\, f^{\{\mathfrak{s}\}}(Y_1). \tag{2.32}$$

More generally when a row of GARK coefficients is repeated in multiple partitions, the number of unknowns in (2.28) and the dimension of the Newton iteration is reduced. We call this *stage reducibility*. Compound-fast methods, for example, have this property in the first s stages.

In Section 2.5, we develop new multirate coupling strategies that utilize this simplification. An interesting property is that the solves involve matrices of the form $I_{d\times d} - h\,\gamma J_i^{\{\mathfrak{f}\}} - H\,\gamma J_i^{\{\mathfrak{s}\}}$. Note $J_i^{\{\mathfrak{f}\}}$ is scaled by the micro-step, while $J_i^{\{\mathfrak{s}\}}$ is scaled by the macro-step. If the multirate ratio is based on partition stiffness, then the scaled matrices should have similar spectral radii. By damping the fast, stiff modes, the conditioning of this system can be much better than the traditional $I_{d\times d} - H\,\gamma J_i$.

2.4.4 Low Rank Structure of Matrices in Newton Iteration

When a GARK method has stage reducibility, $\mathbf{A}$ cannot be full rank due to at least one repeated row. An alternative simplification arises by applying the Woodbury matrix identity to reduce the dimension of the linear solve. Using the GARK method below, we demonstrate that this idea can be extended to a broader set of schemes:

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline 1 & 1 \\ \hline 1 & 1 \end{array}.$$

We have the following simplification in the Newton iteration:

$$\begin{aligned}
&\left(I_{\mathbf{s}\times\mathbf{s}} \otimes I_{d\times d} - H\,(\mathbf{A}\otimes I_{d\times d})\,\widehat{J}\right)^{-1} \\
&= \left(\begin{bmatrix} I_{d\times d} & 0 \\ 0 & I_{d\times d}\end{bmatrix} - H\begin{bmatrix} \frac{1}{2}\,J_1^{\{\mathfrak{f}\}} & \frac{1}{2}\,J_1^{\{\mathfrak{s}\}} \\ J_1^{\{\mathfrak{f}\}} & J_1^{\{\mathfrak{s}\}}\end{bmatrix}\right)^{-1} \\
&= \begin{bmatrix} I_{d\times d} & 0 \\ 0 & I_{d\times d}\end{bmatrix} + \begin{bmatrix} \frac{H}{2}\,I_{d\times d} \\ H\,I_{d\times d}\end{bmatrix}\left(I_{d\times d} - \frac{H}{2}\,J_1^{\{\mathfrak{f}\}} - H\,J_1^{\{\mathfrak{s}\}}\right)^{-1}\begin{bmatrix} J_1^{\{\mathfrak{f}\}} & J_1^{\{\mathfrak{s}\}}\end{bmatrix}.
\end{aligned}$$

Compared to (2.32), additional matrix-vector products are required, but ultimately, the same matrix inverse appears. Thus, the potential to have improved conditioning is still present.

2.5 Practical implicit MrGARK methods

In this section, we present new implicit MrGARK methods of orders one to four. All methods are telescopic and based on single singly diagonally implicit Runge–Kutta (SDIRK) methods. At high order, coupling coefficients can become complicated rational functions of λ and M.

2.5.1 First Order

Multirate methods of order one have no coupling conditions which allows a great amount of freedom in deriving coefficients, but for implicit methods, stability does impose some important constraints. Proposition 2.13 eliminates one subset of first order methods from being scalar A-stable.

Theorem 2.13. *An internally consistent MrGARK method of order exactly one is only scalar A-stable for a finite number of multirate ratios.*

Proof. Using the internal consistency assumptions, the magnitude of the scalar stability function can be expanded as

$$\begin{aligned}\left|R_1(i\,\omega^{\{\mathfrak{f}\}}\,y, i\,\omega^{\{\mathfrak{s}\}}\,y)\right|^2 &= 1 + y^2\left(\omega^{\{\mathfrak{f}\}} + \omega^{\{\mathfrak{s}\}}\right)^2 \\ &\quad - 2y^2\left(\left(\omega^{\{\mathfrak{f}\}} + \omega^{\{\mathfrak{s}\}}\right)\left(\omega^{\{\mathfrak{f}\}}\,\mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{c}^{\{\mathfrak{f}\}} + \omega^{\{\mathfrak{s}\}}\,\mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{c}^{\{\mathfrak{s}\}}\right)\right) \\ &\quad + \mathcal{O}\big(y^4\big) \\ &= 1 + y^2\,p\big(\omega^{\{\mathfrak{f}\}}, \omega^{\{\mathfrak{s}\}}\big) + \mathcal{O}\big(y^4\big).\end{aligned} \tag{2.33}$$

Let H be the Hessian matrix of the homogeneous polynomial of degree two p. Note that $\det(H) = -4\left(r^{\{\mathfrak{f}\}} - r^{\{\mathfrak{s}\}}\right)^2$, where $r^{\{\mathfrak{f}\}} = \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{c}^{\{\mathfrak{f}\}} - \frac{1}{2}$ and $r^{\{\mathfrak{s}\}} = \mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{c}^{\{\mathfrak{s}\}} - \frac{1}{2}$ which are the second order residuals. These residuals cannot both be zero because the GARK method would be order two by internal consistency. When one base method is order one and the other is higher order, these residuals must differ. Otherwise, when both base methods have order one, $r^{\{\mathfrak{f}\}}$ is a function of M which approaches zero while $r^{\{\mathfrak{s}\}}$ is a fixed nonzero constant. For all but a finite set of M, these residuals must differ. Whenever the residuals differ, p is saddle-shaped, and there exist $\omega^{\{\mathfrak{f}\}}$ and $\omega^{\{\mathfrak{s}\}}$ such that the polynomial is positive. For sufficiently small values of y, the positive $y^2\,p\big(\omega^{\{\mathfrak{f}\}}, \omega^{\{\mathfrak{s}\}}\big)$ term will dominate the $\mathcal{O}(y^4)$ term in (2.33). Thus, for all but a finite set of M, there are $\omega^{\{\mathfrak{f}\}}$, $\omega^{\{\mathfrak{s}\}}$, and y such that $\left|R_1(i\,\omega^{\{\mathfrak{f}\}}\,y, i\,\omega^{\{\mathfrak{s}\}}\,y)\right| > 1$. □

Remark 2.14. Note that proposition 2.13 imposes no restriction on the multirate strategy. It only requires the defining characteristic of a multirate method: the fast error asymptotically approaches zero as M increases.

At first order, the natural choice for an implicit base method is backward Euler. There is currently a plethora of multirate backward Euler schemes in the literature (see [70, 128, 158, 166]). These schemes feature nearly all the different combinations of coupled or decoupled, internal consistency or internal inconsistency, and parallel or sequential methods. In the search for a multirate backward Euler method with excellent stability and accuracy properties, we developed the coupling strategy given by the following standard MrGARK coupling coefficients:

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \begin{cases} 0 & \lambda < \frac{M}{2} \\ 1 & \text{otherwise} \end{cases} \end{bmatrix}, \qquad A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} \begin{cases} 1 & \lambda \leq \frac{M+1}{2} \\ 0 & \text{otherwise} \end{cases} \end{bmatrix}. \tag{2.34}$$

This method has one coupled stage, but with stage reducibility (section 2.4.3), and all other stages are decoupled. Further, it is internally inconsistent and is scalar L- and algebraically stable for all M. A decoupled counterpart is given by the following coupling coefficients:

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \begin{cases} 0 & \lambda \leq \frac{M}{2} \\ 1 & \text{otherwise} \end{cases} \end{bmatrix}, \qquad A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} \begin{cases} 1 & \lambda \leq \frac{M}{2} \\ 0 & \text{otherwise} \end{cases} \end{bmatrix}. \tag{2.35}$$

This method is internally inconsistent, has no second order coupling error when M is even, and is scalar L- and algebraically stable for all M.

We note this method is closely connected to the following subcycled Strang splitting [152]:

$$\varphi_H^f = \left(\varphi_h^{f^{\{\mathfrak{f}\}}}\right)^{M/2} \circ \varphi_H^{f^{\{\mathfrak{s}\}}} \circ \left(\varphi_h^{f^{\{\mathfrak{f}\}}}\right)^{M/2} + \mathcal{O}\big(H^2\big).$$

Here, the operator φ_t^g maps an initial condition for the ODE $y' = g(y)$ to the solution at time t. If we approximate these exact ODE solutions with one step of the backward Euler method, we recover the decoupled multirate backward Euler scheme (2.35).

2.5.2 Second Order

The simplest second order base method is the one stage implicit midpoint method:

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}.$$

The standard MrGARK coupling coefficients

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} 0 & \lambda < L \\ \frac{1}{2} & \lambda = L \\ 1 & \lambda > L \end{bmatrix}, \qquad A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 1 & \lambda < L \\ \frac{1}{2} & \lambda = L \\ 0 & \lambda > L \end{bmatrix},$$

for odd M and $L = \frac{M+1}{2}$ give a coupled multirate midpoint method. Similar to the coupled backward Euler method (2.34), one stage is coupled but with stage reducibility, and all other stages are decoupled. Reusing the coupling coefficients (2.35) with even M and the midpoint method as the base, we derive a decoupled multirate midpoint method. Notably, both schemes maintain the algebraic stability, symmetry, and symplecticity [162] of the midpoint method. With only odd order terms appearing in the error expansion, they can be used to build efficient multirate extrapolation methods.

We also consider the L-stable, order two SDIRK base method from [4]

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \\ \hline & \frac{3}{5} & \frac{2}{5} \end{array}, \tag{2.36}$$

with $\gamma = 1 - 1/\sqrt{2}$. For this base method, an internally consistent standard MrGARK method must have at least one coupled stage. Enforcing stiff accuracy for both partitions uniquely determines a lightly coupled method:

$$\begin{aligned} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \begin{bmatrix} \frac{\lambda-1+\gamma}{M} & 0 \\ \begin{cases} 1-\gamma & \lambda = M \\ \frac{\lambda}{M} & \text{otherwise} \end{cases} & \begin{cases} \gamma & \lambda = M \\ 0 & \text{otherwise} \end{cases} \end{bmatrix}, \\ A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} &= \begin{bmatrix} \begin{cases} M\gamma & \lambda = 1 \\ 0 & \text{otherwise} \end{cases} & 0 \\ 1-\gamma & \gamma \end{bmatrix}. \end{aligned} \tag{2.37}$$

For this method, the first slow and fast stages are coupled, but with low rank structure. The last slow and fast stages are also coupled, but with stage reducibility. All other stages are decoupled. Another coupling strategy is that of Kværnø and Rentrop [66, 94] in which the first micro-step and the macro-step are computed together. The following coupling coefficients take this approach and also enforce (2.14):

$$\begin{aligned} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \begin{bmatrix} \frac{\gamma(2\lambda-1)}{M} & \begin{cases} 0 & \lambda = 1 \\ \frac{(\lambda-1)(1-2\gamma)}{M} & \lambda > 1 \end{cases} \\ \frac{1-3\gamma+2\gamma\lambda}{M} & \frac{3\gamma-1+(1-2\gamma)\lambda}{M} \end{bmatrix}, \\ A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} &= \begin{cases} M \begin{bmatrix} \gamma & 0 \\ 1-\gamma & \gamma \end{bmatrix} & \lambda = 1 \\ 0_{2\times 2} & \text{otherwise} \end{cases}. \end{aligned} \tag{2.38}$$

Here, the first two fast and slow stages are coupled, but with low rank structure.

An interesting feature of these two coupled methods is their scalar linear stability functions coincide for all M. Unfortunately, instabilities appear near the origin as M increases. At $M = 2$, the methods are only scalar $L(69.2°)$-stable (as defined in proposition 2.3), and by $M = 6$ they are not even scalar $L(0°)$-stable. While (2.37) and (2.38) may be effective for some problems, we cannot recommend them as general-purpose multirate methods. This is a surprising result as these seemingly reasonable coupling structures lead to methods with worse stability and a more expensive implementation than the decoupled multirate midpoint method. As is the case with order one, it appears that internal consistency negatively affects the stability.

The following compound-fast method, which we will call compound-fast MrGARK SDIRK2, can be derived from stability condition (2.27) and internal consistency:

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \frac{-\gamma((M-2)\gamma+3)+(2\gamma-1)\lambda+1}{M(\gamma-1)} & \frac{\gamma((M-1)\gamma-\lambda+1)}{M(\gamma-1)} \\ \frac{M\gamma^2-2\lambda\gamma+\lambda}{M-M\gamma} & \frac{\gamma(M\gamma-\lambda)}{M(\gamma-1)} \end{bmatrix}. \tag{2.39}$$

The angles of L(α)-stability for several values of M are listed in table 2.1. Unlike the aforementioned internally consistent methods of order two, compound-fast MrGARK SDIRK2 maintains a wide angle when M is large.

2.5.3 Higher Order Methods

Following the results at order two, we focus our search for implicit multirate GARK methods of orders three and four on compound-fast methods. Stiff accuracy, L-stability of the base method, and (2.27) are enforced to ensure acceptable stability properties. Internal consistency drastically reduces the number of order conditions at these higher orders and allows us to use the simplified conditions in (2.9). Finally, coupling coefficients are derived such as to be bounded functions of λ and M. Without this, methods are susceptible to catastrophic cancellation when M is large. In appendices A.1 and A.2, we have listed the coefficients of the third and fourth order methods we derived with the aforementioned constraints.

Despite the stability issues observed at second order, we also consider a third order implicit multirate method with Kværnø–Rentrop coupling using the following algebraically stable base method from [101]:

$$\begin{array}{c|cc} \gamma & \gamma & \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & 1/2 & 1/2 \end{array}, \qquad \gamma = \frac{3+\sqrt{3}}{6}. \tag{2.40}$$

Following the approach in [66, 94], the slow to fast coupling is chosen to be

$$\begin{aligned} A^{\{\mathfrak{f},\mathfrak{s},\lambda+1\}} &= \frac{1}{M}\left(A^{\{\mathfrak{f},\mathfrak{s}\}} + F(\lambda)\right), \\ F(\lambda) &= \mathbb{1}_{s^{\{\mathfrak{f}\}}}\begin{bmatrix}\eta_1(\lambda) & \dots & \eta_{s^{\{\mathfrak{s}\}}}(\lambda)\end{bmatrix}, \qquad \lambda = 0,\dots,M-1, \end{aligned}$$

where the η_j satisfy $\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \eta_j(\lambda) = \lambda$. This results in the internal consistency condition reducing to

$$A^{\{\mathfrak{f},\mathfrak{s}\}} \mathbb{1}_{s^{\{\mathfrak{s}\}}} = c, \tag{2.42}$$

and the third order coupling condition becoming

$$\frac{M}{6} = b^T \left(A^{\{\mathfrak{f},\mathfrak{s}\}} + \frac{1}{M} \sum_{\lambda=1}^{M} F(\lambda) \right) c. \tag{2.43}$$

At third order, this approach creates coefficients that grow unbounded with M. Moreover, we were unable to find an A(0°)-stable method satisfying the constraints (2.42) and (2.43).

2.5.4 Scalar Stability of New Compound-Fast Methods

The scalar stability of compound-fast methods (2.39), (A.2) and (A.4) are summarized in table 2.1. In all cases, the methods are just a few degrees short of scalar L-stability. As M increases, the stability angles decrease by less than 2° before stabilizing.

Compound-fast method	$M = 2$	$M = 3$	$M = 4$	$M = 8$	$M = 16$	$M = 32$
SDIRK2 from (2.39)	84.6°	83.5°	83.2°	83.0°	83.0°	83.0°
SDIRK3 from (A.2)	88.6°	87.8°	87.3°	86.9°	86.8°	86.8°
SDIRK4 from (A.4)	81.7°	81.2°	81.2°	81.2°	81.2°	81.2°

Table 2.1: Scalar $L(\alpha)$-stability (as defined in proposition 2.3) for new compound-fast Mr-GARK methods.

2.6 Numerical Experiments

In this section, we use the new methods to integrate two test problems. First, the CUSP model is used to verify the order of accuracy. Next, the inverter chain model is used to compare the performance of multirate methods against single rate and implicit-explicit (IMEX) methods.

2.6.1 CUSP Model

The CUSP model, as reported in [72, Chapter IV.10], is a reaction-diffusion model defined with the equations

$$\begin{aligned}
\frac{\partial y}{\partial t} &= -\frac{1}{\varepsilon}\left(y^3 + a\,y + b\right) + \sigma\,\frac{\partial^2 y}{\partial x^2},\\
\frac{\partial a}{\partial t} &= b + 0.07\,v + \sigma\,\frac{\partial^2 a}{\partial x^2},\\
\frac{\partial b}{\partial t} &= b\,(1-a^2) - a - 0.4\,y + 0.035\,v + \sigma\,\frac{\partial^2 b}{\partial x^2},
\end{aligned} \tag{2.44}$$

where $v = \frac{u}{u+0.1}$ and $u = (y-0.7)\,(y-1.3)$. The parameters are $\sigma = \frac{1}{144}$ and $\varepsilon = 10^{-4}$, which makes the problem stiff. Equation (2.44) is integrated from $t = 0$ to $t = 1.1$ over the spatial domain $x \in [0, 1]$. In our numerical experiments, we use second order central finite differences on a uniform mesh with $N = 32$ points and periodic boundary conditions. The initial conditions are

$$y_i(0) = 0, \quad a_i(0) = -2\cos\left(\frac{2\pi i}{N}\right), \quad b_i(0) = 2\sin\left(\frac{2\pi i}{N}\right), \quad \text{for} \quad i = 1, \ldots, N.$$

The splitting of the right-hand side function is done over the physics: diffusion is considered as the slow function, and the remaining reactive terms are the fast function. The MATLAB implementation of the CUSP problem is available in [119]. We use MATLAB's `ode15s` to compute a high-accuracy reference solution with absolute and relative tolerances set to 10^{-13}. Error is measured as the 2-norm of the difference of the numerical solution and this reference solution at time $t = 1.1$.

Figure 2.2 shows convergence results for the decoupled multirate midpoint method and compound-fast MrGARK SDIRK methods of orders two, three, and four using a range of multirate ratios. For compound-fast MrGARK SDIRK4, the numerical rate of convergence is slightly higher than the nominal order. In all other cases, the numerical orders closely match the theoretical ones. We also note that for a fixed number of steps, the error decreases as the multirate ratio increases as expected.

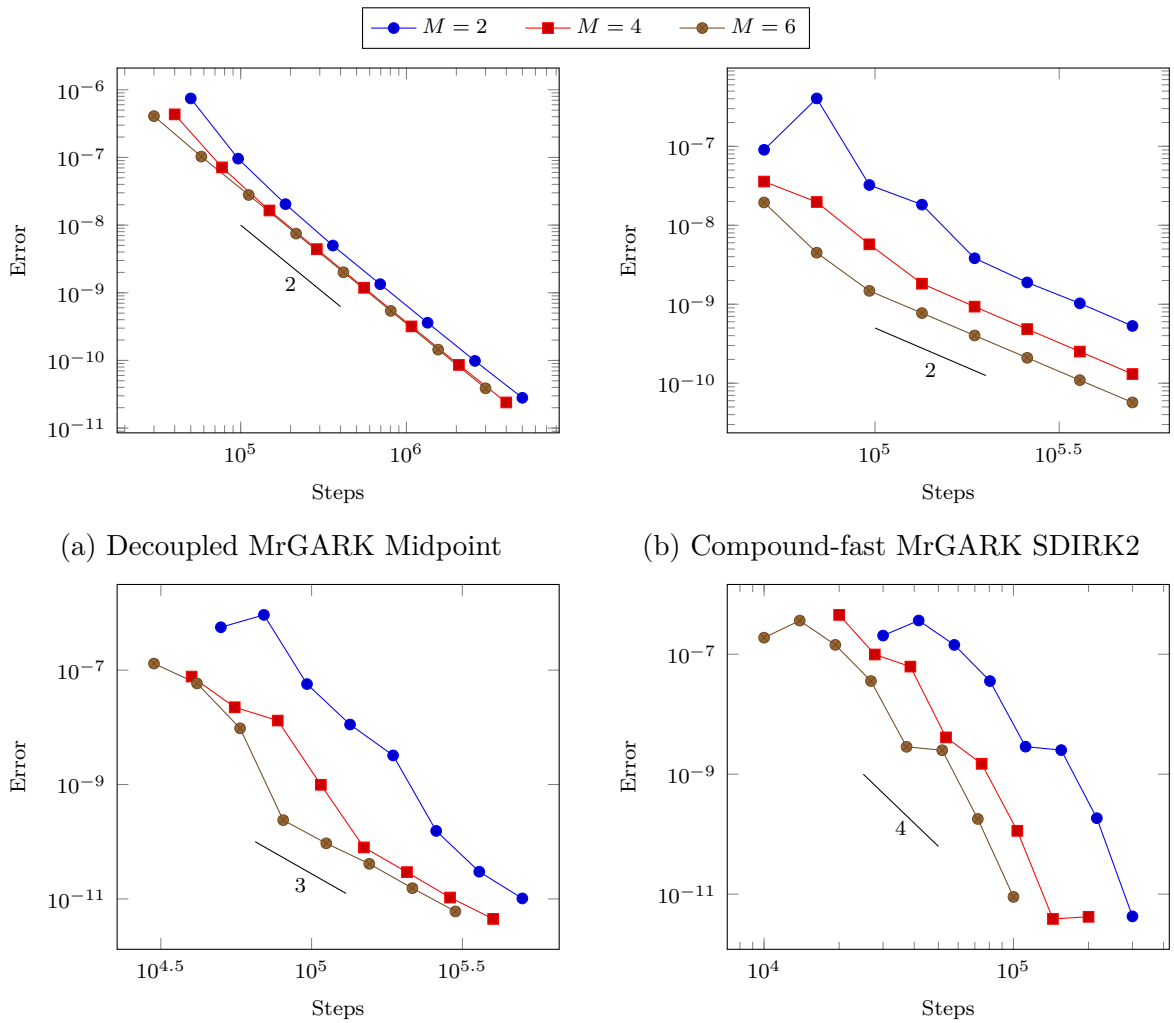
2.6.2 Inverter Chain Model

We also consider the inverter chain model of [16, 94] given by the equations

$$\begin{aligned}
U_1' &= U_{op} - U_1 - \Gamma\,g(U_{in}, U_1, U_0),\\
U_i' &= U_{op} - U_i - \Gamma\,g(U_{i-1}, U_i, U_0), \qquad i = 2, \ldots m,
\end{aligned} \tag{2.45}$$

with

$$g(U_g, U_D, U_S) = (\max(U_G - U_S - U_T, 0))^2 - (\max(U_G - U_D - U_T, 0))^2.$$



(a) Decoupled MrGARK Midpoint

(b) Compound-fast MrGARK SDIRK2

(c) Compound-fast MrGARK SDIRK3

(d) Compound-fast MrGARK SDIRK4

Figure 2.2: Error vs. number of macro-steps for the decoupled midpoint method (2.35) and compound-fast methods (2.39), (A.2) and (A.4) applied to the CUSP problem (2.44). Reference slopes are included to compare with the numerical orders.

It models the propagation of the input signal

$$U_{in}(t) = \begin{cases} t - 5 & 5 \leq t \leq 10 \\ 5 & 10 \leq t \leq 15 \\ \frac{5}{2}(17 - t) & 15 \leq t \leq 17 \\ 0 & \text{otherwise} \end{cases}$$

through a sequence of m metal-oxide-semiconductor field-effect transistor (MOSFET) inverters. The ground voltage is $U_0 = 0$, the operating voltage is $U_{op} = 5$, and the threshold

voltage separating the on and off states is $U_T = 1$. Stiffness is controlled by Γ and is taken to be 100 as in the stiff case used in [16]. The initial conditions of the system are

$$U_i(0) = \begin{cases} 6.246 \times 10^{-3} & i \text{ even} \\ 5 & i \text{ odd} \end{cases}.$$

For the numerical experiments, we use $m = 500$ inverters and a timespan of $[0, 120]$ to allow the signal to reach the end of the chain.

In this numerical experiment, we compare the performance of three types of methods on the inverter chain: single rate, IMEX Runge–Kutta, and compound-fast MrGARK. While compound-fast MrGARK can use dynamic partitioning to select the fast inverters as described in section 2.2.2, there is not a direct analog for IMEX Runge–Kutta methods. For this reason, we use a fixed, time-dependent partitioning that follows the propagation of the signal though the chain. Inverters with indices in the range

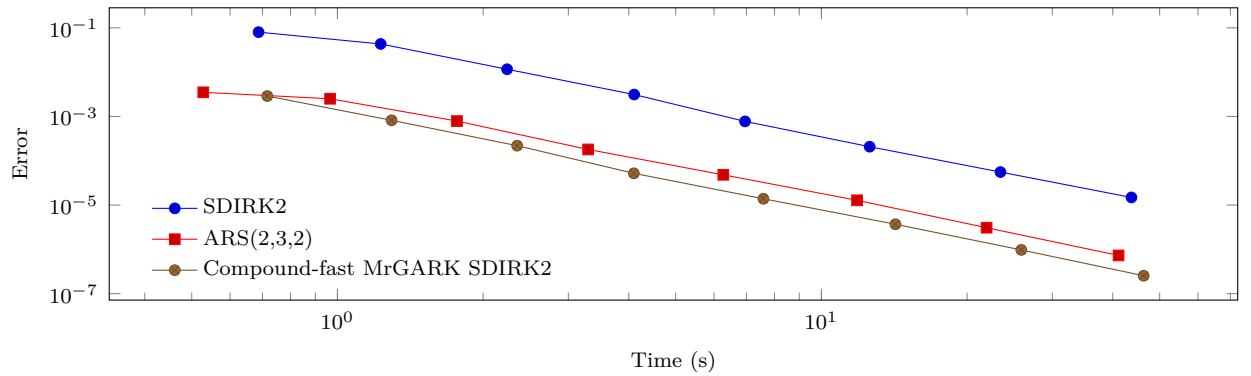
$$[\min(\max(1, \lfloor 4.75t - 95 \rfloor), m + 1), \min(\max(0, \lfloor 4.75t - 15 \rfloor), m)]$$

form the fast partition. Only these inverters are treated with a microstep by multirate schemes and implicitly by IMEX schemes.

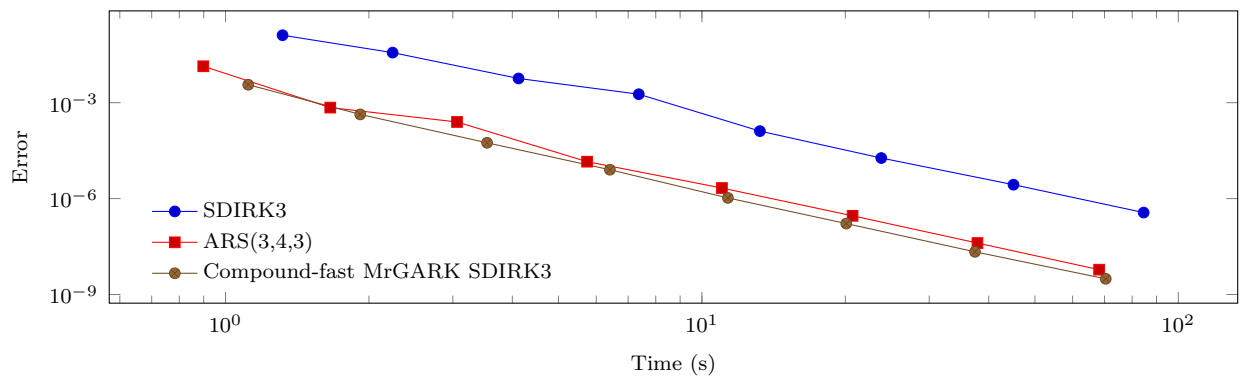
At second order, we consider the performance of compound-fast MrGARK SDIRK2 from (2.39). The primary baseline is its base method: SDIRK2 from (2.36). SDIRK2 is a traditional Runge–Kutta method and treats all inverters with the same timestep. We also test the IMEX Runge–Kutta scheme ARS(2,3,2) from [14, Section 2.5]. Note that the implicit part of ARS(2,3,2) is SDIRK2, which makes for a fair comparison among all three second order methods. At third order, we use compound-fast MrGARK SDIRK3 from (A.2), its base method SDIRK3 from (A.1), and ARS(3,4,3) from [14, Section 2.7]. The implicit part of ARS(3,4,3) is SDIRK3. At fourth order, we use compound-fast MrGARK SDIRK4 from (A.4). SDIRK4 from [72, pg. 100] is slightly more optimized than (A.3), so we use that for the traditional Runge–Kutta baseline. Finally, ARK4(3)6L[2]SA from [85] is used as the fourth order IMEX scheme. The multirate ratios we use are $M = 14, 10, 6$ for orders two, three, and four, respectively.

A serial C implementation of the inverter chain and integrators was run on the Cascades cluster managed by Advanced Research Computing (ARC) at Virginia Tech. In the experiment, the error and runtime were recorded for a range of eight stepsizes for all nine methods. Error is computed in the infinity-norm with respect to a high-accuracy reference solution. Figure 2.3 plots the timing results. At orders two and three, we can see the compound-fast MrGARK methods reach a fixed accuracy four to six times faster than the single rate methods and are slightly more efficient than the IMEX methods. The fourth order multirate and IMEX methods have similar performance and are approximately three times faster than the single rate method.

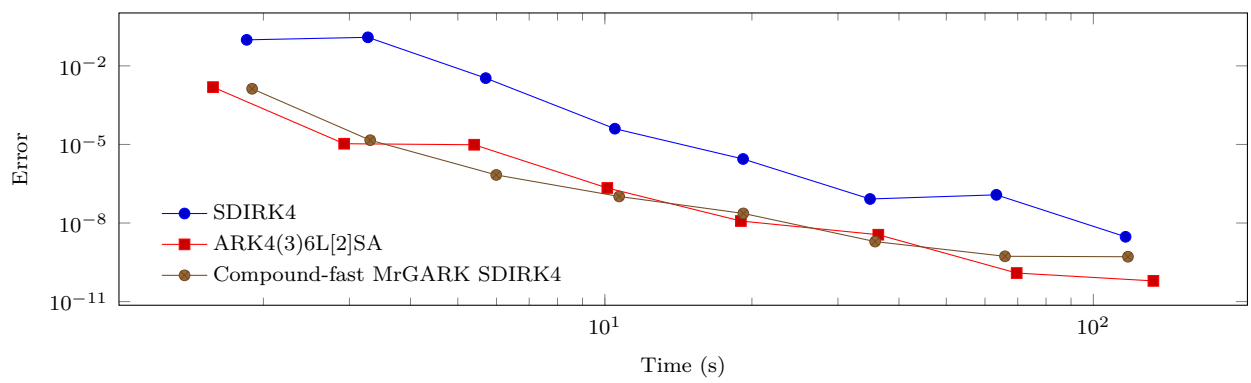
Despite the similar performance of the IMEX and multirate methods, the number of steps required to reach a desired accuracy is very different. The stiffness of the inverter chain prob-



(a) Second order



(b) Third order



(c) Fourth order

Figure 2.3: Error vs. time for single rate, IMEX, and compound-fast methods applied to the inverter chain problem (2.45).

lem, even in the slow partition, forces the IMEX schemes to take relatively small timesteps. Table 2.2 lists the largest timestep each of the tested methods could take. For comparison, we have also included single rate explicit methods. In particular, we use Ralston's optimal second and third order methods [114] and the classical fourth order Runge–Kutta method.

Note that the IMEX methods have an explicit-like stepsize restriction for this problem. The compound-fast MrGARK methods have the same maximum stepsize as the implicit single rate methods, which indicates stiffness in the slow partition is likely limiting the stepsize.

	Single rate implicit	Single rate explicit	IMEX	Compound-fast
Order 2	7.1×10^{-2}	3.1×10^{-3}	3.2×10^{-3}	7.1×10^{-2}
Order 3	5.2×10^{-2}	3.2×10^{-3}	3.5×10^{-3}	5.2×10^{-2}
Order 4	6.0×10^{-2}	3.4×10^{-3}	5.2×10^{-3}	6.0×10^{-2}

Table 2.2: Approximate largest stepsizes to ensure stability and convergence of Newton iterations for the inverter chain problem (2.45).

2.7 Conclusions

In this work, we have explored multirate Runge–Kutta methods in which all time-scales are treated implicitly. By taking different timesteps for different partitions of an ODE, these methods can more efficiently integrate stiff, multiscale problems. We have examined their order conditions, their linear stability, and techniques for solving implicit stage equations. In appendix B, we also have added a short note on conservation of linear invariants.

Compared to single rate methods, the linear stability for multirate methods is much more intricate. It not only depends on the base methods and coupling structure but also the choice of test problem. The scalar and 2D test problems present a trade-off of generality versus simplicity to analyze. The theoretical limitations and observed degradation of multirate stability often come from problems that are oscillatory. These problems are challenging because the error introduced by the coupling is not damped by any partition. In addition, we found that forgoing internal consistency can improve stability but increases the number of order conditions and limits the stage order to zero.

The coupling structure of MrGARK methods has a significant effect on the computational cost of the Newton iterations. Decoupled methods are the cheapest and simplest to implement, especially for component partitioned problems. Coupled methods have the potential to become prohibitively expensive but can be implemented efficiently by exploiting stage reducibility or low rank structure in the method.

The GARK framework provides new insight into the compound-fast methods. Instead of taking the approach of finding a dense output formula for coupling, we use the precise GARK order conditions. This approach facilitated the development of methods up to order four, which to our knowledge, is the highest of this type. Stability depends heavily on this coupling, so we derived a practical and general form for the scalar stability function. By taking the limit as the partitions become infinitely stiff, we found a simple condition to ensure L(α)-stability.

New standard MrGARK methods based on backward Euler and the midpoint method show excellent stability properties. For base methods with more than one stage, however, we were unable to find methods with satisfactory stability. Extrapolation may be the most practical way to achieve high-order, but this warrants additional investigation.

2.8 Disclaimer

The work of S. Roberts (in part), J. Loffeld, and C.S Woodward was supported by the Lawrence Livermore Laboratory Directed Research and Development Program under tracking number 17-ERD-035. Their work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC. LLNL-JRNL-795454.

Chapter 3

Coupled Multirate Infinitesimal GARK Schemes for Stiff Systems with Multiple Time Scales

First Published in SIAM Journal on Scientific Computing in Volume 42, Issue 3, published by the Society for Industrial and Applied Mathematics (SIAM) [121].

3.1 Introduction

In this paper, we consider the additively partitioned ordinary differential equation (ODE)

$$y' = f(t, y) = f^{\{\mathfrak{f}\}}(t, y) + f^{\{\mathfrak{s}\}}(t, y), \qquad y(t_0) = y_0 \in \mathbb{R}^d, \tag{3.1}$$

where component $f^{\{\mathfrak{f}\}}$ represents the fast dynamics of the system, and component $f^{\{\mathfrak{s}\}}$ the slow dynamics. This structure models a feature appearing in many dynamical systems of practical interest: multiple characteristic time scales.

Multirate time integration methods are designed to efficiently solve (3.1) by using different timesteps for the fast and slow components. First explored by Rice [118] and Andrus [9, 10], the multirating strategy has been expanded to numerous types of traditional time integration methods. This includes Runge–Kutta methods [41, 65, 67, 93, 94, 123, 135], linear multistep methods [61, 84, 130], Rosenbrock-W methods [68], extrapolation methods [43, 51], Galerkin discretizations [97], and combined multiscale methodologies [50].

Multirate infinitesimal step (MIS) methods, first proposed by Knoth and Wolke [92], and later extended by others [91, 141, 142, 143, 161], introduce a new multirating philosophy in which the fast method solves a modified ODE that advances the solution between slow stages. While the slow system is solved discretely, the fast system can be solved with arbitrary small steps, hence the naming "infinitesimal step". In [66], Günther and Sandu cast MIS methods into the General-structure Additive Runge–Kutta (GARK) framework. This framework was subsequently leveraged by Sandu in [129] to create the multirate infinitesimal GARK (MRI-GARK) class of methods. One step of an MRI-GARK method advances the solution from

t_n to $t_n + H$ by

$$Y_1 = y_n \tag{3.2a}$$

$$\begin{cases} v_i(0) = Y_i, \\ \quad T_i = t_n + c_i^{\{\mathfrak{s}\}} H, \\ \quad v_i' = \Delta c_i^{\{\mathfrak{s}\}} f^{\{\mathfrak{f}\}}\Big(T_i + \Delta c_i^{\{\mathfrak{s}\}} \theta, v_i\Big) + \displaystyle\sum_{j=1}^{i+1} \gamma_{i,j}\big(\tfrac{\theta}{H}\big) f^{\{\mathfrak{s}\}}(T_j, Y_j), \\ \qquad \text{for } \theta \in [0, H], \\ Y_{i+1} = v_i(H), \qquad i = 1, \ldots, s^{\{\mathfrak{s}\}}, \end{cases} \tag{3.2b}$$

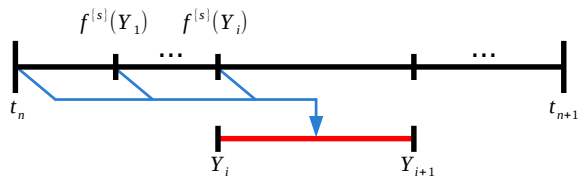
$$y_{n+1} = Y_{s^{\{\mathfrak{s}\}}+1}, \tag{3.2c}$$

where $c^{\{\mathfrak{s}\}}$ are the slow method abscissae, and the modified fast ODEs $v_i' = \ldots$ advance the solution between the slow stages. In [129], Sandu presents MRI-GARK methods (3.2) of orders up to four that are explicit or implicit in the fast and slow systems but are not coupled across partitions. This work also provides new techniques in investigating the stability of partitioned methods that we have adopted in our paper. Recent developments in the field include the work of Sexton and Reynolds [146] where a new structure for fast integration weights is considered and shown to help reduce order conditions; the "relaxed MIS" methods derived retain the same order as traditional MIS and it is possible to pair them for error control and adaptivity purposes.

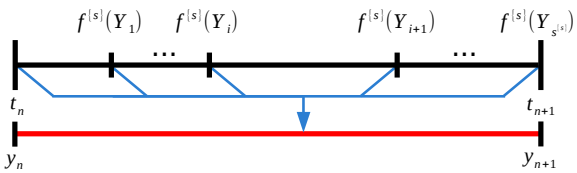
The paper is organized as follows. The new family of step predictor corrector MRI-GARK schemes is introduced in section 3.2, followed by its order condition theory and the stability analysis. In section 3.3, internal stage predictor corrector MRI-GARK schemes are defined and their order conditions and stability are established. Numerical results are reported in section 3.4, and concluding remarks are drawn in section 3.5. Appendix C presents the lists of coefficients and stability plots for the newly developed methods.

3.2 Step Predictor-Corrector MRI-GARK Methods

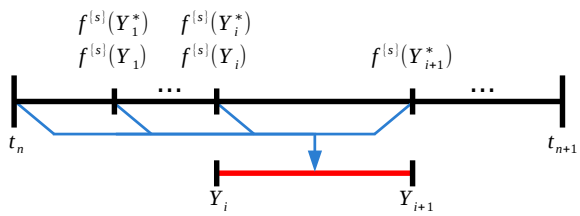
One coupling strategy commonly used in discrete multirate methods is a predictor-corrector approach, where the predictor evolves the entire system, while the corrector is only applied to the fast partition whose solution was "predicted" inaccurately (see [118, 137, 139]). First, a combined Runge–Kutta macro-step is taken which serves as the predictor. The fast parts of the predicted stages are inaccurate and are refined by sub-stepping the fast component only. Approximations of the slow values needed during the micro-steps are obtained from interpolating the slow predicted values. The step predictor-corrector MRI-GARK methods, as depicted in fig. 3.1b, can be viewed as an extreme case of this coupling strategy where the multirate ratio is infinite, i.e., the corrector takes infinitely many steps to refine the fast solution.



(a) Traditional multirate infinitesimal schemes use previously computed stages, which means the slow tendencies are extrapolated in the formulation of modified fast systems.



(b) The newly proposed step predictor-corrector MRI-GARK schemes use all stages for computing slow tendencies and solve a single fast ODE over the entire step.



(c) The newly proposed internal stage predictor-corrector MRI-GARK schemes use previously computed stages and a predicted next stage, which allows for interpolation of slow tendencies in the formulation of modified fast systems.

Figure 3.1: Comparison of MRI-GARK schemes: blue arrows indicate stage dependencies of the modified fast ODE and the red lines indicate the intervals over which a fast ODE is solved.

3.2.1 Method Definition

We start with a "slow" Runge–Kutta base method

$$\begin{array}{c|c} c^{\{\mathfrak{s}\}} & \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline & b^{\{\mathfrak{s}\}T} \\ \hline & \widehat{b}^{\{\mathfrak{s}\}T} \end{array} \tag{3.3}$$

with $s^{\{\mathfrak{s}\}}$ stages. Unlike other multirate infinitesimal strategies, the base method is not restricted to be explicit or diagonally implicit.

Definition 3.1 (Step predictor corrector MRI-GARK methods)**.** One step of a step predictor-corrector MRI-GARK (SPC-MRI-GARK) scheme applied to (3.1) is given by

$$Y_i = y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s}\}} f_j, \quad i = 1, \dots, s^{\{\mathfrak{s}\}}, \tag{3.4a}$$

$$\begin{cases} v(0) = y_n, \\ \quad v' = f^{\{\mathfrak{f}\}}(t_n + \theta, v) + \displaystyle\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \gamma_j\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}}, \quad \text{for } \theta \in [0, H], \\ y_{n+1} = v(H), \end{cases} \tag{3.4b}$$

where $f_j \coloneqq f\big(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j\big)$ and $f_j^{\{\mathfrak{s}\}} \coloneqq f^{\{\mathfrak{s}\}}\big(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j\big)$.

Definition 3.2 (Slow tendency coefficients [129, Definition 2.2])**.** The time-dependent coefficients in (3.4b) are defined as polynomials:

$$\gamma_i(t) \coloneqq \sum_{k\geq 0} \gamma_i^k t^k, \quad \widetilde{\gamma}_i(t) \coloneqq \int_0^t \gamma_i(\tau)\,\mathrm{d}\tau = \sum_{k\geq 0} \gamma_i^k \frac{t^{k+1}}{k+1}, \quad \overline{\gamma}_i \coloneqq \widetilde{\gamma}_i(1). \tag{3.5}$$

Remark 3.3 (Embedded method)**.** An embedded solution for an SPC-MRI-GARK method can be computed by solving the additional ODE

$$\begin{cases} \widehat{v}(0) = y_n, \\ \quad \widehat{v}' = f^{\{\mathfrak{f}\}}(t_n + \theta, \widehat{v}) + \displaystyle\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \widehat{\gamma}_j\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}}, \quad \text{for } \theta \in [0, H], \\ \widehat{y}_{n+1} = \widehat{v}(H), \end{cases}$$

which uses the embedded polynomials $\widehat{\gamma}_i$ and produces a solution of a different order. We note that although this additional integration can be expensive, it can be done in parallel with (3.4b).

Consider the trivial partitioning $f^{\{\mathfrak{f}\}} = 0$, $f^{\{\mathfrak{s}\}} = f$ of (3.1). In this case, it is natural to expect an SPC-MRI-GARK method to degenerate into the slow base method. Note that the final solution of (3.4b) simplifies to

$$y_{n+1} = y_n + \int_0^H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \gamma_j\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}} \,\mathrm{d}\theta = y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \overline{\gamma}_j \, f_j^{\{\mathfrak{s}\}}. \tag{3.6}$$

Thus, we enforce the condition

$$\mathfrak{b}^{\{\mathfrak{s}\}} = \overline{\gamma}. \tag{3.7}$$

An important special case of (3.1) are component partitioned systems:

$$\begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}' = \begin{bmatrix} f^{\{\mathfrak{f}\}}\left(t, y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right) \\ f^{\{\mathfrak{s}\}}\left(t, y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right) \end{bmatrix} = \begin{bmatrix} f^{\{\mathfrak{f}\}}\left(t, y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ f^{\{\mathfrak{s}\}}\left(t, y^{\{\mathfrak{f}\}}, y^{\{\mathfrak{s}\}}\right) \end{bmatrix}. \tag{3.8}$$

One step of an SPC-MRI-GARK method (3.4) applied to (3.8) reads:

$$\begin{bmatrix} Y_i^{\{\mathfrak{f}\}} \\ Y_i^{\{\mathfrak{s}\}} \end{bmatrix} = \begin{bmatrix} y_n^{\{\mathfrak{f}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s}\}} \, f_j^{\{\mathfrak{f}\}} \\ y_n^{\{\mathfrak{s}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s}\}} \, f_j^{\{\mathfrak{s}\}} \end{bmatrix}, \tag{3.9a}$$

$$\begin{cases} v^{\{\mathfrak{f}\}}(0) = y_n^{\{\mathfrak{f}\}}, \\ \quad v^{\{\mathfrak{f}\}\prime} = f^{\{\mathfrak{f}\}}\left(t_n + \theta, v, y_n^{\{\mathfrak{s}\}} + H \displaystyle\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \widetilde{\gamma}_j\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}}\right), \quad \text{for } \theta \in [0, H] \\ \begin{bmatrix} y_{n+1}^{\{\mathfrak{f}\}} \\ y_{n+1}^{\{\mathfrak{s}\}} \end{bmatrix} = \begin{bmatrix} v^{\{\mathfrak{f}\}}(H) \\ y_n^{\{\mathfrak{s}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \mathfrak{b}_j^{\{\mathfrak{s}\}} \, f_j^{\{\mathfrak{s}\}} \end{bmatrix}, \end{cases} \tag{3.9b}$$

where $f_j^{\{\mathfrak{f}\}} := f^{\{\mathfrak{f}\}}\left(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j^{\{\mathfrak{f}\}}, Y_j^{\{\mathfrak{s}\}}\right)$ and $f_j^{\{\mathfrak{s}\}} := f^{\{\mathfrak{s}\}}\left(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j^{\{\mathfrak{f}\}}, Y_j^{\{\mathfrak{s}\}}\right)$. With (3.6) and (3.7) the internal ODE integrates (and corrects) only the fast component, while the slow component is solved with the traditional base Runge–Kutta method (3.3).

3.2.2 Order Conditions

Following [129], we use an arbitrarily accurate Runge–Kutta method $\left(A^{\{\mathfrak{f},\mathfrak{f}\}}, b^{\{\mathfrak{f}\}}, c^{\{\mathfrak{f}\}}\right)$ to discretize the continuous ODE appearing in the method formulation, which casts the SPC-MRI-GARK scheme (3.4) into the GARK framework. The discrete corrector stages, denoted

$Y_i^{\{\mathfrak{f},\mathfrak{c}\}}$, are computed as

$$\begin{aligned}
Y_i^{\{\mathfrak{f},\mathfrak{c}\}} &= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}} \left(f_j^{\{\mathfrak{f},\mathfrak{c}\}} + \sum_{\ell=1}^{s^{\{\mathfrak{s}\}}} \gamma_\ell\big(c_j^{\{\mathfrak{f}\}}\big)\, f_\ell^{\{\mathfrak{s}\}} \right), \\
&= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}}\, f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{\ell=1}^{s^{\{\mathfrak{s}\}}} \left(\sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}}\, \gamma_\ell\big(c_j^{\{\mathfrak{f}\}}\big) \right) f_\ell^{\{\mathfrak{s}\}}, \\
&= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}}\, f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \left(\sum_{k\geq 0} A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k}\, \gamma^{k^T} \right)_{i,j} f_j^{\{\mathfrak{s}\}},
\end{aligned}$$

where $f_j^{\{\mathfrak{f},\mathfrak{c}\}} := f^{\{\mathfrak{f}\}}\big(t_n + c_j^{\{\mathfrak{f}\}}\, H, Y_j^{\{\mathfrak{f},\mathfrak{c}\}}\big)$ and the superscript $\times k$ denotes the elementwise vector power. Similarly, the final solution reads

$$\begin{aligned}
y_{n+1} &= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}}\, f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \left(\sum_{k\geq 0} b^{\{\mathfrak{f}\}T}\, c^{\{\mathfrak{f}\}\times k} \right) f_j^{\{\mathfrak{s}\}}, \\
&= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}}\, f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \left(\sum_{k\geq 0} \frac{1}{k+1} \gamma_j^k \right) f_j^{\{\mathfrak{s}\}}, \\
&= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \overline{\gamma}_j\, f_j^{\{\mathfrak{s}\}}, \\
&= y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}}\, f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \mathscr{b}_j^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{s}\}}.
\end{aligned}$$

Now, the corresponding GARK tableau for an SPC-MRI-GARK method is

$$\begin{array}{c|cc|c|c}
\mathscr{c}^{\{\mathfrak{s}\}} & \mathscr{A}^{\{\mathfrak{s},\mathfrak{s}\}} & 0 & \mathscr{A}^{\{\mathfrak{s},\mathfrak{s}\}} & \mathscr{c}^{\{\mathfrak{s}\}} \\
c^{\{\mathfrak{f}\}} & 0 & A^{\{\mathfrak{f},\mathfrak{f}\}} & \sum_{k\geq 0} A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k}\, \gamma^{k^T} & c^{\{\mathfrak{f},\mathfrak{s}\}} \\
\hline
\mathscr{c}^{\{\mathfrak{s}\}} & \mathscr{A}^{\{\mathfrak{s},\mathfrak{s}\}} & 0 & \mathscr{A}^{\{\mathfrak{s},\mathfrak{s}\}} & \mathscr{c}^{\{\mathfrak{s}\}} \\
\hline
 & 0 & b^{\{\mathfrak{f}\}T} & \mathscr{b}^{\{\mathfrak{s}\}T} &
\end{array},$$

with $c^{\{\mathfrak{f},\mathfrak{s}\}} = \sum_{k\geq 0} A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k}\, \gamma^{k^T}\, \mathbb{1}^{\{\mathfrak{s}\}}$.

Internal consistency

Theorem 3.4 (Internal consistency conditions)**.** *An SPC-MRI-GARK method* (3.4) *satisfies the "internal consistency" conditions*

$$\mathbf{c}^{\{\mathfrak{s},\mathfrak{f}\}} = \mathbf{c}^{\{\mathfrak{s},\mathfrak{s}\}} \equiv \mathscr{c}^{\{\mathfrak{s}\}} \quad \textit{and} \quad \mathbf{c}^{\{\mathfrak{f},\mathfrak{f}\}} = \mathbf{c}^{\{\mathfrak{f},\mathfrak{s}\}}$$

for any fast method iff the following conditions hold:

$$\gamma^{0^T} \mathbb{1}^{\{s\}} = 1 \quad \textit{and} \quad \gamma^{k^T} \mathbb{1}^{\{s\}} = 0 \quad \forall k \geq 1. \tag{3.10}$$

Proof. All internal consistency equations are automatically satisfied except for the following one, which needs to be imposed explicitly:

$$c^{\{f\}} = \sum_{k \geq 0} A^{\{f,f\}} \, c^{\{f\} \times k} \, \gamma^{k^T} \mathbb{1}^{\{s\}}.$$

It is easy to confirm (3.10) is sufficient to satisfy this condition, and thus, internal consistency. Since the equality must hold for all $A^{\{f,f\}}$, it must hold when all $A^{\{f,f\}} \, c^{\{f\} \times k}$ are linearly independent. Matching powers of the left- and right-hand sides proves the necessity of (3.10). □

If an SPC-MRI-GARK method has a slow base method (3.3) of order two, then internal consistency is sufficient to guarantee the method is order two [132].

Fourth order conditions

In this section, we derive order conditions of the SPC-MRI-GARK schemes for up to order four. First, we define a set of useful coefficients.

Definition 3.5 (Some useful coefficients [129, Definition 3.3])**.** Consider the "bushy" Butcher tree [74]

$$\mathfrak{t}_k := \underbrace{[\tau, \ldots, \tau]}_{k \text{ times}} \in T, \tag{3.11}$$

where $\tau \in T$ is the tree of order one and $[\cdot]$ is the operation of joining subtrees by a root. An arbitrarily accurate Runge–Kutta method $\left(A^{\{f,f\}}, b^{\{f\}}, c^{\{f\}}\right)$ satisfies the following equations:

$$\begin{aligned}
\zeta_k &:= \frac{1}{\gamma([\mathfrak{t}_k])} &&= b^{\{f\}T} \, A^{\{f,f\}} \, c^{\{f\} \times k} &&= \frac{1}{(k+1)(k+2)}, \\
\omega_k &:= \frac{1}{\gamma([\tau, \mathfrak{t}_k])} &&= \left(b^{\{f\}} \times c^{\{f\}}\right)^T A^{\{f,f\}} \, c^{\{f\} \times k} &&= \frac{1}{(k+1)(k+3)}, \\
\xi_k &:= \frac{1}{\gamma([[\mathfrak{t}_k]])} &&= b^{\{f\}T} \, A^{\{f,f\}} \, A^{\{f,f\}} \, c^{\{f\} \times k} &&= \frac{1}{(k+1)(k+2)(k+3)}.
\end{aligned} \tag{3.12}$$

Theorem 3.6 (Fourth order coupling conditions)**.** *An internally consistent SPC-MRI-GARK method* (3.4) *satisfying* (3.7) *has order four iff the slow base scheme* (3.3) *has order*

at least four, and the following coupling conditions hold:

$$\frac{1}{6} = \sum_{k\geq 0} \zeta_k \, \gamma^{k^T} \, c^{\{\mathfrak{s}\}}, \qquad \textit{(order 3)} \tag{3.13a}$$

$$\frac{1}{8} = \sum_{k\geq 0} \omega_k \, \gamma^{k^T} \, c^{\{\mathfrak{s}\}}, \qquad \textit{(order 4)} \tag{3.13b}$$

$$\frac{1}{12} = \sum_{k\geq 0} \zeta_k \, \gamma^{k^T} \, c^{\{\mathfrak{s}\}\times 2}, \qquad \textit{(order 4)} \tag{3.13c}$$

$$\frac{1}{24} = \sum_{k\geq 0} \zeta_k \, \gamma^{k^T} \, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} \, c^{\{\mathfrak{s}\}}. \qquad \textit{(order 4)} \tag{3.13d}$$

Proof. An internally consistent GARK scheme is order four iff the base methods are order four and the 12 coupling conditions up to order four are satisfied [132]. We proceed with checking each coupling condition.

Condition 3a The first third order condition gives (3.13a):

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{f}\}T} \, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \, \mathbf{c}^{\{\mathfrak{s}\}} = \sum_{k\geq 0} \zeta_k \, \gamma^{k^T} \, c^{\{\mathfrak{s}\}}.$$

Condition 3b The other third order condition is automatically satisfied if the slow base method is order three:

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{s}\}T} \, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} \, \mathbf{c}^{\{\mathfrak{f}\}} = b^{\{\mathfrak{s}\}T} \, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} \, c^{\{\mathfrak{s}\}}.$$

Condition 4a The first fourth order condition gives (3.13b):

$$\frac{1}{8} = \left(\mathbf{b}^{\{\mathfrak{f}\}} \times \mathbf{c}^{\{\mathfrak{f}\}}\right)^T \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \, \mathbf{c}^{\{\mathfrak{s}\}} = \sum_{k\geq 0} \omega_k \, \gamma^{k^T} \, c^{\{\mathfrak{s}\}}.$$

Condition 4b This condition is automatically satisfied if the slow base method is order four:

$$\frac{1}{8} = \left(\mathbf{b}^{\{\mathfrak{s}\}} \times \mathbf{c}^{\{\mathfrak{s}\}}\right)^T \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} \, \mathbf{c}^{\{\mathfrak{f}\}} = \left(b^{\{\mathfrak{s}\}} \times c^{\{\mathfrak{s}\}}\right)^T \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} \, c^{\{\mathfrak{s}\}}.$$

Condition 4c This condition proves (3.13c):

$$\frac{1}{12} = \mathbf{b}^{\{\mathfrak{f}\}T} \, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \, \mathbf{c}^{\{\mathfrak{s}\}\times 2} = \sum_{k\geq 0} \zeta_k \, \gamma^{k^T} \, c^{\{\mathfrak{s}\}\times 2}.$$

Condition 4d This condition is automatically satisfied if the slow base method is order four:

$$\frac{1}{12} = \mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}\times 2} = \mathcal{b}^{\{\mathfrak{s}\}T}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{c}^{\{\mathfrak{s}\}\times 2}.$$

Condition 4e This condition is the redundant since it is the difference of Condition 3a and 4a:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \sum_{k\geq 0} \xi_k\,\gamma^{k^T}\,\mathcal{c}^{\{\mathfrak{s}\}}.$$

Condition 4f This condition proves (3.13d):

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \sum_{k\geq 0} \zeta_k\,\gamma^{k^T}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{c}^{\{\mathfrak{s}\}}.$$

Condition 4g The following condition is identical to Condition 4f:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \sum_{k\geq 0} \zeta_k\,\gamma^{k^T}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{c}^{\{\mathfrak{s}\}}.$$

Condition 4h This condition is automatically satisfied if the slow base method is order four:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \mathcal{b}^{\{\mathfrak{s}\}T}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{c}^{\{\mathfrak{s}\}}.$$

Condition 4i This condition is automatically satisfied if the slow base method is order four:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \mathcal{b}^{\{\mathfrak{s}\}T}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{c}^{\{\mathfrak{s}\}}.$$

Condition 4j This condition is automatically satisfied if the slow base method is order four:

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \mathcal{b}^{\{\mathfrak{s}\}T}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathcal{c}^{\{\mathfrak{s}\}}.$$

□

Remark 3.7. In the proof of proposition 3.6, all coupling order conditions that start with $\mathbf{b}^{\{\mathfrak{s}\}T}$ collapse onto the order conditions of the slow base method. In the context of two-trees [12, 132], trees containing both slow and fast nodes with a slow root can be recolored into purely slow trees. The purely fast trees are of no concern since the fast base method is arbitrarily accurate. The remaining trees contain slow and fast nodes with a fast root, which

correspond to coupling conditions (3.13) that must be explicitly enforced through the choice of γ.

3.2.3 Stability Analysis

Scalar Stability Analysis

Consider the partitioned, linear, scalar test problem

$$y' = \lambda^{\{\mathfrak{f}\}}\, y + \lambda^{\{\mathfrak{s}\}}\, y, \qquad \lambda^{\{\mathfrak{f}\}}, \lambda^{\{\mathfrak{s}\}} \in \mathbb{C}^-, \tag{3.14}$$

and let $z^{\{\mathfrak{f}\}} := H\,\lambda^{\{\mathfrak{f}\}}$, $z^{\{\mathfrak{s}\}} := H\,\lambda^{\{\mathfrak{s}\}}$, and $z := z^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}$. Applying the SPC-MRI-GARK method (3.4) to (3.14) yields

$$\begin{aligned}
&Y = \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\right)^{-1} \mathbb{1}^{\{\mathfrak{s}\}} y_n,\\
&\begin{cases}
v(0) = y_n,\\
\quad v' = \lambda^{\{\mathfrak{f}\}}\, v + \lambda^{\{\mathfrak{s}\}} \displaystyle\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \gamma_j\left(\tfrac{\theta}{H}\right) Y_j,\\
\qquad\quad = \lambda^{\{\mathfrak{f}\}}\, v + \lambda^{\{\mathfrak{s}\}} \displaystyle\sum_{k\geq 0} \left(\tfrac{\theta}{H}\right)^k \gamma^{k^T}\, Y,\\
y_{n+1} = \varphi_0\left(z^{\{\mathfrak{f}\}}\right) y_n + z^{\{\mathfrak{s}\}}\, \mu\left(z^{\{\mathfrak{f}\}}\right)^T Y,
\end{cases}
\end{aligned}$$

which leads to the stability function

$$\begin{aligned}
y_{n+1} &= R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) y_n,\\
R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) &:= \varphi_0\left(z^{\{\mathfrak{f}\}}\right) + z^{\{\mathfrak{s}\}}\, \mu\left(z^{\{\mathfrak{f}\}}\right)^T \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\right)^{-1} \mathbb{1}^{\{\mathfrak{s}\}},
\end{aligned} \tag{3.15}$$

where, following [129]:

$$\begin{aligned}
&\mu\left(z^{\{\mathfrak{f}\}}\right) := \sum_{k\geq 0} \gamma^k\, \varphi_{k+1}\left(z^{\{\mathfrak{f}\}}\right),\\
&\varphi_0(z) := e^z, \qquad \varphi_{k+1}(z) := \int_0^1 e^{z(1-t)} t^k \,\mathrm{d}t = \begin{cases} \frac{e^z-1}{z} & k = 0\\ \frac{k\varphi_k(z)-1}{z} & k > 0 \end{cases}.
\end{aligned}$$

Of special interest are cases when a partition becomes infinitely stiff. If the base method has bounded internal stability, the stability function (3.15) enjoys the following property:

$$\lim_{z^{\{\mathfrak{f}\}}\to-\infty} R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) = 0. \tag{3.16a}$$

Provided $\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}$ is invertible, e.g. the base method is SDIRK,

$$\lim_{z^{\{\mathfrak{s}\}}\to-\infty} R\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\big) = \varphi_0\big(z^{\{\mathfrak{f}\}}\big) - \mu\big(z^{\{\mathfrak{f}\}}\big)^T \left(\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\right)^{-1} \mathbb{1}^{\{\mathfrak{s}\}}. \tag{3.16b}$$

Although (3.16b) cannot be zero for all $z^{\{\mathfrak{f}\}}$ due to the linear independence of φ functions, its modulus is bounded for $z^{\{\mathfrak{f}\}} \in \mathbb{C}^-$.

Matrix Stability Analysis

Following [93, 129], consider the matrix test problem

$$\begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}' = \begin{bmatrix} \lambda^{\{\mathfrak{f}\}} & \eta^{\{\mathfrak{s}\}} \\ \eta^{\{\mathfrak{f}\}} & \lambda^{\{\mathfrak{s}\}} \end{bmatrix} \begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix} = \underbrace{\begin{bmatrix} \lambda^{\{\mathfrak{f}\}} & \frac{1-\xi}{\alpha}\left(\lambda^{\{\mathfrak{f}\}} - \lambda^{\{\mathfrak{s}\}}\right) \\ -\alpha\,\xi\left(\lambda^{\{\mathfrak{f}\}} - \lambda^{\{\mathfrak{s}\}}\right) & \lambda^{\{\mathfrak{s}\}} \end{bmatrix}}_{\mathbf{\Omega}} \begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}. \tag{3.17}$$

The change of variables that produces $\mathbf{\Omega}$ [129]

$$\alpha := \frac{\lambda^{\{\mathfrak{f}\}} - \lambda^{\{\mathfrak{s}\}} + \delta}{2\,\eta^{\{\mathfrak{s}\}}}, \quad \xi := \frac{\lambda^{\{\mathfrak{f}\}} - \lambda^{\{\mathfrak{s}\}} - \delta}{2\left(\lambda^{\{\mathfrak{f}\}} - \lambda^{\{\mathfrak{s}\}}\right)}, \quad \delta = \sqrt{4\,\eta^{\{\mathfrak{f}\}}\,\eta^{\{\mathfrak{s}\}} + \left(\lambda^{\{\mathfrak{f}\}} - \lambda^{\{\mathfrak{s}\}}\right)^2},$$

allows the matrix eigenvalues to be written as linear combinations of the diagonal entries: $\xi\,\lambda^{\{\mathfrak{f}\}} + (1-\xi)\,\lambda^{\{\mathfrak{s}\}}$ and $(1-\xi)\,\lambda^{\{\mathfrak{f}\}} + \xi\,\lambda^{\{\mathfrak{s}\}}$. The coupling between the fast and slow variables is controlled by ξ. Values close to zero indicate the slow system is weakly influenced by the fast one, while values close to one indicate the fast system is weakly influenced by the slow one.

Let

$$Z := \begin{bmatrix} z^{\{\mathfrak{f}\}} & w^{\{\mathfrak{s}\}} \\ w^{\{\mathfrak{f}\}} & z^{\{\mathfrak{s}\}} \end{bmatrix} := H \begin{bmatrix} \lambda^{\{\mathfrak{f}\}} & \eta^{\{\mathfrak{s}\}} \\ \eta^{\{\mathfrak{f}\}} & \lambda^{\{\mathfrak{s}\}} \end{bmatrix}, \qquad \widetilde{\mu}\big(z^{\{\mathfrak{f}\}}\big) := \sum_{k\geq 0} \frac{\gamma^k}{k+1} \varphi_{k+2}\big(z^{\{\mathfrak{f}\}}\big).$$

The component partitioned SPC-MRI-GARK method (3.9) applied to the matrix test problem (3.17) gives

$$\begin{bmatrix} Y^{\{\mathfrak{f}\}} \\ Y^{\{\mathfrak{s}\}} \end{bmatrix} = \left(I_{2s^{\{\mathfrak{s}\}}\times 2s^{\{\mathfrak{s}\}}} - Z \otimes \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\right)^{-1} \begin{bmatrix} y_n^{\{\mathfrak{f}\}}\,\mathbb{1}^{\{\mathfrak{s}\}} \\ y_n^{\{\mathfrak{s}\}}\,\mathbb{1}^{\{\mathfrak{s}\}} \end{bmatrix},$$

$$\begin{cases} v^{\{\mathfrak{f}\}}(0) = y_n^{\{\mathfrak{f}\}}, \\ \quad v^{\{\mathfrak{f}\}\prime} = \lambda^{\{\mathfrak{f}\}}\,v^{\{\mathfrak{f}\}} + \eta^{\{\mathfrak{s}\}}\,y_n^{\{\mathfrak{s}\}} + \eta^{\{\mathfrak{s}\}} \displaystyle\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \widetilde{\gamma}_j\left(\tfrac{\theta}{H}\right)\left(w^{\{\mathfrak{f}\}}\,Y_j^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}\,Y_j^{\{\mathfrak{s}\}}\right), \\ \qquad\quad = \lambda^{\{\mathfrak{f}\}}\,v^{\{\mathfrak{f}\}} + \eta^{\{\mathfrak{s}\}}\,y^{\{\mathfrak{s}\}} + \eta^{\{\mathfrak{s}\}} \displaystyle\sum_{k\geq 0} \frac{(\theta/H)^{k+1}}{k+1} \gamma^{k^T}\left(w^{\{\mathfrak{f}\}}\,Y^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}\,Y^{\{\mathfrak{s}\}}\right), \\ \begin{bmatrix} y_{n+1}^{\{\mathfrak{f}\}} \\ y_{n+1}^{\{\mathfrak{s}\}} \end{bmatrix} = \begin{bmatrix} v^{\{\mathfrak{f}\}}(H) \\ y_n^{\{\mathfrak{s}\}} + w^{\{\mathfrak{f}\}}\,\mathfrak{b}^{\{\mathfrak{s}\}T}\,Y^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}\,\mathfrak{b}^{\{\mathfrak{s}\}T}\,Y^{\{\mathfrak{s}\}} \end{bmatrix}, \end{cases}$$

We write the solution of the ODE as

$$y_{n+1}^{\{\mathfrak{f}\}} = \varphi_0\big(z^{\{\mathfrak{f}\}}\big)\, y_n^{\{\mathfrak{f}\}} + w^{\{\mathfrak{s}\}}\, \varphi_1\big(z^{\{\mathfrak{f}\}}\big)\, y_n^{\{\mathfrak{s}\}} + w^{\{\mathfrak{s}\}}\, \widetilde{\mu}\big(z^{\{\mathfrak{f}\}}\big)^T \left(w^{\{\mathfrak{f}\}}\, Y^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}\, Y^{\{\mathfrak{s}\}}\right).$$

The transfer matrix for the matrix test problem can be written as

$$\begin{aligned}
\begin{bmatrix} y_{n+1}^{\{\mathfrak{f}\}} \\ y_{n+1}^{\{\mathfrak{s}\}} \end{bmatrix} &= \mathbf{M}\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}, w^{\{\mathfrak{s}\}}, w^{\{\mathfrak{f}\}}\big) \begin{bmatrix} y_n^{\{\mathfrak{f}\}} \\ y_n^{\{\mathfrak{s}\}} \end{bmatrix}, \\
\mathbf{M}\big(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}, w^{\{\mathfrak{s}\}}, w^{\{\mathfrak{f}\}}\big) &\coloneqq \begin{bmatrix} \varphi_0\big(z^{\{\mathfrak{f}\}}\big) & w^{\{\mathfrak{s}\}}\varphi_1\big(z^{\{\mathfrak{f}\}}\big) \\ 0 & 1 \end{bmatrix} \\
&\quad + \begin{bmatrix} w^{\{\mathfrak{s}\}}\, w^{\{\mathfrak{f}\}}\, \widetilde{\mu}\big(z^{\{\mathfrak{f}\}}\big)^T & w^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{s}\}}\, \widetilde{\mu}\big(z^{\{\mathfrak{f}\}}\big)^T \\ w^{\{\mathfrak{f}\}}\, \boldsymbol{b}^{\{\mathfrak{s}\}T} & z^{\{\mathfrak{s}\}}\, \boldsymbol{b}^{\{\mathfrak{s}\}T} \end{bmatrix} \mathfrak{Y}(Z),
\end{aligned} \tag{3.18}$$

where $\mathfrak{Y}(Z)$ is the internal stability matrix:

$$\mathfrak{Y}(Z) \coloneqq \left(I_{2s^{\{\mathfrak{s}\}}\times 2s^{\{\mathfrak{s}\}}} - Z \otimes \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\right)^{-1} \left(I_{2\times 2} \otimes \mathbb{1}^{\{\mathfrak{s}\}}\right).$$

3.2.4 Construction of Practical SPC-MRI-GARK Methods

We develop new implicit SPC-MRI-GARK methods of up to order four. Their coefficients are presented in appendix C.1. The base methods are chosen to be existing, high-quality schemes that have either singly diagonally implicit (SDIRK) or explicit first stage single diagonally implicit (ESDIRK) structures. These offer a nice balance between stability and computational complexity. We note that explicit and fully implicit base methods can be employed as well. The $\gamma(t)$ coupling coefficients for each method are determined by first enforcing the order conditions, and then using remaining free parameters to optimize for stability. Plots of the scalar and matrix stability regions are provided in figs. C.1 and C.2, respectively. These regions are significantly larger than those of the decoupled MRI-GARK counterparts developed in [129].

3.3 Internal Stage Predictor-Corrector MRI-GARK Methods

Traditional multirate infinitesimal methods subdivide the integration interval $[t_n, t_{n+1}]$ into subintervals $[t_n + c_i^{\{\mathfrak{s}\}}\, H, t_n + c_{i+1}^{\{\mathfrak{s}\}}\, H]$, and solve a fast ODE over each subinterval. This advances the solution from one abscissa to the next, and then to the final solution. As illustrated in fig. 3.1c, an internal stage predictor-corrector MRI-GARK method follows this strategy, but also incorporates a predictor-corrector strategy similar to that used in SPC-MRI-GARK schemes. On each subinterval, the solution is first predicted with a traditional

Runge-Kutta stage calculation. Next, the fast components are refined by solving an ODE which uses previous predictor and corrector stages, as well as the current predictor stage, to implement the slow tendencies.

3.3.1 Method Definition

Again, we start with a slow Runge–Kutta base method (3.3), but now enforce that is has a diagonally implicit structure and the abscissae are non-decreasing:

$$0 \leq c_1^{\{\mathfrak{s}\}} \leq c_2^{\{\mathfrak{s}\}} \leq \ldots \leq c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} \leq 1.$$

This ensures that each ODE between stages is not integrated backward in time. We define the abscissa increments:

$$\Delta c^{\{\mathfrak{s}\}} = \left[c_1^{\{\mathfrak{s}\}}, c_2^{\{\mathfrak{s}\}} - c_1^{\{\mathfrak{s}\}}, \ldots, c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} - c_{s^{\{\mathfrak{s}\}}-1}^{\{\mathfrak{s}\}}\right]^T.$$

The final integration from $c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}}$ to 1 can introduce special cases that increase the complexity of the notation, order conditions, and stability analysis. We will impose that the base slow method is stiffly accurate [72], which makes the last stage equal to the final solution, and simplifies the subsequent analyses. This comes at no loss of generality since we can always rewrite a Runge–Kutta method into a reducible, but stiffly accurate form. In Butcher tableau notation we have:

$$\begin{array}{c|c} c^{\{\mathfrak{s}\}} & \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline & b^{\{\mathfrak{s}\}T} \end{array} \quad \rightarrow \quad \begin{array}{c|cc} c^{\{\mathfrak{s}\}} & \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} & 0^{\{\mathfrak{s}\}} \\ 1 & b^{\{\mathfrak{s}\}T} & 0 \\ \hline & b^{\{\mathfrak{s}\}T} & 0 \end{array}.$$

Definition 3.8 (Internal stage predictor corrector MRI-GARK methods)**.** One step of an internal stage predictor-corrector MRI-GARK (IPC-MRI-GARK) scheme applied to (3.1) is

given by

$$Y_0 := y_n, \quad c_0^{\{\mathfrak{s}\}} := 0, \tag{3.19a}$$

$$\begin{cases} Y_i^* = y_n + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} f_j + H\, a_{i,i}^{\{\mathfrak{s}\}} f_i^*, \\ T_{i-1} = t_n + c_{i-1}^{\{\mathfrak{s}\}} H, \\ v_i(0) = Y_{i-1}, \\ v_i' = \Delta c_i^{\{\mathfrak{s}\}} f^{\{\mathfrak{f}\}}\Big(T_{i-1} + \Delta c_i^{\{\mathfrak{s}\}}\, \theta, v_i\Big) + \sum_{j=1}^{i-1} \gamma_{i,j}\big(\tfrac{\theta}{H}\big)\, f_j^{\{\mathfrak{s}\}} \\ \qquad + \sum_{j=1}^{i} \psi_{i,j}\big(\tfrac{\theta}{H}\big)\, f_i^{\{\mathfrak{s}\}*}, \quad \text{for } \theta \in [0, H], \\ Y_i = v_i(H), \qquad i = 1, \dots, s^{\{\mathfrak{s}\}}, \end{cases} \tag{3.19b}$$

$$y_{n+1} = Y_{s^{\{\mathfrak{s}\}}}, \tag{3.19c}$$

with $f_j^{\{\mathfrak{s}\}*} := f^{\{\mathfrak{s}\}}\big(T_j, Y_j^*\big)$ and $f_j^* := f\big(T_j, Y_j^*\big)$. Stages and functions with an asterisk are predictor values, and terms without the asterisk are corrector values. In order to enforce that only previously computed stages appear in the ODE, we require that $\gamma_{i,j}(\tau) = 0$ for $j \geq i$ and $\psi_{i,j}(\tau) = 0$ for $j > i$.

Once again, we can take each $\gamma_{i,j}(t)$ and $\psi_{i,j}(t)$ to be polynomial in time. These and their integral terms $\widetilde{\gamma}_{i,j}(t)$, $\overline{\gamma}_{i,j}$, $\widetilde{\psi}_{i,j}(t)$, $\overline{\psi}_{i,j}$ are defined analogously to (3.5). The capitalized versions are used to denote the matrices of coefficients.

Remark 3.9 (Embedded method)**.** Following the strategy used in [129], an embedded solution can be obtained via the additional integration

$$\begin{aligned} \widehat{v}' &= \Delta c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} f^{\{\mathfrak{f}\}}\Big(T_{s^{\{\mathfrak{s}\}}-1} + \Delta c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}}\, \theta, \widehat{v}\Big) + \sum_{j=1}^{s^{\{\mathfrak{s}\}}-1} \widehat{\gamma}_j\big(\tfrac{\theta}{H}\big)\, f_j^{\{\mathfrak{s}\}} \\ &\quad + \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \widehat{\psi}_j\big(\tfrac{\theta}{H}\big)\, f_i^{\{\mathfrak{s}\}*}, \quad \text{for } \theta \in [0, H], \\ \widehat{y}_{n+1} &= \widehat{v}(H). \end{aligned}$$

With the trivial partitioning $f^{\{\mathfrak{s}\}} = f$, $f^{\{\mathfrak{f}\}} = 0$, the corrector stages simplify to

$$\begin{aligned} Y_i &= Y_{i-1} + H \sum_{j=1}^{i-1} \overline{\gamma}_{i,j}\, f_j^{\{\mathfrak{s}\}} + H \sum_{j=1}^{i} \overline{\psi}_{i,j}\, f_j^{\{\mathfrak{s}\}*} \\ &= y_n + H \sum_{j=1}^{i-1} \left(\sum_{\ell=j+1}^{i} \overline{\gamma}_{\ell,j} \right) f_j^{\{\mathfrak{s}\}} + H \sum_{j=1}^{i} \left(\sum_{\ell=j}^{i} \overline{\psi}_{\ell,j} \right) f_j^{\{\mathfrak{s}\}*}. \end{aligned} \tag{3.20}$$

The IPC-MRI-GARK method becomes a $2s^{\{\mathfrak{s}\}}$ stage Runge–Kutta method with $s^{\{\mathfrak{s}\}}$ predictor stages and $s^{\{\mathfrak{s}\}}$ corrector stages. In the absence of the fast component, it is natural to expect the predictor and corrector stages to coincide and for the method to degenerate into the slow base scheme. Matching coefficients of (3.20) to those of the predictor stage of (3.19) gives the self-consistency conditions

$$\mathcal{T}^{\{\mathfrak{s}\}} = E\,\overline{\Gamma} \quad \text{and} \quad \mathcal{D}^{\{\mathfrak{s}\}} = E\,\overline{\Psi}, \tag{3.21}$$

where $\mathcal{T}^{\{\mathfrak{s}\}}$ is the strictly lower triangular part of $\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}$ and

$$E = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{s^{\{\mathfrak{s}\}} \times s^{\{\mathfrak{s}\}}}, \qquad \mathcal{D}^{\{\mathfrak{s}\}} = \operatorname{diag}\left(a^{\{\mathfrak{s}\}}_{1,1}, \dots, a^{\{\mathfrak{s}\}}_{s^{\{\mathfrak{s}\}},s^{\{\mathfrak{s}\}}}\right).$$

Remark 3.10 (Repeated abscissae)**.** When $c_i^{\{\mathfrak{s}\}} = c_{i-1}^{\{\mathfrak{s}\}}$, the fast function disappears from the ODE in (3.19b) as it is scaled by zero. The corrector stage simplifies to

$$\begin{aligned} Y_i &= Y_{i-1} + \int_0^H \left(\sum_{j=1}^{i-1} \gamma_{i,j}\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}} + \sum_{j=1}^{i} \psi_{i,j}\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}*} \right) \mathrm{d}\theta \\ &= Y_{i-1} + H \sum_{j=1}^{i-1} \overline{\gamma}_{i,j}\, f_j^{\{\mathfrak{s}\}} + H \sum_{j=1}^{i} \overline{\psi}_{i,j}\, f_j^{\{\mathfrak{s}\}*}. \end{aligned}$$

Clearly, an ODE solver is no longer needed to compute Y_i. This can be viewed as modifying (the slow part of) the initial conditions for the next step's ODE.

For component partitioned systems (3.8), an IPC-MRI-GARK step reads:

$$Y_0^{\{\mathfrak{f}\}} = y_n^{\{\mathfrak{f}\}}, \quad Y_0^{\{\mathfrak{s}\}} = y_n^{\{\mathfrak{s}\}}, \quad c_0^{\{\mathfrak{s}\}} = 0, \tag{3.22a}$$

$$\begin{cases} \begin{bmatrix} Y_i^{\{\mathfrak{f}\}*} \\ Y_i^{\{\mathfrak{s}\}*} \end{bmatrix} = \begin{bmatrix} y_n^{\{\mathfrak{f}\}} + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{f}\}} + H a_{i,i}^{\{\mathfrak{s}\}} f_i^{\{\mathfrak{f}\}*} \\ y_n^{\{\mathfrak{s}\}} + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{s}\}} + H a_{i,i}^{\{\mathfrak{s}\}} f_i^{\{\mathfrak{s}\}*} \end{bmatrix}, \\ \quad Y_i^{\{\mathfrak{s}\}} = Y_i^{\{\mathfrak{s}\}*} \\ v^{\{\mathfrak{f}\}}(0) = Y_{i-1}^{\{\mathfrak{f}\}}, \\ \quad T_{i-1} = t_n + c_{i-1}^{\{\mathfrak{s}\}} H, \\ \quad v_i^{\{\mathfrak{f}\}\prime} = \Delta c_i^{\{\mathfrak{s}\}} f^{\{\mathfrak{f}\}} \left(T_{i-1} + \Delta c_i^{\{\mathfrak{s}\}}\, \theta, v_i^{\{\mathfrak{f}\}}, Y_{i-1}^{\{\mathfrak{s}\}} + H \sum_{j=1}^{i} \widetilde{\delta}_{i,j}\left(\tfrac{\theta}{H}\right) f_j^{\{\mathfrak{s}\}} \right), \\ \qquad \text{for } \theta \in [0, H], \\ \quad Y_i^{\{\mathfrak{f}\}} = v_i^{\{\mathfrak{f}\}}(H), \quad i = 1, \dots, s^{\{\mathfrak{s}\}}, \end{cases} \tag{3.22b}$$

$$\begin{bmatrix} y_{n+1}^{\{\mathfrak{f}\}} \\ y_{n+1}^{\{\mathfrak{s}\}} \end{bmatrix} = \begin{bmatrix} Y_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{f}\}} \\ Y_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} \end{bmatrix}, \tag{3.22c}$$

where $\widetilde{\delta}_{i,j}\left(\frac{\theta}{H}\right) = \widetilde{\gamma}_{i,j}\left(\frac{\theta}{H}\right) + \widetilde{\psi}_{i,j}\left(\frac{\theta}{H}\right)$ and $f_j^{\{\mathfrak{f}\}*} \coloneqq f^{\{\mathfrak{f}\}}\left(T_j, Y_j^{\{\mathfrak{f}\}*}, Y_j^{\{\mathfrak{s}\}*}\right)$.

3.3.2 Order Conditions

Following section 3.2.2, we look to utilize GARK order condition theory to derive order conditions for IPC-MRI-GARK methods. Again, we apply an arbitrarily accurate Runge–Kutta method $\left(A^{\{\mathfrak{f},\mathfrak{f}\}}, b^{\{\mathfrak{f}\}}, c^{\{\mathfrak{f}\}}\right)$ to discretize the ODEs and recover the GARK stages and GARK tableau. We use the labels $\mathfrak{p}$ and $\mathfrak{c}$ to denote predictor and corrector stages, respectively. Also, we define $Y_k^{\{\mathfrak{f},i\}}$ to be the k-th stage of the discretized ODE between abscissae $c_{i-1}^{\{\mathfrak{s}\}}$ and $c_i^{\{\mathfrak{s}\}}$. Now, the i-th step of (3.19) is composed of the GARK stages

$$
\begin{aligned}
Y_i^{\{\mathfrak{f},\mathfrak{p}\}} &= Y_i^{\{\mathfrak{s},\mathfrak{p}\}} = y_n + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{f},\mathfrak{c}\}} + H\, a_{i,i}^{\{\mathfrak{s}\}} f_i^{\{\mathfrak{f},\mathfrak{p}\}} + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{s},\mathfrak{c}\}} \\
&\quad + H\, a_{i,i}^{\{\mathfrak{s}\}} f_i^{\{\mathfrak{s},\mathfrak{p}\}}, \\
Y_k^{\{\mathfrak{f},i\}} &= Y_{i-1}^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{k,j}^{\{\mathfrak{f},\mathfrak{f}\}} \left(\Delta c_i^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{f},i\}} + \sum_{\ell=1}^{i-1} \gamma_{i,\ell}\left(c_j^{\{\mathfrak{f}\}}\right) f_\ell^{\{\mathfrak{s},\mathfrak{c}\}} \right. \\
&\quad \left. + \sum_{\ell=1}^{i} \psi_{i,\ell}\left(c_j^{\{\mathfrak{f}\}}\right) f_\ell^{\{\mathfrak{s},\mathfrak{p}\}} \right), \\
Y_i^{\{\mathfrak{f},\mathfrak{c}\}} &= Y_i^{\{\mathfrak{s},\mathfrak{c}\}} = Y_{i-1}^{\{\mathfrak{f},\mathfrak{c}\}} + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} \left(\Delta c_i^{\{\mathfrak{s}\}} f_j^{\{\mathfrak{f},i\}} + \sum_{\ell=1}^{i-1} \gamma_{i,\ell}\left(c_j^{\{\mathfrak{f}\}}\right) f_\ell^{\{\mathfrak{s},\lambda\}} \right. \\
&\quad \left. + \sum_{\ell=1}^{i} \psi_{i,\ell}\left(c_j^{\{\mathfrak{f}\}}\right) f_\ell^{\{\mathfrak{s},\mathfrak{p}\}} \right),
\end{aligned}
$$

with $f_j^{\{\mathfrak{f},i\}} := f^{\{\mathfrak{f}\}}\big(T_{i-1} + \Delta c_i^{\{\mathfrak{s}\}} c_j^{\{\mathfrak{f}\}}, Y_j^{\{\mathfrak{f},i\}}\big)$ and $f_j^{\{\sigma,\nu\}} := f^{\{\sigma\}}\big(T_j, Y_j^{\{\sigma,\nu\}}\big)$ for $\sigma \in \{\mathfrak{f},\mathfrak{s}\}$ and $\nu \in \{\mathfrak{p},\mathfrak{c}\}$. Now, we simplify $Y_i^{\{\mathfrak{f},\mathfrak{c}\}}$ to obtain:

$$
\begin{aligned}
Y_i^{\{\mathfrak{f},\mathfrak{c}\}} &= Y_{i-1}^{\{\mathfrak{f},\mathfrak{c}\}} + \Delta c_i^{\{\mathfrak{s}\}} H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} f_j^{\{\mathfrak{f},i\}} + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} \sum_{\ell=1}^{i-1} \left(\sum_{k\geq 0} \gamma_{i,\ell}^k \, c_j^{\{\mathfrak{f}\}\times k} \right) f_\ell^{\{\mathfrak{s},\mathfrak{c}\}} \\
&\quad + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} \sum_{\ell=1}^{i} \left(\sum_{k\geq 0} \psi_{i,\ell}^k \, c_j^{\{\mathfrak{f}\}\times k} \right) f_\ell^{\{\mathfrak{s},\mathfrak{p}\}} \\
&= Y_{i-1}^{\{\mathfrak{f},\mathfrak{c}\}} + \Delta c_i^{\{\mathfrak{s}\}} H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} f_j^{\{\mathfrak{f},i\}} + H \sum_{\ell=1}^{i-1} \left(\sum_{k\geq 0} \frac{\gamma_{i,\ell}^k}{k+1} \right) f_\ell^{\{\mathfrak{s},\mathfrak{c}\}} \\
&\quad + H \sum_{\ell=1}^{i} \left(\sum_{k\geq 0} \frac{\psi_{i,\ell}^k}{k+1} \right) f_\ell^{\{\mathfrak{s},\mathfrak{p}\}} \\
&= Y_{i-1}^{\{\mathfrak{f},\mathfrak{c}\}} + \Delta c_i^{\{\mathfrak{s}\}} \, H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} f_j^{\{\mathfrak{f},i\}} + H \sum_{j=1}^{i-1} \overline{\gamma}_{i,j} \, f_j^{\{\mathfrak{s},\mathfrak{c}\}} + H \sum_{j=1}^{i} \overline{\psi}_{i,j} \, f_j^{\{\mathfrak{s},\mathfrak{p}\}} \\
&= y_n + H \sum_{j=1}^{i} \Delta c_j^{\{\mathfrak{s}\}} \sum_{\ell=1}^{s^{\{\mathfrak{f}\}}} b_\ell^{\{\mathfrak{f}\}} \, f_\ell^{\{\mathfrak{f},j\}} + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} \, f_j^{\{\mathfrak{s},\mathfrak{c}\}} + H \, a_{i,i}^{\{\mathfrak{s}\}} \, f_i^{\{\mathfrak{s},\mathfrak{p}\}}.
\end{aligned}
$$

The stages of the discretized ODE simplify to

$$
\begin{aligned}
Y_k^{\{\mathfrak{f},i\}} &= y_n + H \sum_{j=1}^{i-1} \Delta c_j^{\{\mathfrak{s}\}} \sum_{\ell=1}^{s^{\{\mathfrak{f}\}}} b_\ell^{\{\mathfrak{f}\}} \, f_\ell^{\{\mathfrak{f},j\}} + \Delta c_i^{\{\mathfrak{s}\}} H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{k,j}^{\{\mathfrak{f},\mathfrak{f}\}} \, f_j^{\{\mathfrak{f},i\}} \\
&\quad + H \sum_{j=1}^{i-2} a_{i,j}^{\{\mathfrak{s}\}} \, f_j^{\{\mathfrak{s},\mathfrak{c}\}} + H \sum_{j=1}^{i-1} \left(\sum_{\ell=1}^{s^{\{\mathfrak{f}\}}} a_{k,\ell}^{\{\mathfrak{f},\mathfrak{f}\}} \, \gamma_{i,j}\big(c_j^{\{\mathfrak{f}\}}\big) \right) f_j^{\{\mathfrak{s},\mathfrak{c}\}} \\
&\quad + H \, a_{i-1,i-1}^{\{\mathfrak{s}\}} \, f_{i-1}^{\{\mathfrak{s},\mathfrak{p}\}} + H \sum_{j=1}^{i} \left(\sum_{\ell=1}^{s^{\{\mathfrak{f}\}}} a_{k,\ell}^{\{\mathfrak{f},\mathfrak{f}\}} \, \psi_{i,j}\big(c_j^{\{\mathfrak{f}\}}\big) \right) f_j^{\{\mathfrak{s},\mathfrak{p}\}}.
\end{aligned}
$$

The coefficients appearing in the stages can be organized into the following GARK tableau:

$$
\begin{array}{c|ccc|cc|c}
c^{\{\mathfrak{s}\}} & \mathcal{D}^{\{\mathfrak{s}\}} & 0 & \mathcal{T}^{\{\mathfrak{s}\}} & \mathcal{D}^{\{\mathfrak{s}\}} & \mathcal{T}^{\{\mathfrak{s}\}} & c^{\{\mathfrak{s}\}} \\
c^{\{\mathfrak{f},\mathfrak{f},i\}} & 0 & A^{\{\mathfrak{f},\mathfrak{f},i,i\}} & 0 & A^{\{\mathfrak{f},\mathfrak{s},i,\mathfrak{p}\}} & A^{\{\mathfrak{f},\mathfrak{s},i,\mathfrak{c}\}} & c^{\{\mathfrak{f},\mathfrak{s},i\}} \\
c^{\{\mathfrak{s}\}} & 0 & A^{\{\mathfrak{f},\mathfrak{f},\mathfrak{c},i\}} & 0 & \mathcal{D}^{\{\mathfrak{s}\}} & \mathcal{T}^{\{\mathfrak{s}\}} & c^{\{\mathfrak{s}\}} \\
\hline
c^{\{\mathfrak{s}\}} & \mathcal{D}^{\{\mathfrak{s}\}} & 0 & \mathcal{T}^{\{\mathfrak{s}\}} & \mathcal{D}^{\{\mathfrak{s}\}} & \mathcal{T}^{\{\mathfrak{s}\}} & c^{\{\mathfrak{s}\}} \\
c^{\{\mathfrak{s}\}} & 0 & A^{\{\mathfrak{s},\mathfrak{f},\mathfrak{c},i\}} & 0 & \mathcal{D}^{\{\mathfrak{s}\}} & \mathcal{T}^{\{\mathfrak{s}\}} & c^{\{\mathfrak{s}\}} \\
\hline
 & 0 & \Delta c^{\{\mathfrak{s}\}T} \otimes b^{\{\mathfrak{f}\}T} & 0 & e_{s^{\{\mathfrak{s}\}}}^T \mathcal{D}^{\{\mathfrak{s}\}} & e_{s^{\{\mathfrak{s}\}}}^T \mathcal{T}^{\{\mathfrak{s}\}} &
\end{array}
\tag{3.23}
$$

The unspecified entries are

$$
\begin{aligned}
A^{\{\mathfrak{f},\mathfrak{f},i,i\}} &= L\,\Delta\mathcal{C}^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}}\, b^{\{\mathfrak{f}\}T} + \operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right) \otimes A^{\{\mathfrak{f},\mathfrak{f}\}},\\
A^{\{\mathfrak{f},\mathfrak{s},i,\mathfrak{p}\}} &= \sum_{k\geq 0} \Psi^k \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k} + L\,\mathcal{D}^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}},\\
A^{\{\mathfrak{f},\mathfrak{s},i,\mathfrak{c}\}} &= \sum_{k\geq 0} \Gamma^k \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k} + L\,\mathcal{T}^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}},\\
A^{\{\mathfrak{f},\mathfrak{f},\mathfrak{c},i\}} &= A^{\{\mathfrak{s},\mathfrak{f},\mathfrak{c},i\}} = \Delta\mathcal{C}^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}T},\\
c^{\{\mathfrak{f},\mathfrak{f},i\}} &= L\, c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \Delta c^{\{\mathfrak{s}\}} \otimes c^{\{\mathfrak{f}\}},\\
c^{\{\mathfrak{f},\mathfrak{s},i\}} &= L\, c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0} \left(\Psi^k + \Gamma^k\right) \mathbb{1}^{\{\mathfrak{s}\}} \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k},
\end{aligned}
$$

with

$$
\Delta\mathcal{C}^{\{\mathfrak{s}\}} = \begin{bmatrix} \Delta c_1^{\{\mathfrak{s}\}} & & & \\ \Delta c_1^{\{\mathfrak{s}\}} & \Delta c_2^{\{\mathfrak{s}\}} & & \\ \vdots & \vdots & \ddots & \\ \Delta c_1^{\{\mathfrak{s}\}} & \Delta c_2^{\{\mathfrak{s}\}} & \dots & \Delta c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} \end{bmatrix},
$$

and $L \in \mathrm{Re}^{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}}$ is a lower shift matrix with entries $L_{i,j} = \delta_{i,j+1}$.

Internal Consistency

Theorem 3.11 (Internal consistency conditions)**.** *An IPC-MRI-GARK method* (3.19) *fulfills the "internal consistency" conditions*

$$
\mathbf{c}^{\{\mathfrak{s},\mathfrak{f}\}} = \mathbf{c}^{\{\mathfrak{s},\mathfrak{s}\}} \equiv c^{\{\mathfrak{s}\}} \quad \textit{and} \quad \mathbf{c}^{\{\mathfrak{f},\mathfrak{f}\}} = \mathbf{c}^{\{\mathfrak{f},\mathfrak{s}\}}
$$

for any fast method iff the following conditions hold:

$$
\left(\Psi^0 + \Gamma^0\right) \mathbb{1}^{\{\mathfrak{s}\}} = \Delta c^{\{\mathfrak{s}\}} \quad \textit{and} \quad \left(\Psi^k + \Gamma^k\right) \mathbb{1}^{\{\mathfrak{s}\}} = 0 \quad \forall k \geq 1. \tag{3.24}
$$

Proof. All internal consistency equations are automatically satisfied except

$$
\begin{aligned}
c^{\{\mathfrak{f},\mathfrak{f},i\}} = c^{\{\mathfrak{f},\mathfrak{s},i\}} \qquad &\Leftrightarrow\\
L c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \Delta c^{\{\mathfrak{s}\}} \otimes c^{\{\mathfrak{f}\}} = L c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} &+ \sum_{k\geq 0} \left(\Psi^k + \Gamma^k\right) \mathbb{1}^{\{\mathfrak{s}\}} \otimes A^{\{\mathfrak{f},\mathfrak{f}\}} c^{\{\mathfrak{f}\}\times k}.
\end{aligned}
$$

It is easy to confirm (3.24) is sufficient to satisfy this condition, and thus, internal consistency. Since the equality must hold for all $A^{\{\mathfrak{f},\mathfrak{f}\}}$, it must hold when all $A^{\{\mathfrak{f},\mathfrak{f}\}} c^{\{\mathfrak{f}\}\times k}$ are linearly independent. Matching powers of the left- and right-hand sides proves the necessity of (3.24). □

Like with SPC-MRI-GARK methods, internal consistency and a slow base method of order two guarantees an IPC-MRI-GARK method is order two [132].

Fourth order conditions

In this section, we derive order conditions of the IPC-MRI-GARK schemes for up to order four.

Lemma 3.12 (Intermediate matrix products)**.** *The coefficients of the GARK tableau* (3.23) *satisfy*

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}\times\ell} = \begin{bmatrix} \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times\ell} \\ \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times\ell} \end{bmatrix}, \tag{3.25a}$$

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}\times\ell} = \begin{bmatrix} \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times\ell} \\ \frac{1}{\ell+1}c^{\{\mathfrak{s}\}\times(\ell+1)} \end{bmatrix}, \tag{3.25b}$$

$$\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}\times\ell} = \begin{bmatrix} \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times\ell} \\ L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times\ell}\otimes\mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0}\left(\Psi^k+\Gamma^k\right)c^{\{\mathfrak{s}\}\times\ell}\otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\,c^{\{\mathfrak{f}\}\times k} \\ \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times\ell} \end{bmatrix}. \tag{3.25c}$$

Theorem 3.13 (Fourth order coupling conditions)**.** *An internally consistent IPC-MRI-GARK method* (3.19) *satisfying* (3.21) *has order four iff the slow base scheme has order at least four, and the following coupling conditions hold:*

$$\frac{1}{6} = \Delta c^{\{\mathfrak{s}\}T}\left(L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\left(\Psi^k+\Gamma^k\right)\right)c^{\{\mathfrak{s}\}}, \qquad \textit{(order 3)} \tag{3.26a}$$

$$\frac{1}{6} = e_{s^{\{\mathfrak{s}\}}}^T\left(\mathcal{D}^{\{\mathfrak{s}\}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}} + \frac{1}{2}\mathcal{T}^{\{\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times 2}\right), \qquad \textit{(order 3)} \tag{3.26b}$$

$$\begin{aligned}\frac{1}{8} &= \left(\Delta c^{\{\mathfrak{s}\}}\times L\,c^{\{\mathfrak{s}\}}\right)^T\left(L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\left(\Psi^k+\Gamma^k\right)\right)c^{\{\mathfrak{s}\}} \\ &\quad + \left(\Delta c^{\{\mathfrak{s}\}\times 2}\right)^T\left(\frac{1}{2}L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\psi_k\left(\Psi^k+\Gamma^k\right)\right)c^{\{\mathfrak{s}\}},\end{aligned} \qquad \textit{(order 4)} \tag{3.26c}$$

$$\begin{aligned}\frac{1}{8} &= \left(e_{s^{\{\mathfrak{s}\}}}^T\,\mathcal{D}^{\{\mathfrak{s}\}}\times c^{\{\mathfrak{s}\}T}\right)\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}} \\ &\quad + \frac{1}{2}\left(e_{s^{\{\mathfrak{s}\}}}^T\,\mathcal{T}^{\{\mathfrak{s}\}}\times c^{\{\mathfrak{s}\}T}\right)c^{\{\mathfrak{s}\}\times 2}\end{aligned} \qquad \textit{(order 4)} \tag{3.26d}$$

$$\frac{1}{12} = \Delta c^{\{\mathfrak{s}\}T}\left(L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\left(\Psi^k+\Gamma^k\right)\right)c^{\{\mathfrak{s}\}\times 2}, \qquad \textit{(order 4)} \tag{3.26e}$$

$$\frac{1}{12} = e_{s^{\{\mathfrak{s}\}}}^T\left(\mathcal{D}^{\{\mathfrak{s}\}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times 2} + \frac{1}{3}\mathcal{T}^{\{\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times 3}\right), \qquad \textit{(order 4)} \tag{3.26f}$$

$$\frac{1}{24} = \Delta c^{\{\mathfrak{s}\}T}\, L\, \Delta \mathcal{C}^{\{\mathfrak{s}\}} \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}} + \left(\Delta c^{\{\mathfrak{s}\}\times 2}\right)^T \left(\frac{1}{2} L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \xi_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}}, \quad \textit{(order 4)} \tag{3.26g}$$

$$\frac{1}{24} = \Delta c^{\{\mathfrak{s}\}T} \left(L\, \mathcal{D}^{\{\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \Psi^k \right) \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} + \frac{1}{2} \Delta c^{\{\mathfrak{s}\}T} \left(L\, \mathcal{T}^{\{\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \Gamma^k \right) c^{\{\mathfrak{s}\}\times 2}, \quad \textit{(order 4)} \tag{3.26h}$$

$$\frac{1}{24} = \Delta c^{\{\mathfrak{s}\}T} \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}}, \quad \textit{(order 4)} \tag{3.26i}$$

$$\frac{1}{24} = e^T_{s^{\{\mathfrak{s}\}}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} \left(\mathcal{D}^{\{\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} + \frac{1}{2} \mathcal{T}^{\{\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}\times 2} \right), \quad \textit{(order 4)} \tag{3.26j}$$

$$\frac{1}{24} = e^T_{s^{\{\mathfrak{s}\}}}\, \mathcal{D}^{\{\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} + e^T_{s^{\{\mathfrak{s}\}}}\, \mathcal{T}^{\{\mathfrak{s}\}}\, \Delta \mathcal{C}^{\{\mathfrak{s}\}} \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}}. \quad \textit{(order 4)} \tag{3.26k}$$

Proof. An internally consistent GARK scheme is order four iff the base methods are order four and the 12 coupling conditions up to order four are satisfied [132]. We proceed with checking each coupling condition.

Condition 3a By using (3.25c), the first third order condition gives (3.26a):

$$\begin{aligned}
\frac{1}{6} &= \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} \\
&= \left(\Delta c^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}}\right)^T \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0} \left(\Psi^k + \Gamma^k\right) c^{\{\mathfrak{s}\}} \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k} \right) \\
&= \Delta c^{\{\mathfrak{s}\}T} \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}}.
\end{aligned}$$

Condition 3b The other third order condition is expanded with (3.25b) to get (3.26b):

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} = e^T_{s^{\{\mathfrak{s}\}}} \left(\mathcal{D}^{\{\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} + \frac{1}{2} \mathcal{T}^{\{\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}\times 2} \right).$$

Condition 4a By using (3.25c), the first fourth order condition gives (3.26c):

$$
\begin{aligned}
\frac{1}{8} &= \left(\mathbf{b}^{\{\mathfrak{f}\}} \times \mathbf{c}^{\{\mathfrak{f}\}}\right)^T \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}} \\
&= \left(\left(\Delta c^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}}\right) \times \left(L\, c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \Delta c^{\{\mathfrak{s}\}} \otimes c^{\{\mathfrak{f}\}}\right)\right)^T \\
&\quad \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0} \left(\Psi^k + \Gamma^k\right) c^{\{\mathfrak{s}\}} \otimes A^{\{\mathfrak{f},\mathfrak{f}\}} c^{\{\mathfrak{f}\}\times k}\right) \\
&= \left(\Delta c^{\{\mathfrak{s}\}} \times L\, c^{\{\mathfrak{s}\}}\right)^T \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}} \\
&\quad + \left(\Delta c^{\{\mathfrak{s}\}\times 2}\right)^T \left(\frac{1}{2} L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \psi_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}}.
\end{aligned}
$$

Condition 4b We derive (3.26d) with (3.25b):

$$
\begin{aligned}
\frac{1}{8} &= \left(\mathbf{b}^{\{\mathfrak{s}\}} \times \mathbf{c}^{\{\mathfrak{s}\}}\right)^T \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} \\
&= \left(e^T_{s^{\{\mathfrak{s}\}}}\, \mathcal{D}^{\{\mathfrak{s}\}} \times c^{\{\mathfrak{s}\}T}\right) \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} + \frac{1}{2}\left(e^T_{s^{\{\mathfrak{s}\}}}\, \mathcal{T}^{\{\mathfrak{s}\}} \times c^{\{\mathfrak{s}\}T}\right) c^{\{\mathfrak{s}\}\times 2}.
\end{aligned}
$$

Condition 4c We derive (3.26e) with (3.25c):

$$
\frac{1}{12} = \mathbf{b}^{\{\mathfrak{f}\}T}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\, \mathbf{c}^{\{\mathfrak{s}\}\times 2} = \Delta c^{\{\mathfrak{s}\}T} \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}\times 2}.
$$

Condition 4d We derive (3.26f) with (3.25b):

$$
\frac{1}{12} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}\times 2} = e^T_{s^{\{\mathfrak{s}\}}} \left(\mathcal{D}^{\{\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}\times 2} + \frac{1}{3} \mathcal{T}^{\{\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}\times 3}\right).
$$

Condition 4e We derive (3.26g) with (3.25c):

$$
\begin{aligned}
\frac{1}{24} &= \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} \\
&= \left(\Delta c^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}}\right)^T \left(L\,\Delta\mathcal{C}^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}}\,b^{\{\mathfrak{f}\}T} + \operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right) \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\right) \\
&\quad \left(L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0}\left(\Psi^k + \Gamma^k\right) c^{\{\mathfrak{s}\}} \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\,c^{\{\mathfrak{f}\}\times k}\right) \\
&= \Delta c^{\{\mathfrak{s}\}T}\,L\,\Delta\mathcal{C}^{\{\mathfrak{s}\}}\left(L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}} \\
&\quad + \left(\Delta c^{\{\mathfrak{s}\}\times 2}\right)^T \left(\frac{1}{2}L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\xi_k\left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}}.
\end{aligned}
$$

Condition 4f We derive (3.26h) with (3.25b):

$$
\begin{aligned}
\frac{1}{24} &= \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} \\
&= \frac{1}{2}\left(\Delta c^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}}\right)^T \left(L\,\mathcal{T}^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0}\Gamma^k \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\,c^{\{\mathfrak{f}\}\times k}\right) c^{\{\mathfrak{s}\}\times 2} \\
&\quad + \left(\Delta c^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}}\right)^T \left(L\,\mathcal{D}^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0}\Psi^k \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\,c^{\{\mathfrak{f}\}\times k}\right) \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}} \\
&= \Delta c^{\{\mathfrak{s}\}T}\left(L\,\mathcal{D}^{\{\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\Psi^k\right)\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}} \\
&\quad + \frac{1}{2}\Delta c^{\{\mathfrak{s}\}T}\left(L\,\mathcal{T}^{\{\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\Gamma^k\right) c^{\{\mathfrak{s}\}\times 2}.
\end{aligned}
$$

Condition 4g We derive (3.26i) with (3.25a):

$$
\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \Delta c^{\{\mathfrak{s}\}T}\left(L\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0}\zeta_k\left(\Psi^k + \Gamma^k\right)\right)\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}}.
$$

Condition 4h We derive (3.26j) with (3.25b):

$$
\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = e^T_{s^{\{\mathfrak{s}\}}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\left(\mathcal{D}^{\{\mathfrak{s}\}}\,\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}} + \frac{1}{2}\mathcal{T}^{\{\mathfrak{s}\}}\,c^{\{\mathfrak{s}\}\times 2}\right).
$$

Condition 4i We derive (3.26k) with (3.25c):

$$
\begin{aligned}
\frac{1}{24} &= \mathbf{b}^{\{\mathfrak{s}\}T}\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\mathbf{c}^{\{\mathfrak{s}\}} \\
&= e_{s^{\{\mathfrak{s}\}}}^T\, \mathcal{D}^{\{\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} + e_{s^{\{\mathfrak{s}\}}}^T\, \mathcal{T}^{\{\mathfrak{s}\}} \left(\Delta \mathcal{C}^{\{\mathfrak{s}\}} \otimes b^{\{\mathfrak{f}\}T}\right) \\
&\quad \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} \otimes \mathbb{1}^{\{\mathfrak{f}\}} + \sum_{k\geq 0} \left(\Psi^k + \Gamma^k\right) c^{\{\mathfrak{s}\}} \otimes A^{\{\mathfrak{f},\mathfrak{f}\}}\, c^{\{\mathfrak{f}\}\times k}\right) \\
&= e_{s^{\{\mathfrak{s}\}}}^T\, \mathcal{D}^{\{\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}\, c^{\{\mathfrak{s}\}} \\
&\quad + e_{s^{\{\mathfrak{s}\}}}^T\, \mathcal{T}^{\{\mathfrak{s}\}}\, \Delta \mathcal{C}^{\{\mathfrak{s}\}} \left(L\, \mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}} + \sum_{k\geq 0} \zeta_k \left(\Psi^k + \Gamma^k\right)\right) c^{\{\mathfrak{s}\}}.
\end{aligned}
$$

Condition 4j This condition is equivalent to condition 4d since the fast base method has an arbitrarily large stage order, and thus, $\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\mathbf{c}^{\{\mathfrak{f}\}} = \frac{1}{2}\mathbf{c}^{\{\mathfrak{f}\}\times 2}$:

$$
\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}} \quad \Leftrightarrow \quad \frac{1}{12} = \mathbf{b}^{\{\mathfrak{s}\}T}\, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\, \mathbf{c}^{\{\mathfrak{f}\}\times 2}.
$$

□

3.3.3 Linear Stability Analysis

Scalar Stability Analysis

We revisit the scalar linear test problem (3.14) now for IPC-MRI-GARK methods. The predictor stages in vector form are given by

$$
\begin{aligned}
Y^* &= y_n\, \mathbb{1}^{\{\mathfrak{s}\}} + z\, \mathcal{T}^{\{\mathfrak{s}\}}\, Y + z\, \mathcal{D}^{\{\mathfrak{s}\}}\, Y^* \\
&= \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\, \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \left(y_n\, \mathbb{1}^{\{\mathfrak{s}\}} + z\, \mathcal{T}^{\{\mathfrak{s}\}}\, Y\right).
\end{aligned} \tag{3.27}
$$

The internal ODEs become

$$
v' = \lambda^{\{\mathfrak{f}\}}\, \mathrm{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right) v + \lambda^{\{\mathfrak{s}\}}\, \Psi\left(\tfrac{\theta}{H}\right) Y^* + \lambda^{\{\mathfrak{s}\}}\, \Gamma\left(\tfrac{\theta}{H}\right) Y.
$$

Integrating and substituting in the predicted stages (3.27) gives

$$
\begin{aligned}
Y &= v(H) \\
&= \operatorname{diag}\left(\varphi_0\left(\Delta c^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right)\right) L\, Y + \varphi_0\left(\Delta c_1^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right) y_n\, e_1 + z^{\{\mathfrak{s}\}}\, \mu\left(z^{\{\mathfrak{f}\}}\right) Y \\
&\quad + z^{\{\mathfrak{s}\}}\, \nu\left(z^{\{\mathfrak{f}\}}\right) Y^* \\
&= \operatorname{diag}\left(\varphi_0\left(\Delta c^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right)\right) L\, Y + \varphi_0\left(\Delta c_1^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right) y_n\, e_1 + z^{\{\mathfrak{s}\}} \mu\left(z^{\{\mathfrak{f}\}}\right) Y \\
&\quad + z^{\{\mathfrak{s}\}}\, \nu\left(z^{\{\mathfrak{f}\}}\right) \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\, \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \left(y_n\, \mathbb{1}^{\{\mathfrak{s}\}} + z\, \mathcal{T}^{\{\mathfrak{s}\}}\, Y\right) \\
&= \mathfrak{M}\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right)^{-1} \\
&\quad \left(\varphi_0\left(\Delta c_1^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right) e_1 + z^{\{\mathfrak{s}\}}\, \nu\left(z^{\{\mathfrak{f}\}}\right) \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\, \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \mathbb{1}^{\{\mathfrak{s}\}}\right) y_n,
\end{aligned}
$$

Applying (3.19) to (3.14) gives the scalar stability function

$$
\begin{aligned}
R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) &:= e_{s^{\{\mathfrak{s}\}}}^T \mathfrak{M}\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right)^{-1} \\
&\quad \left(\varphi_0\left(\Delta c_1^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right) e_1 + z^{\{\mathfrak{s}\}}\, \nu\left(z^{\{\mathfrak{f}\}}\right) \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\, \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \mathbb{1}^{\{\mathfrak{s}\}}\right),
\end{aligned} \tag{3.28}
$$

with

$$
\begin{aligned}
\mu\left(z^{\{\mathfrak{f}\}}\right) &:= \sum_{k\geq 0} \operatorname{diag}\left(\varphi_{k+1}\left(\Delta c^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right)\right) \Gamma^k, \\
\nu\left(z^{\{\mathfrak{f}\}}\right) &:= \sum_{k\geq 0} \operatorname{diag}\left(\varphi_{k+1}\left(\Delta c^{\{\mathfrak{s}\}} z^{\{\mathfrak{s}\}}\right)\right) \Psi^k, \\
\mathfrak{M}\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) &:= I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - \operatorname{diag}\left(\varphi_0\left(\Delta c^{\{\mathfrak{s}\}} z^{\{\mathfrak{f}\}}\right)\right) L - z^{\{\mathfrak{s}\}} \mu\left(z^{\{\mathfrak{f}\}}\right) \\
&\quad - z^{\{\mathfrak{s}\}}\, z\, \nu\left(z^{\{\mathfrak{f}\}}\right) \left(I_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} - z\, \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \mathcal{T}^{\{\mathfrak{s}\}}.
\end{aligned}
$$

It can be verified that (3.16a) still holds. As the slow part of the test problem becomes infinitely stiff, however, we would like the stability function to be bounded. One natural way to enforce this is by ensuring $\mathfrak{M}$, and thus $\mathfrak{M}^{-1}$, remains bounded in the limit. The last two terms in $\mathfrak{M}$ are problematic since they are $\mathcal{O}\left(z^{\{\mathfrak{s}\}}\right)$. If $\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}$ is invertible, the following condition ensures these terms cancel in the limit:

$$
\mu\left(z^{\{\mathfrak{f}\}}\right) = \nu\left(z^{\{\mathfrak{f}\}}\right) \left(\mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \mathcal{T}^{\{\mathfrak{s}\}}.
$$

Note that μ and ν are sums over linearly independent φ functions. By matching terms in this summation, we arrive at the stability simplifying assumption

$$
\Gamma^k = \Psi^k \left(\mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1} \mathcal{T}^{\{\mathfrak{s}\}}, \qquad \forall k \geq 0. \tag{3.29}
$$

If Ψ and Γ are degree zero polynomials, then (3.21) automatically ensures (3.29) is satisfied.

Matrix Stability Analysis

Now we consider the component partitioned IPC-MRI-GARK method (3.22) applied to the matrix test problem (3.17). First, we define the following intermediate quantities:

$$
\begin{aligned}
\mathfrak{P}_1(Z) &\coloneqq \left(I_{2s^{\{\mathfrak{s}\}}\times 2s^{\{\mathfrak{s}\}}} - Z\otimes \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1}\left(I_{2\times 2}\otimes \mathbb{1}^{\{\mathfrak{s}\}}\right),\\
\mathfrak{P}_2(Z) &\coloneqq \left(I_{2s^{\{\mathfrak{s}\}}\times 2s^{\{\mathfrak{s}\}}} - Z\otimes \mathcal{D}^{\{\mathfrak{s}\}}\right)^{-1}\left(Z\otimes \mathcal{T}^{\{\mathfrak{s}\}}\right),\\
\widetilde{\mu}\left(z^{\{\mathfrak{f}\}}\right) &\coloneqq \sum_{k\geq 0}\operatorname{diag}\left(\frac{\Delta c^{\{\mathfrak{s}\}}\times \varphi_{k+2}\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)}{k+1}\right)\Gamma^k,\\
\widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) &\coloneqq \sum_{k\geq 0}\operatorname{diag}\left(\frac{\Delta c^{\{\mathfrak{s}\}}\times \varphi_{k+2}\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)}{k+1}\right)\Psi^k.
\end{aligned}
$$

The predictor stages become

$$
\begin{bmatrix} Y^{\{\mathfrak{f}\}*}\\ Y^{\{\mathfrak{s}\}*}\end{bmatrix} = \mathfrak{P}_1(Z)\begin{bmatrix} y_n^{\{\mathfrak{f}\}}\\ y_n^{\{\mathfrak{s}\}}\end{bmatrix} + \mathfrak{P}_2(Z)\begin{bmatrix} Y^{\{\mathfrak{f}\}}\\ Y^{\{\mathfrak{s}\}}\end{bmatrix}.
$$

The fast internal ODEs become

$$
\begin{aligned}
v^{\{\mathfrak{f}\}\prime} = {}& \lambda^{\{\mathfrak{f}\}}\operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right) v^{\{\mathfrak{f}\}} + \eta^{\{\mathfrak{s}\}}\operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right) L\,Y^{\{\mathfrak{f}\}}\\
&+ \eta^{\{\mathfrak{s}\}}\operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right)\widetilde{\Gamma}\left(\tfrac{\theta}{H}\right)\left(w^{\{\mathfrak{f}\}}\,Y^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}\,Y^{\{\mathfrak{s}\}}\right)\\
&+ \eta^{\{\mathfrak{s}\}}\operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\right)\widetilde{\Psi}\left(\tfrac{\theta}{H}\right)\left(w^{\{\mathfrak{f}\}}\,y^{\{\mathfrak{f}\}*} + z^{\{\mathfrak{s}\}}\,y^{\{\mathfrak{s}\}*}\right).
\end{aligned}
$$

The solution to the system of ODEs gives the corrector stages

$$
\begin{aligned}
Y^{\{\mathfrak{f}\}} &= v^{\{\mathfrak{f}\}}(H)\\
&= \operatorname{diag}\left(\varphi_0\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)\right)\left(L\,Y^{\{\mathfrak{f}\}} + y_n^{\{\mathfrak{f}\}}\,e_1\right)\\
&\quad + w^{\{\mathfrak{s}\}}\operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}}\times\varphi_1\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)\right)\left(L\,Y^{\{\mathfrak{s}\}} + y_n^{\{\mathfrak{s}\}}\,e_1\right)\\
&\quad + w^{\{\mathfrak{s}\}}\,\widetilde{\mu}\left(z^{\{\mathfrak{f}\}}\right)\left(w^{\{\mathfrak{f}\}}\,Y^{\{\mathfrak{f}\}} + z^{\{\mathfrak{s}\}}\,Y^{\{\mathfrak{s}\}}\right)\\
&\quad + w^{\{\mathfrak{s}\}}\,\widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right)\left(w^{\{\mathfrak{f}\}}\,y^{\{\mathfrak{f}\}*} + z^{\{\mathfrak{s}\}}\,y^{\{\mathfrak{s}\}*}\right).
\end{aligned}
$$

The combined fast and slow corrector stages are

$$
\begin{aligned}
\begin{bmatrix} Y^{\{\mathfrak{f}\}} \\ Y^{\{\mathfrak{s}\}} \end{bmatrix} &= \begin{bmatrix} \operatorname{diag}\left(\varphi_0\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)\right) L & w^{\{\mathfrak{s}\}} \operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}} \times \varphi_1\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)\right) L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^{\{\mathfrak{f}\}} \\ Y^{\{\mathfrak{s}\}} \end{bmatrix} \\
&\quad + \begin{bmatrix} \varphi_0\left(z^{\{\mathfrak{f}\}}\,\Delta c_1^{\{\mathfrak{s}\}}\right) e_1 & w^{\{\mathfrak{s}\}} \Delta c_1^{\{\mathfrak{s}\}} \varphi_1\left(z^{\{\mathfrak{f}\}} \Delta c_1^{\{\mathfrak{s}\}}\right) e_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_n^{\{\mathfrak{f}\}} \\ y_n^{\{\mathfrak{s}\}} \end{bmatrix} \\
&\quad + \begin{bmatrix} w^{\{\mathfrak{s}\}}\, w^{\{\mathfrak{f}\}}\, \widetilde{\mu}\left(z^{\{\mathfrak{f}\}}\right) & w^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{s}\}}\, \widetilde{\mu}\left(z^{\{\mathfrak{f}\}}\right) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^{\{\mathfrak{f}\}} \\ Y^{\{\mathfrak{s}\}} \end{bmatrix} \\
&\quad + \begin{bmatrix} w^{\{\mathfrak{s}\}}\, w^{\{\mathfrak{f}\}}\, \widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) & w^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{s}\}}\, \widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) \\ 0 & I_{s^{\{\mathfrak{s}\}} \times s^{\{\mathfrak{s}\}}} \end{bmatrix} \begin{bmatrix} y^{\{\mathfrak{f}\}*} \\ y^{\{\mathfrak{s}\}*} \end{bmatrix} \\
&= \mathfrak{N}_1(Z)^{-1} \mathfrak{N}_2(Z) \begin{bmatrix} y_n^{\{\mathfrak{f}\}} \\ y_n^{\{\mathfrak{s}\}} \end{bmatrix}.
\end{aligned}
$$

The stability matrix is given by

$$
\mathbf{M}(Z) \coloneqq \left(I_{2\times 2} \otimes e_{s^{\{\mathfrak{s}\}}}^T\right) \mathfrak{N}_1(Z)^{-1}\, \mathfrak{N}_2(Z), \tag{3.30}
$$

with

$$
\begin{aligned}
\mathfrak{N}_1(Z) &= -\begin{bmatrix} \operatorname{diag}\left(\varphi_0\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)\right) L & w^{\{\mathfrak{s}\}} \operatorname{diag}\left(\Delta c^{\{\mathfrak{s}\}} \times \varphi_1\left(z^{\{\mathfrak{f}\}}\,\Delta c^{\{\mathfrak{s}\}}\right)\right) L \\ 0 & 0 \end{bmatrix} \\
&\quad - \begin{bmatrix} w^{\{\mathfrak{s}\}}\, w^{\{\mathfrak{f}\}}\, \widetilde{\mu}\left(z^{\{\mathfrak{f}\}}\right) & w^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{s}\}}\, \widetilde{\mu}\left(z^{\{\mathfrak{f}\}}\right) \\ 0 & 0 \end{bmatrix} \\
&\quad - \begin{bmatrix} w^{\{\mathfrak{s}\}}\, w^{\{\mathfrak{f}\}}\, \widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) & w^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{s}\}}\, \widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) \\ 0 & I_{s^{\{\mathfrak{s}\}} \times s^{\{\mathfrak{s}\}}} \end{bmatrix} \mathfrak{P}_2 + I_{2s^{\{\mathfrak{s}\}} \times 2s^{\{\mathfrak{s}\}}}, \\
\mathfrak{N}_2(Z) &= \begin{bmatrix} \varphi_0\left(z^{\{\mathfrak{f}\}}\,\Delta c_1^{\{\mathfrak{s}\}}\right) e_1 & w^{\{\mathfrak{s}\}}\, \Delta c_1^{\{\mathfrak{s}\}}\, \varphi_1\left(z^{\{\mathfrak{f}\}}\,\Delta c_1^{\{\mathfrak{s}\}}\right) e_1 \\ 0 & 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} w^{\{\mathfrak{s}\}}\, w^{\{\mathfrak{f}\}}\, \widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) & w^{\{\mathfrak{s}\}}\, z^{\{\mathfrak{s}\}}\, \widetilde{\nu}\left(z^{\{\mathfrak{f}\}}\right) \\ 0 & I_{s^{\{\mathfrak{s}\}} \times s^{\{\mathfrak{s}\}}} \end{bmatrix} \mathfrak{P}_1.
\end{aligned}
$$

3.3.4 Construction of Practical Methods

We develop new implicit IPC-MRI-GARK methods up to order four which are presented in appendix C.2. The second order base methods are reused from SPC-MRI-GARK, but the third and fourth order base methods are custom due to the nondecreasing abscissae constraint. Upon deriving a parameterized family of Γ and Ψ coefficients that satisfy the coupling order conditions, we use free coefficients to satisfy the stability simplifying assumption (3.29). Any remaining parameters are used to optimize the size of the stability region. Plots of the scalar and matrix stability regions are provided in figs. C.3 and C.4, respectively. Compared to SPC-MRI-GARK methods, we found it significantly more challenging to achieve large stability regions at high orders.

3.4 Numerical Results

In this section, we present the numerical tests performed on the SPC-MRI-GARK and IPC-MRI-GARK methods.

3.4.1 Additive Partitioning: the Gray–Scott Model

The first test problem considered is the Gray–Scott reaction-diffusion PDE [108]:

$$\underbrace{\begin{bmatrix} u \\ v \end{bmatrix}'}_{y'} = \underbrace{\begin{bmatrix} \nabla \cdot (\varepsilon_u \, \nabla u) \\ \nabla \cdot (\varepsilon_v \, \nabla v) \end{bmatrix}}_{f^{\{\mathfrak{s}\}}(y)} + \underbrace{\begin{bmatrix} -u\, v^2 + \mathfrak{f}\,(1-u) \\ u\, v^2 - (\mathfrak{f} + \mathfrak{k})\, v \end{bmatrix}}_{f^{\{\mathfrak{f}\}}(y)}. \tag{3.31}$$

It is solved over the 2D spatial domain $[0,1] \times [0,1]$, which is discretized with second order finite differences. The timespan is taken to be $[0, 30]$, and the model parameters are $\varepsilon_u = 0.0625$, $\varepsilon_v = 0.0312$, $\mathfrak{k} = 0.0520$, and $\mathfrak{f} = 0.0180$. The linear diffusions terms of (3.31) make up the slow partition while the nonlinear reaction terms make up the fast partition.

MATLAB is used to carry out the convergence experiments. ODEs that appear within the integrators are solved using `ode45` with the tolerances `abstol = reltol = 1e-10`. Convergence diagrams for the new methods presented in appendix C are shown in fig. 3.2. The numerical orders of accuracy are consistent with theoretical orders.

3.4.2 Component Partitioning: the KPR problem

For a component partitioned test problem of the form (3.8), we use the KPR system [43] as a multi-scale extension to the scalar Prothero-Robinson [16, 72, 111] problem. We define the system as:

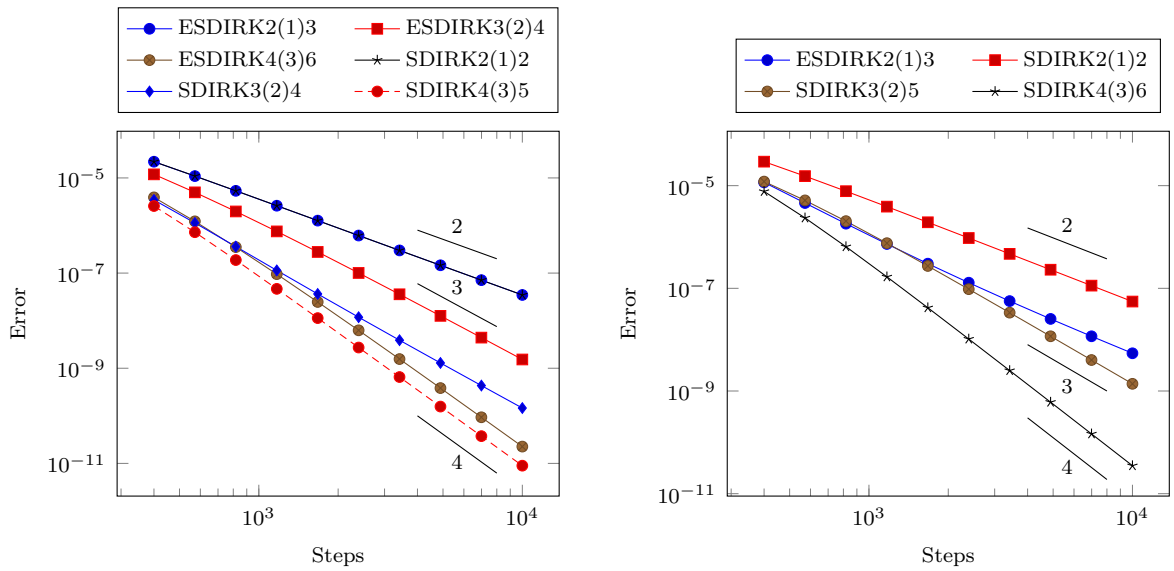
$$\begin{bmatrix} y^{\{\mathfrak{f}\}} \\ y^{\{\mathfrak{s}\}} \end{bmatrix}' = \boldsymbol{\Omega} \cdot \begin{bmatrix} \frac{-3 + y^{\{\mathfrak{f}\}\times 2} - \cos(\omega\, t)}{2\, y^{\{\mathfrak{f}\}}} \\ \frac{-2 + y^{\{\mathfrak{s}\}\times 2} - \cos(t)}{2\, y^{\{\mathfrak{s}\}}} \end{bmatrix} - \begin{bmatrix} \frac{\omega \sin(\omega\, t)}{2\, y^{\{\mathfrak{f}\}}} \\ \frac{\sin(t)}{2\, y^{\{\mathfrak{s}\}}} \end{bmatrix}. \tag{3.32a}$$

The parameters are chosen as $\lambda^{\{\mathfrak{f}\}} = -10$, $\lambda^{\{\mathfrak{s}\}} = -1$, $\xi = 0.1$, $\alpha = 1$, and $\omega = 20$. The exact solution of (3.32a) is given by:

$$y^{\{\mathfrak{f}\}}(t) = \sqrt{3 + \cos(\omega\, t)}, \qquad y^{\{\mathfrak{s}\}}(t) = \sqrt{2 + \cos(t)}. \tag{3.32b}$$

The tests are performed from $t = 0$ to $t = 5\pi/2$ with the initial condition coming from evaluating (3.32b) at $t = 0$. From the exact solution we can also see that the differences in the fast and slow time scales are driven by ω and not $\lambda^{\{\mathfrak{f}\}}$ and $\lambda^{\{\mathfrak{s}\}}$.

The fast integration (3.22b) is also carried out using `ode45` solver with `abstol = reltol = 1e-10`. The convergence diagrams reported in fig. 3.3 indicate that the methods perform at their theoretical orders for this problem.



(a) SPC-MRI-GARK methods (b) IPC-MRI-GARK methods

Figure 3.2: Error vs. number of steps for the Gray–Scott problem (3.31). Reference lines are used to indicate orders.

3.4.3 Multirate Performance: the Inverter Chain Problem

We also consider the inverter chain model of [94] given by the equations

$$\begin{aligned} U_1' &= U_{op} - U_1 - \Gamma\, g(U_{in}, U_1, U_0), \\ U_i' &= U_{op} - U_i - \Gamma\, g(U_{i-1}, U_i, U_0), \qquad i = 2, \ldots m, \end{aligned} \tag{3.33}$$

with $U_0 = 0$, $U_{op} = 5$, $U_T = 1$, $\Gamma = 100$, and

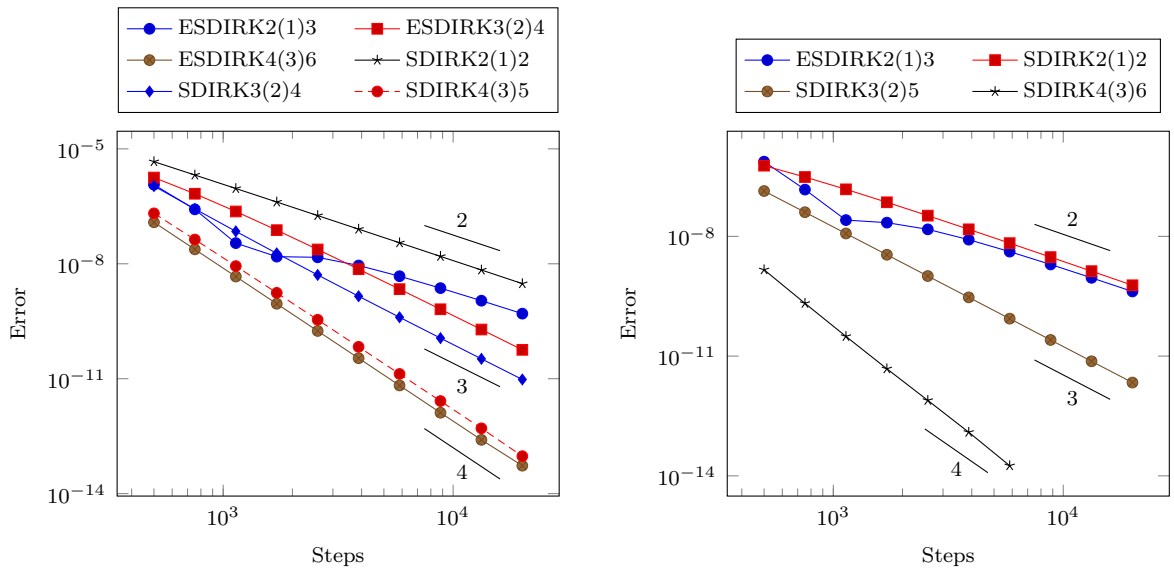
$$g(U_G, U_D, U_S) = (\max(U_G - U_S - U_T, 0))^2 - (\max(U_G - U_D - U_T, 0))^2.$$

The initial conditions of the system are

$$U_i(0) = \begin{cases} 6.246 \times 10^{-3} & i \text{ even} \\ 5 & i \text{ odd} \end{cases},$$

and the input signal is taken to be

$$U_{in}(t) = \begin{cases} t - 5 & 5 \le t \le 10 \\ 5 & 10 \le t \le 15 \\ \frac{5}{2}(17 - t) & 15 \le t \le 17 \\ 0 & \text{otherwise} \end{cases}.$$



(a) SPC-MRI-GARK methods (b) IPC-MRI-GARK methods

Figure 3.3: Error vs. number of steps for the KPR problem (3.32). Reference lines are used to indicate orders.

For the numerical experiments, we use $m = 500$ and a timespan of $[0, 100]$. As the signal propagates through the circuit, only a small percentage of the inverters experience a change in voltage while the other inverters maintain a constant voltage. A componentwise partitioning of (3.33) is used where the fast components come from a sliding window that follows the signal, and the remaining components form the slow partition.

A C implementation of (3.33) is used to measure the performance gains provided by SPC-MRI-GARK and IPC-MRI-GARK over a single rate base method of the same order. For order two, we compare SPC SDIRK2(1)2 from appendix C.1.1 and IPC SDIRK2(1)2 from appendix C.2.1 to their shared base method SDIRK2(1)2. The results are plotted in fig. 3.4a. Figure 3.4b compares SPC SDIRK3(2)4 from appendix C.1.3 to its base method SDIRK3(2)4 and to IPC SDIRK3(2)5 from appendix C.2.3. Finally, fig. 3.4c compares SPC ESDIRK4(3)6 from appendix C.1.6 to its base method ESDIRK4(3)6 and to SPC SDIRK4(3)5 from appendix C.1.5. We did not include results for IPC SDIRK4(3)6 as it was only stable for timesteps much smaller than those used for the SPC methods. This observation is consistent with the stability regions presented in figs. C.3 and C.4.

Fixed timesteps were used for the coupled MRI-GARK methods, as well as for the method ESDIRK5(4)7[2]SA$_2$ from [87] used to solve the internal ODEs. In the experiments, ten timesteps were taken to solve these internal ODEs, except for IPC SDIRK2(1)2 and SPC SDIRK3(2)4 where five and 15 steps were used, respectively.

In all of the performance results presented in fig. 3.4, the multirate methods are able to

achieve a desired accuracy in significantly less time than the single rate schemes. The best results occurred at order two where the speedup ranged from 8 to 60. This can be attributed to the excellent multirate characteristics of the inverter chain problem as well as the flexibility of multirate infinitesimal methods to use any method to solve the modified fast ODEs.

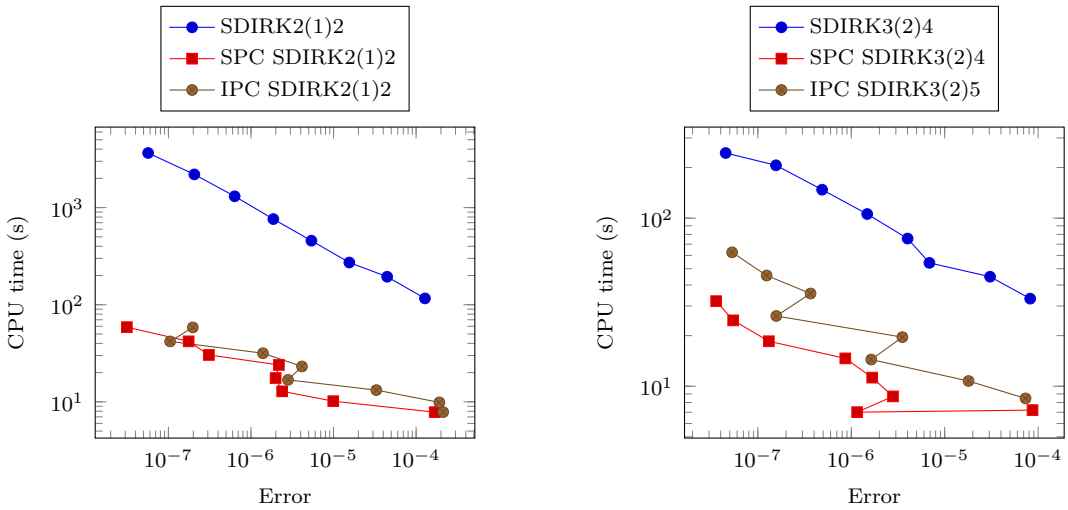
3.5 Conclusions and Future Work

This work extends the class of multirate infinitesimal GARK schemes developed in [129] to include coupled methods. Such methods compute (some of) the stages by solving implicit systems that involve both the fast and the slow components, which gives their "coupled" character. The coupled approach allows us to construct multirate infinitesimal schemes with improved stability for stiff systems with multiple scales, at the additional cost of solving more complex, or larger, nonlinear systems.

Two approaches to formulating the coupling are studied herein. Both of them employ a predictor-corrector structure. The first approach, named step predictor-corrector MRI-GARK, starts with computing all predictor stages in a coupled fashion. The predicted stages are then used to formulate a modified fast ODE, and a single infinitesimal integration is carried out to correct the fast component of the system. The second approach, named internal stage predictor-corrector MRI-GARK, alternates prediction and correction stages. Specifically, each discrete predictor stage is followed by a corrector stage, which integrates a modified fast ODE system and corrects the fast components of that stage.

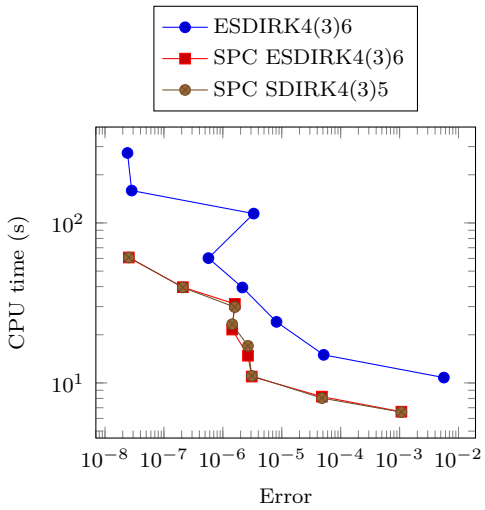
Elegant formulations of the order conditions for both families of methods are developed, and stability requirements for practical methods are analyzed. Methods of order up to four are constructed. Numerical tests verify the orders of convergence on additive and component partitioned cases. Finally, we demonstrate computational efficiency of these multirate methods when compared to their single rate counterparts. Our numerical experiments indicate a performance edge for both MRI-GARK strategies compared to single rate ones, with SPC-MRI-GARK methods having slightly better performance in general. Our analysis also shows larger stability regions for SPC-MRI-GARK methods compared to IPC-MRI-GARK.

The succession of discrete and infinitesimal integration stages in the IPC-MRI-GARK schemes requires non-decreasing abscissae for the slow base method. This requirement, in conjunction with stability, can become difficult to satisfy for high order methods. One solution to alleviate this restriction is to construct MRI-GARK methods that compute their own initial conditions for the infinitesimal stage integration. The authors plan to study these extensions in future works.



(a) Second order methods

(b) Third order methods



(c) Fourth order methods

Figure 3.4: Work precision diagrams for single rate, SPC-MRI-GARK, and IPC-MRI-GARK methods applied to the inverter chain problem (3.33).

Chapter 4

Parallel Implicit-Explicit General Linear Methods

Material from: Steven Roberts, Arash Sarshar, and Adrian Sandu. Parallel implicit-explicit general linear methods. *Communications on Applied Mathematics and Computation*, 2020. doi:10.1007/s42967-020-00083-5

4.1 Introduction

In this work, we consider the autonomous, additively partitioned system of ordinary differential equations (ODEs)

$$y' = f(y) + g(y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq t_F, \tag{4.1}$$

where f is nonstiff, g is stiff, and $y \in \mathbb{R}^d$. Such systems frequently arise from applying the methods of lines to semidiscretize a partial differential equation (PDE). For example, processes such as diffusion, advection, and reaction all have different stiffnesses, CFL conditions, and optimal integration schemes. Implicit-explicit (IMEX) methods offer a specialized approach for solving (4.1) by treating f with an inexpensive explicit method and limiting the application of an implicit method, which is generally more expensive, to g.

The IMEX strategy has a relatively long history in the context of Runge–Kutta methods [14, 24, 42, 85, 105] and linear multistep methods [13, 56, 79]. Zhang, Zharovski, and Sandu proposed IMEX schemes based on two-step Runge–Kutta (TSRK) and General Linear Methods (GLM) [163, 164, 167] with further developments reported in [25, 26, 34, 35, 36, 81, 83, 165]. Similarly, Peer methods, a subclass of GLMs, have been utilized for IMEX integration in the literature such as [48, 96, 145, 149].

High-order IMEX GLMs do not face the stability barriers that constrain multistep counterparts and have much simpler order conditions than IMEX Runge–Kutta methods. Moreover, they can attain high stage order making them resilient to the order reduction phenomena seen in very stiff problems and PDEs with time-dependent boundary conditions.

A major challenge when deriving high-order IMEX GLMs is ensuring the stability region is large enough to be competitive with IMEX Runge–Kutta schemes. One can directly optimize

for the area of the stability region under the constraints of the order conditions, but this is quite challenging as the objective and constraint functions are highly nonlinear and expensive to evaluate. In addition, this optimization is not scalable, with sixth order appearing to be the highest order achieved with this strategy [83].

Parallelism for IMEX schemes is scarcely explored [40, 48], but it is well-studied for traditional, unpartitioned GLMs [30, 31, 32, 82]. One step of a GLM is:

$$
\begin{aligned}
Y_i &= h \sum_{j=1}^{s} a_{i,j} \left(f(Y_j) + g(Y_j) \right) + \sum_{j=1}^{r} u_{i,j}\, y_j^{[n-1]}, \qquad i = 1, \ldots, s, \\
y_i^{[n]} &= h \sum_{j=1}^{s} b_{i,j} \left(f(Y_j) + g(Y_j) \right) + \sum_{j=1}^{r} v_{i,j}\, y_j^{[n-1]}, \qquad i = 1, \ldots, r.
\end{aligned}
$$

Methods are frequently categorized into one of four types to characterize suitability for stiff problems and parallelism [29]. Type 1 and 2 are serial and have the structure

$$
\mathbf{A} = \begin{bmatrix} \lambda & & & \\ a_{2,1} & \lambda & & \\ \vdots & \vdots & \ddots & \\ a_{s,1} & a_{s,2} & \cdots & \lambda \end{bmatrix}. \tag{4.2}
$$

When $\lambda = 0$, the method is of type 1, and of type 2 for $\lambda > 0$. Of interest to this paper are methods of type 3 and 4, which have $\mathbf{A} = \lambda\, I_{s \times s}$ so that all internal stages are independent and can be computed in parallel. Type 3 methods are explicit with $\lambda = 0$, while type 4 methods are implicit with $\lambda > 0$.

This work extends traditional, parallel GLMs into the IMEX setting and proposes two systematic approaches for designing stable methods of arbitrary order. The first uses the popular DIMSIM framework for the base methods. In particular, we use a family of type 4 methods proposed by Butcher [31] for the implicit base and show an explicit counterpart is uniquely determined. This eliminates the need to perform a sophisticated optimization procedure to determine coefficients. The second approach can be interpreted as a generalization of the simplest IMEX scheme: IMEX Euler. It starts with an ensemble of states each approximating the ODE solution at different points in time. In parallel, they are propagated one timestep forward using IMEX Euler, which is only first order accurate. A new, highly-accurate ensemble of states is computed by taking linear combinations of the IMEX Euler solutions. This scheme, which we call *parallel ensemble IMEX Euler*, can be described in the framework of IMEX GLMs. Notably, it maintains the exact same stability region and roughly the same runtime in a parallel setting as IMEX Euler while achieving arbitrarily high orders of consistency. Again, coefficients are determined uniquely, and we show they are very simple to compute using basic matrix operations.

To assess the quality of the two new families of parallel IMEX GLMs, we apply them to a PDE with time-dependent forcing and boundary conditions, as well as to a singularly perturbed

PDE. Convergence is verified as high as eighth order for these challenging problems which can cause order reduction for methods of low stage order. For the performance tests, the parallel methods were run on several nodes in a cluster using MPI and compared to existing, high-quality, serial IMEX Runge–Kutta and IMEX GLMs run on a single node. The best parallel methods could reach a desired solution accuracy approximately two to four times faster.

The structure of this paper is as follows. Section 4.2 reviews the formulation, order conditions, and stability analysis of IMEX GLMs. This is then specialized in section 4.3 for parallel IMEX GLMs. Sections 4.4 and 4.5 present and analyze two new families of parallel IMEX GLMs. The convergence and performance of these new schemes is compared to other IMEX GLMs and IMEX Runge–Kutta methods in section 4.6. We summarize our findings and provide final remarks in section 4.7.

4.2 Background on IMEX GLMs

An IMEX GLM [164] computes s internal and r external stages using timestep h according to:

$$Y_i = h \sum_{j=1}^{i-1} a_{i,j}\, f(Y_j) + h \sum_{j=1}^{i} \widehat{a}_{i,j}\, g(Y_j) + \sum_{j=1}^{r} u_{i,j}\, y_j^{[n-1]}, \qquad i = 1, \dots, s, \tag{4.3a}$$

$$y_i^{[n]} = h \sum_{j=1}^{s} \Big(b_{i,j}\, f(Y_j) + \widehat{b}_{i,j}\, g(Y_j) \Big) + \sum_{j=1}^{r} v_{i,j}\, y_j^{[n-1]}, \qquad i = 1, \dots, r. \tag{4.3b}$$

Using the matrix notation for the coefficients

$$\mathbf{A} := (a_{i,j}), \quad \mathbf{B} := (b_{i,j}), \quad \mathbf{U} := (u_{i,j}), \quad \widehat{\mathbf{A}} := (\hat{a}_{i,j}), \quad \widehat{\mathbf{B}} := (\hat{b}_{i,j}) \quad \mathbf{V} := (v_{i,j}),$$

the IMEX GLM can be represented in the Butcher tableau

$$\begin{array}{c|c|c|c} \mathbf{c} & \mathbf{A} & \widehat{\mathbf{A}} & \mathbf{U} \\ \hline & \mathbf{B} & \widehat{\mathbf{B}} & \mathbf{V} \end{array}. \tag{4.4}$$

Assuming the incoming external stages to a step satisfy

$$\begin{aligned} y_i^{[n-1]} = w_{i,0}\, y(t_{n-1}) &+ \sum_{k=1}^{p} w_{i,k}\, h^k \, \frac{\mathrm{d}^{k-1} f(y(t))}{\mathrm{d}t^{k-1}}(t_{n-1}) \\ &+ \sum_{k=1}^{p} \widehat{w}_{i,k}\, h^k \, \frac{\mathrm{d}^{k-1} g(y(t))}{\mathrm{d}t^{k-1}}(t_{n-1}) + \mathcal{O}\big(h^{p+1}\big), \qquad i = 1, \dots, r, \end{aligned} \tag{4.5}$$

an IMEX GLM is said to have *stage order* q if

$$Y_i = y(t_{n-1} + c_i\, h) + \mathcal{O}\big(h^{q+1}\big), \qquad i = 1, \dots, s,$$

and *order* p if

$$\begin{aligned} y_i^{[n]} &= w_{i,0}\, y(t_n) + \sum_{k=1}^{p} w_{i,k}\, h^k\, \frac{\mathrm{d}^{k-1} f(y(t))}{\mathrm{d}t^{k-1}}(t_n) \\ &\quad + \sum_{k=1}^{p} \widehat{w}_{i,k}\, h^k\, \frac{\mathrm{d}^{k-1} g(y(t))}{\mathrm{d}t^{k-1}}(t_n) + \mathcal{O}\big(h^{p+1}\big), \qquad i = 1, \dots, r. \end{aligned}$$

The Taylor series weights for the external stages are also described in the matrix form

$$\mathbf{W} = (w_{i,j}) \in \mathbb{R}^{r \times (p+1)}, \qquad \widehat{\mathbf{W}} = (\widehat{w}_{i,j}) \in \mathbb{R}^{r \times (p+1)}, \tag{4.6}$$

with $w_{i,0} = \widehat{w}_{i,0}$ for $i = 1, \dots, r$.

The order conditions for IMEX GLMs are discussed in detail in [164]. Notably, a preconsistent IMEX GLM has order p and stage order $q \in \{p, p-1\}$ if and only if the base methods have order p and stage order $q \in \{p, p-1\}$. Here, we present the order conditions in a compact matrix form. First, we define the Toeplitz matrices

$$\mathbf{K}_n = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}, \qquad \mathbf{E}_n = \exp(\mathbf{K}_n) = \begin{bmatrix} 1 & 1 & \frac{1}{2} & \cdots & \frac{1}{(n-1)!} \\ & 1 & 1 & \cdots & \frac{1}{(n-2)!} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

and scaled Vandermonde matrix

$$\mathbf{C}_n = \begin{bmatrix} \mathbb{1}_s & \mathbf{c} & \frac{\mathbf{c}^2}{2} & \cdots & \frac{\mathbf{c}^{n-1}}{(n-1)!} \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{4.7}$$

Powers of a vector are understood to be component-wise, and $\mathbb{1}_s$ represents the vector of ones of dimension s.

Theorem 4.1 (Compact IMEX GLM order conditions [164])**.** *Assume $y^{[n-1]}$ satisfies (4.5). The IMEX GLM (4.3) has order p and stage order $q \in \{p, p-1\}$ if and only if*

$$\mathbf{C}_{q+1} - \mathbf{A}\, \mathbf{C}_{q+1}\, \mathbf{K}_{q+1} - \mathbf{U}\, \mathbf{W}_{:,0:q} = 0_{s \times (q+1)}, \tag{4.8a}$$

$$\mathbf{C}_{q+1} - \widehat{\mathbf{A}}\, \mathbf{C}_{q+1}\, \mathbf{K}_{q+1} - \mathbf{U}\, \widehat{\mathbf{W}}_{:,0:q} = 0_{s \times (q+1)}, \tag{4.8b}$$

$$\mathbf{W}\, \mathbf{E}_{p+1} - \mathbf{B}\, \mathbf{C}_{p+1}\, \mathbf{K}_{p+1} - \mathbf{V}\, \mathbf{W} = 0_{r \times (p+1)}, \tag{4.8c}$$

$$\widehat{\mathbf{W}}\, \mathbf{E}_{p+1} - \widehat{\mathbf{B}}\, \mathbf{C}_{p+1}\, \mathbf{K}_{p+1} - \mathbf{V}\, \widehat{\mathbf{W}} = 0_{r \times (p+1)}, \tag{4.8d}$$

where $\mathbf{W}_{:,0:q}$ is the first $q+1$ columns of $\mathbf{W}$, and $\widehat{\mathbf{W}}_{:,0:q}$ is defined analogously.

Remark 4.2. The first column in each of the matrix conditions in (4.8) corresponds to a preconsistency condition.

4.2.1 Linear Stability of IMEX GLMs

The standard test problem used to analyze the linear stability of an IMEX method is the partitioned problem

$$y' = \xi\, y + \widehat{\xi}\, y, \tag{4.9}$$

where $\xi\, y$ is considered nonstiff and $\widehat{\xi}\, y$ is considered stiff. Applying the IMEX GLM (4.3) to (4.9) yields the stability matrix

$$\begin{aligned} y^{[n]} &= \mathbf{M}(w, \widehat{w})\, y^{[n-1]}, \\ \mathbf{M}(w, \widehat{w}) &= \mathbf{V} + \left(w\, \mathbf{B} + \widehat{w}\, \widehat{\mathbf{B}}\right) \left(I_{s\times s} - w\, \mathbf{A} - \widehat{w}\, \widehat{\mathbf{A}}\right)^{-1} \mathbf{U}, \end{aligned}$$

where $w = h\,\xi$ and $\widehat{w} = h\,\widehat{\xi}$. The set of $(w, \widehat{w}) \in \mathbb{C} \times \mathbb{C}$ for which $\mathbf{M}(w, \widehat{w})$ is power bounded, and thus the IMEX GLM is stable, is a four-dimensional region that can be difficult to analyze and visualize. Following [164], we also consider the simpler stability regions

$$\widehat{\mathcal{S}}_\alpha = \left\{ \widehat{w} \in \widehat{\mathcal{S}} \,:\, |\mathrm{Im}(\widehat{w})| < \tan(\alpha)\, |\mathrm{Re}(\widehat{w})| \right\}, \tag{4.10a}$$

$$\mathcal{S}_\alpha = \left\{ w \in \mathcal{S} \,:\, \mathbf{M}(w, \widehat{w}) \text{ power bounded } \forall \widehat{w} \in \widehat{\mathcal{S}}_\alpha \right\}, \tag{4.10b}$$

where $\mathcal{S}$ and $\widehat{\mathcal{S}}$ are the stability regions of the explicit and implicit base methods, respectively. Equation (4.10a) is referred to as the *desired stiff stability region* and (4.10b) as the *constrained nonstiff stability region*.

4.3 Parallel IMEX GLMs

An IMEX GLM formed by pairing a type 3 GLM with a type 4 GLM has stages of the form

$$Y_i = h\,\lambda\, g(Y_i) + \sum_{j=1}^{r} u_{i,j}\, y_j^{[n-1]}, \qquad i = 1, \ldots, s, \tag{4.11a}$$

$$y_i^{[n]} = h \sum_{j=1}^{s} \left(b_{i,j}\, f(Y_j) + \widehat{b}_{i,j}\, g(Y_j) \right) + \sum_{j=1}^{r} v_{i,j}\, y_j^{[n-1]}, \qquad i = 1, \ldots, r. \tag{4.11b}$$

The only shared dependencies among the internal stages are the previously computed external stages $y_j^{[n-1]}$. This allows the IMEX method to inherit the parallelism of the base methods.

The tableau for a parallel IMEX GLM is of the form

$$\begin{array}{c|c|c|c} \mathbf{c} & 0_s & \lambda\, I_{s\times s} & \mathbf{U} \\ \hline & \mathbf{B} & \widehat{\mathbf{B}} & \mathbf{V} \end{array}. \tag{4.12}$$

We note that one could more generally define $\widehat{\mathbf{A}} = \operatorname{diag}(\lambda_1, \ldots, \lambda_s)$, however, this introduces additional complexity and degrees of freedom that are not needed for the purposes of this paper.

4.3.1 Simplified order conditions

In this paper, we will consider methods with $p = q = r = s$, distinct $\mathbf{c}$ values (nonconfluent method), and an invertible $\mathbf{U}$. By transforming the base methods into an equivalent formulation, we can then assume without loss of generality that $\mathbf{U} = I_{s\times s}$. With these assumptions, we start by determining the structure of the external stage weights $\mathbf{W}$ and $\widehat{\mathbf{W}}$.

Lemma 4.3. *For a parallel IMEX GLM with $\mathbf{U} = I_{s\times s}$ and $p = q$, the internal stage order conditions (4.8a) and (4.8b) are equivalent to*

$$\mathbf{W} = \mathbf{C}_{p+1}, \quad \textit{and} \quad \widehat{\mathbf{W}} = \mathbf{C}_{p+1} - \lambda\, \mathbf{C}_{p+1}\, \mathbf{K}_{p+1}, \tag{4.13}$$

respectively.

Proof. This follows directly from substituting $\mathbf{A} = 0_s$, $\widehat{\mathbf{A}} = \lambda\, I_{s\times s}$, and $\mathbf{U} = I_{s\times s}$ into (4.8a) and (4.8b). □

Our main theoretical result on parallel IMEX GLMs is presented in proposition 4.4 and provides a practical strategy for method derivation.

Theorem 4.4 (Parallel IMEX GLM order conditions)**.** *Consider a nonconfluent parallel IMEX GLM with $\mathbf{U} = I_{s\times s}$. All of the following are equivalent:*

1. *The method satisfies $p = q = r = s$.*
2. *The explicit base method satisfies $p = q = r = s$ and*

$$\widehat{\mathbf{W}} = \mathbf{C}_{s+1} - \lambda\, \mathbf{C}_{s+1}\, \mathbf{K}_{s+1}, \tag{4.14a}$$
$$\widehat{\mathbf{B}} = \mathbf{B} - \lambda\, \mathbf{C}_s\, \mathbf{E}_s\, \mathbf{C}_s^{-1} + \lambda\, \mathbf{V}. \tag{4.14b}$$

3. *The implicit base method satisfies $p = q = r = s$ and*

$$\mathbf{W} = \mathbf{C}_{s+1}, \tag{4.15a}$$
$$\mathbf{B} = \widehat{\mathbf{B}} + \lambda\, \mathbf{C}_s\, \mathbf{E}_s\, \mathbf{C}_s^{-1} - \lambda\, \mathbf{V}. \tag{4.15b}$$

Remark 4.5. With proposition 4.4, once the implicit base method has been chosen, *all* coefficients for the explicit counterpart are uniquely determined by the order conditions. Conversely, if the explicit base is fixed, then all implicit method coefficients are uniquely determined, but parameterized by λ.

Proof. To start, we will show the first statement of proposition 4.4 is equivalent to the second. Assume a nonconfluent parallel IMEX GLM with $\mathbf{U} = I_{s\times s}$ has $p = q = r = s$. By proposition 4.1, the explicit (and implicit) base method also has $p = q = r = s$ and satisfies the order conditions in (4.8). Further, by proposition 4.3, (4.14a) holds. Subtracting (4.8d) from (4.8c) gives

$$\lambda\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1}\,\mathbf{E}_{s+1} + \left(\widehat{\mathbf{B}} - \mathbf{B}\right)\mathbf{C}_{s+1}\,\mathbf{K}_{s+1} - \lambda\,\mathbf{V}\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1} = 0_{s\times(s+1)}. \tag{4.16}$$

The three terms summed on the left-hand side of (4.16) have zeros in the leftmost column. Removing this yields the following equivalent statement:

$$\lambda\,\mathbf{C}_s\,\mathbf{E}_s + \left(\widehat{\mathbf{B}} - \mathbf{B}\right)\mathbf{C}_s - \lambda\,\mathbf{V}\,\mathbf{C}_s = 0_s.$$

A bit of algebraic manipulation recovers the desired result of (4.14b).

Now assume a nonconfluent parallel IMEX GLM with $\mathbf{U} = I_{s\times s}$ satisfies the properties of the second statement of proposition 4.4. Condition (4.14a) ensures the implicit method has stage order q, and (4.14b) ensure it has order p:

$$\begin{aligned}
&\widehat{\mathbf{W}}\,\mathbf{E}_{s+1} - \widehat{\mathbf{B}}\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1} - \mathbf{V}\,\widehat{\mathbf{W}} \\
&= \widehat{\mathbf{W}}\,\mathbf{E}_{s+1} - \left(\mathbf{B} - \lambda\,\mathbf{C}_s\,\mathbf{E}_s\,\mathbf{C}_s^{-1} + \lambda\,\mathbf{V}\right)\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1} - \mathbf{V}\,\widehat{\mathbf{W}} \\
&= \left(\widehat{\mathbf{W}} - \lambda\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1}\right)\,\mathbf{E}_{s+1} - \mathbf{B}\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1} - \mathbf{V}\left(\widehat{\mathbf{W}} - \lambda\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1}\right) \\
&= \mathbf{W}\,\mathbf{E}_{s+1} - \mathbf{B}\,\mathbf{C}_{s+1}\,\mathbf{K}_{s+1} - \mathbf{V}\,\mathbf{W} \\
&= 0_{s\times(s+1)}.
\end{aligned}$$

Now both base methods have $p = q = r = s$, so by proposition 4.1, the combined IMEX scheme also has $p = q = r = s$.

The process to show statement one is equivalent to statement three, thus completing the proof, follows nearly identical steps, and is therefore omitted. □

4.3.2 Stability

Applying parallel IMEX GLMs to linear stability test (4.9) gives

$$\mathbf{M}(w, \widehat{w}) = \mathbf{V} + \frac{w}{1 - \lambda\,\widehat{w}}\,\mathbf{B}\,\mathbf{U} + \frac{\widehat{w}}{1 - \lambda\,\widehat{w}}\,\widehat{\mathbf{B}}\,\mathbf{U} \tag{4.17a}$$

$$= \mathbf{M}\left(\frac{w}{1 - \lambda\,\widehat{w}}\right) + \widehat{\mathbf{M}}(\widehat{w}) - \mathbf{V}, \tag{4.17b}$$

where $\mathbf{M}(w)$ and $\widehat{\mathbf{M}}(\widehat{w})$ are the stability matrices of the explicit and implicit base methods, respectively. When the implicit partition becomes infinitely stiff,

$$\mathbf{M}(w, \infty) = \widehat{\mathbf{M}}(\infty) = \mathbf{V} - \frac{1}{\lambda}\widehat{\mathbf{B}}\,\mathbf{U}.$$

Stability matrices evaluated at ∞ are understood to be the value in the limit.

4.3.3 Starting Procedure

The starting procedure for nontrivial IMEX GLMs is more complex than traditional GLMs because the external stages for IMEX GLMs weight time derivatives of f and g differently. When computing $y^{[0]}$, the high order time derivatives are usually not readily available, but can be approximated by finite differences [33, 164]. A one-step method can be used to get very accurate approximations to y, and consequently f and g, at a grid of time points around t_0 to construct these finite difference approximations. While this generic approach is applicable to parallel IMEX GLMs, we also describe a specialized strategy that is simpler and more accurate.

Based on the $\mathbf{W}$ and $\widehat{\mathbf{W}}$ weights derived in (4.13),

$$\begin{aligned} y_i^{[0]} &= y(t_0) + \sum_{k=1}^{p} \frac{c_i^k}{k!}\, h^k\, \frac{\mathrm{d}^{k-1} f(y(t))}{\mathrm{d}t^{k-1}} \\ &\quad + \sum_{k=1}^{p} \left(\frac{c_i^k}{k!} - \frac{\lambda\, c_i^{k-1}}{(k-1)!} \right) h^k\, \frac{\mathrm{d}^{k-1} g(y(t))}{\mathrm{d}t^{k-1}} + \mathcal{O}\big(h^{p+1}\big) \\ &= y(t_0 + h\, c_i) - h\, \lambda\, g(y(t_0 + h\, c_i)) + \mathcal{O}\big(h^{p+1}\big). \end{aligned} \tag{4.18}$$

Now, a one-step method can be used to get approximations to y and g at times $t_0 + h\, c_i$ to compute $y^{[0]}$. This eliminates the need to use finite differences and eliminates the error associated with them. Note that negative abscissae would require integrating backwards in time. Although the interval of integration may be quite short, this could still lead to stability issues, and is easily remedied. If $c_{\min}$ is the smallest abscissa, then the one-step method can produce an approximation to $y^{[\ell]}$, where $\ell = \lceil -c_{\min} \rceil$, instead of $y^{[0]}$. Note, $t_\ell + c_i\, h \geq t_0$, and the IMEX GLM will start with $y^{[\ell]}$ to compute $y^{[\ell+1]}$ and so on.

4.3.4 Ending Procedure

We will consider the ending procedure for an IMEX GLM to be of the form

$$y(t_n) \approx h \sum_{j=1}^{s} \Big(\beta_j\, f(Y_j) + \widehat{\beta}_j\, g(Y_j) \Big) + \sum_{j=1}^{r} \gamma_j\, y_j^{[n-1]}. \tag{4.19}$$

Frequently, IMEX GLMs have the last abscissa set to 1, which allows for a particularly simple ending procedure for high stage order methods. The final internal stage Y_s can be used as an $\mathcal{O}\big(h^{\min(p,q+1)}\big)$ accurate approximation to $y(t_n)$. One can easily verify that the coefficients for such an ending procedure are

$$\beta^T = e_s^T\, \mathbf{A}, \quad \widehat{\beta}^T = e_s^T\, \widehat{\mathbf{A}}, \quad \gamma^T = e_s^T\, \mathbf{U}, \tag{4.20}$$

where e_i is the i-th column of $I_{s\times s}$. Indeed, all parallel IMEX GLMs tested in this paper have $c_s = 1$, however, we present an alternative strategy to approximate $y(t_n)$. Suppose a parallel IMEX GLM has $c_i = 0$ for some $i \in \{1, \ldots, s-1\}$ and $c_s = 1$. Then based on the relation in (4.18), we have that

$$\begin{aligned} & h \sum_{j=1}^{s} \Big(b_{i,j}\, f(Y_j) + \widehat{b}_{i,j}\, g(Y_j) \Big) + h\,\lambda\, g(Y_s) + \sum_{j=1}^{r} v_{i,j}\, y_j^{[n-1]} \\ &= y_i^{[n]} + h\,\lambda\, g(Y_s) \\ &= y(t_n) + \mathcal{O}\big(h^{\min(p,q+1)}\big). \end{aligned}$$

This ending procedure has the coefficients

$$\beta^T = e_i^T\, \mathbf{B}, \quad \widehat{\beta}^T = e_i^T\, \widehat{\mathbf{B}} + \lambda\, e_s^T, \quad \gamma^T = e_i^T\, \mathbf{V}. \tag{4.21}$$

For the parallel ensemble IMEX Euler methods of section 4.5, numerical tests revealed this new ending procedure is substantially more accurate. For the parallel IMEX DIMSIMs, the coefficients in (4.20) and (4.21) gave similar results in tests as the accumulated global error dominated the local truncation error of the ending procedure.

4.4 Parallel IMEX DIMSIMs

Diagonally-implicit multi-stage integration methods (DIMSIMs) have become a popular choice of base method to build high-order IMEX GLMs. IMEX DIMSIMs are characterized by the following structural assumptions:

1. $\mathbf{A}$ is strictly lower triangular, and $\widehat{\mathbf{A}}$ is lower triangular with the same element λ on the diagonal as in (4.2).
2. $\mathbf{V}$ is rank one with the single nonzero eigenvalue equal to one to ensure preconsistency.
3. $q \in \{p, p-1\}$ and $r \in \{s, s+1\}$.

Based on proposition 4.1, to build a parallel IMEX DIMSIM with $p = q = r = s$ we only need to choose one of the base methods and the rest of the coefficients will follow. If we start by picking an explicit base, it may be difficult to ensure the resulting implicit method has acceptable stability properties, ideally L-stability. Instead, we start by picking a stable, type 4 DIMSIM for the implicit base method.

In [30, 31], Butcher developed a systematic approach to construct DIMSIMs of type 4 with "perfect damping at infinity." One of his primary results is presented in proposition 4.6.

Theorem 4.6 (Type 4 DIMSIM coefficients [31, Theorem 4.1])**.** *For the type 4 DIMSIM*

$$\begin{array}{c|c|c} \mathbf{c} & \lambda\, I_{s\times s} & I_{s\times s} \\ \hline & \widehat{\mathbf{B}} & \mathbf{V} \end{array}$$

with $p = q = r = s$ and $\mathbf{V}\,\mathbb{1}_s = \mathbb{1}_s$, the transformed coefficients

$$\overline{\mathbf{B}} = \mathbf{T}^{-1}\,\widehat{\mathbf{B}}\,\mathbf{T}, \qquad \overline{\mathbf{V}} = \mathbf{T}^{-1}\,\mathbf{V}\,\mathbf{T},$$

satisfy

$$\overline{\mathbf{V}} e_1 = e_1, \tag{4.22a}$$

$$\overline{\mathbf{B}} = \widehat{\mathbf{E}}_s - \lambda\,\mathbf{E}_s + \overline{\mathbf{V}}\left(\lambda\, I_{s\times s} - \mathbf{K}_s^T\right), \tag{4.22b}$$

where

$$\mathbf{T} = \begin{bmatrix} P^{(s)}(c_1) & P^{(s-1)}(c_1) & \cdots & P'(c_1) \\ P^{(s)}(c_2) & P^{(s-1)}(c_2) & \cdots & P'(c_2) \\ \vdots & \vdots & \ddots & \vdots \\ P^{(s)}(c_s) & P^{(s-1)}(c_s) & \cdots & P'(c_s) \end{bmatrix}, \qquad P(x) = \frac{1}{s!}\prod_{i=1}^{s}(x - c_i),$$

and

$$\widehat{\mathbf{E}}_n = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{6} & \cdots & \frac{1}{(n-1)!} & \frac{1}{n!} \\ 1 & 1 & \frac{1}{2} & \cdots & \frac{1}{(n-2)!} & \frac{1}{(n-1)!} \\ 0 & 1 & 1 & \cdots & \frac{1}{(n-3)!} & \frac{1}{(n-2)!} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \frac{1}{2} \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Proposition 4.6 fully determines the $\widehat{\mathbf{B}}$ coefficient for a type 4 DIMSIM, but $\mathbf{c}$, λ and most of $\mathbf{V}$ remain undetermined. Fortunately, this offers sufficient degrees of freedom to ensure $\widehat{\mathbf{M}}(\infty)$ is nilpotent. In [31, Theorem 5.1], Butcher proves λ must be a solution to

$$L'_{s+1}\left(\frac{s+1}{\lambda}\right) = 0, \tag{4.23a}$$

and

$$\overline{\mathbf{V}} = \begin{bmatrix} v_1 & v_2 & \cdots & v_s \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \qquad v_i = (-1)^{s+1}\frac{s-i+2}{s+1}\lambda^{i-1}L_{s+1}^{(s-i+2)}\left(\frac{s+1}{\lambda}\right). \tag{4.23b}$$

Here, $L_n(x) = \sum_{i=0}^{n}\binom{n}{i}(-x)^i/i!$ is the Laguerre polynomial and $L_n^{(m)}(x)$ is its m-th derivative.

Method order	λ	$c_i = \frac{i-1}{s-1}$	$c_i = 1 - s + i$
2	0.633975	1.38	1.38
3	1.21014	20.38	7.31
4	0.872421	90.86	7.07
5	1.30128	5885.22	29.74
6	1.80569	933038.32	368.93
7	1.35220	10318974.86	303.07
8	1.73680	2557191349.96	3534.00
9	1.38470	41543982719.05	2907.22
10	1.69561	14146161438042.40	41813.39

Table 4.1: Approximate values for the largest coefficient in absolute value from $\mathbf{B}$, $\widehat{\mathbf{B}}$, and $\mathbf{V}$ for parallel IMEX DIMSIMs of orders two to ten.

With the implicit base method determined, we now turn to the explicit method. Indeed, proposition 4.1 could be applied to recover $\mathbf{B}$, but proposition 4.6 provides a more direct approach. Equation (4.22b), which is normally used for type 4 methods, remains valid when $\lambda = 0$, and (4.22a) is fulfilled because the implicit and explicit base methods share $\mathbf{V}$.

In summary, the coefficients for a parallel IMEX DIMSIM with $p = q = r = s$ are given by

$$\begin{aligned}
\mathbf{A} &= 0_s, & \mathbf{B} &= \mathbf{T}\left(\widehat{\mathbf{E}}_s - \lambda\,\mathbf{E}_s + \overline{\mathbf{V}}\left(\lambda\, I_{s\times s} - \mathbf{K}_s^T\right)\right)\mathbf{T}^{-1},\\
\widehat{\mathbf{A}} &= \lambda\, I_{s\times s}, & \widehat{\mathbf{B}} &= \mathbf{T}\left(\widehat{\mathbf{E}}_s - \overline{\mathbf{V}}\,\mathbf{K}_s^T\right)\mathbf{T}^{-1},\\
\mathbf{U} &= I_{s\times s}, & \mathbf{V} &= \mathbf{T}\,\overline{\mathbf{V}}\,\mathbf{T}^{-1},
\end{aligned}$$

with $\mathbf{c}$ remaining as free parameters. The two most "natural" and frequently used choices are $\mathbf{c} = [0, 1/(s-1), 2/(s-2), \ldots, 1]^T$ and $\mathbf{c} = [2-s, 1-s, \ldots, 1]^T$. This presents a tradeoff where the first option has smaller local truncation errors, but the second option results in coefficients that grow slower with order, thus reducing the accumulation of finite precision cancellation errors. Table 4.1 presents the magnitude of these largest coefficients for both strategies.

Before proceeding to the stability analysis, we present two examples of parallel IMEX DIMSIMs. A second order method has the tableau

$$\begin{array}{c|cc|cc|cc}
0 & 0 & 0 & \lambda & 0 & 1 & 0\\
1 & 0 & 0 & 0 & \lambda & 0 & 1\\
\hline
 & \frac{4\lambda-3}{4} & \frac{4\lambda-3}{4} & \frac{(2\lambda+1)(4\lambda-3)}{4} & \frac{-8\lambda^2+10\lambda-3}{4} & \frac{4\lambda-3}{2} & \frac{5-4\lambda}{2}\\
 & \frac{4\lambda-5}{4} & \frac{4\lambda+3}{4} & \frac{8\lambda^2+2\lambda-5}{4} & \frac{-8\lambda^2+6\lambda+3}{4} & \frac{4\lambda-3}{2} & \frac{5-4\lambda}{2}
\end{array},$$

where $\lambda = (3-\sqrt{3})/2$. In a more compact form, a third order method has the coefficients

$$
\begin{aligned}
\mathbf{c} &= \begin{bmatrix} 0 & \frac{1}{2} & 1 \end{bmatrix}^T, \\
\mathbf{B} &= \begin{bmatrix} \frac{6\lambda^2-15\lambda+7}{2} & \frac{6\lambda-5}{3} & -\frac{(3\lambda-2)(6\lambda-13)}{6} \\ \frac{72\lambda^2-180\lambda+89}{24} & \frac{6\lambda-7}{3} & \frac{-24\lambda^2+68\lambda-27}{8} \\ \frac{(3\lambda-4)(6\lambda-7)}{6} & 2\lambda-5 & \frac{-18\lambda^2+51\lambda-7}{6} \end{bmatrix}, \\
\widehat{\mathbf{B}} &= \begin{bmatrix} \frac{72\lambda^3-156\lambda^2+34\lambda+21}{6} & \frac{-72\lambda^3+192\lambda^2-88\lambda-5}{3} & \frac{36\lambda^3-114\lambda^2+80\lambda-13}{3} \\ \frac{288\lambda^3-624\lambda^2+112\lambda+89}{24} & \frac{-72\lambda^3+192\lambda^2-79\lambda-7}{3} & \frac{288\lambda^3-912\lambda^2+592\lambda-81}{24} \\ \frac{2\left(18\lambda^3-39\lambda^2+4\lambda+7\right)}{3} & \frac{-72\lambda^3+192\lambda^2-64\lambda-15}{3} & \frac{72\lambda^3-228\lambda^2+130\lambda-7}{6} \end{bmatrix}, \\
\mathbf{V} &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} \frac{72\lambda^2-174\lambda+79}{6} & -\frac{2\left(36\lambda^2-96\lambda+47\right)}{3} & \frac{72\lambda^2-210\lambda+115}{6} \end{bmatrix}, \\
\lambda &= \frac{2\cos\left(\frac{\pi}{18}\right)\sec\left(\frac{\pi}{9}\right)}{\sqrt{3}} \approx 1.210138312730603.
\end{aligned}
$$

4.4.1 Stability

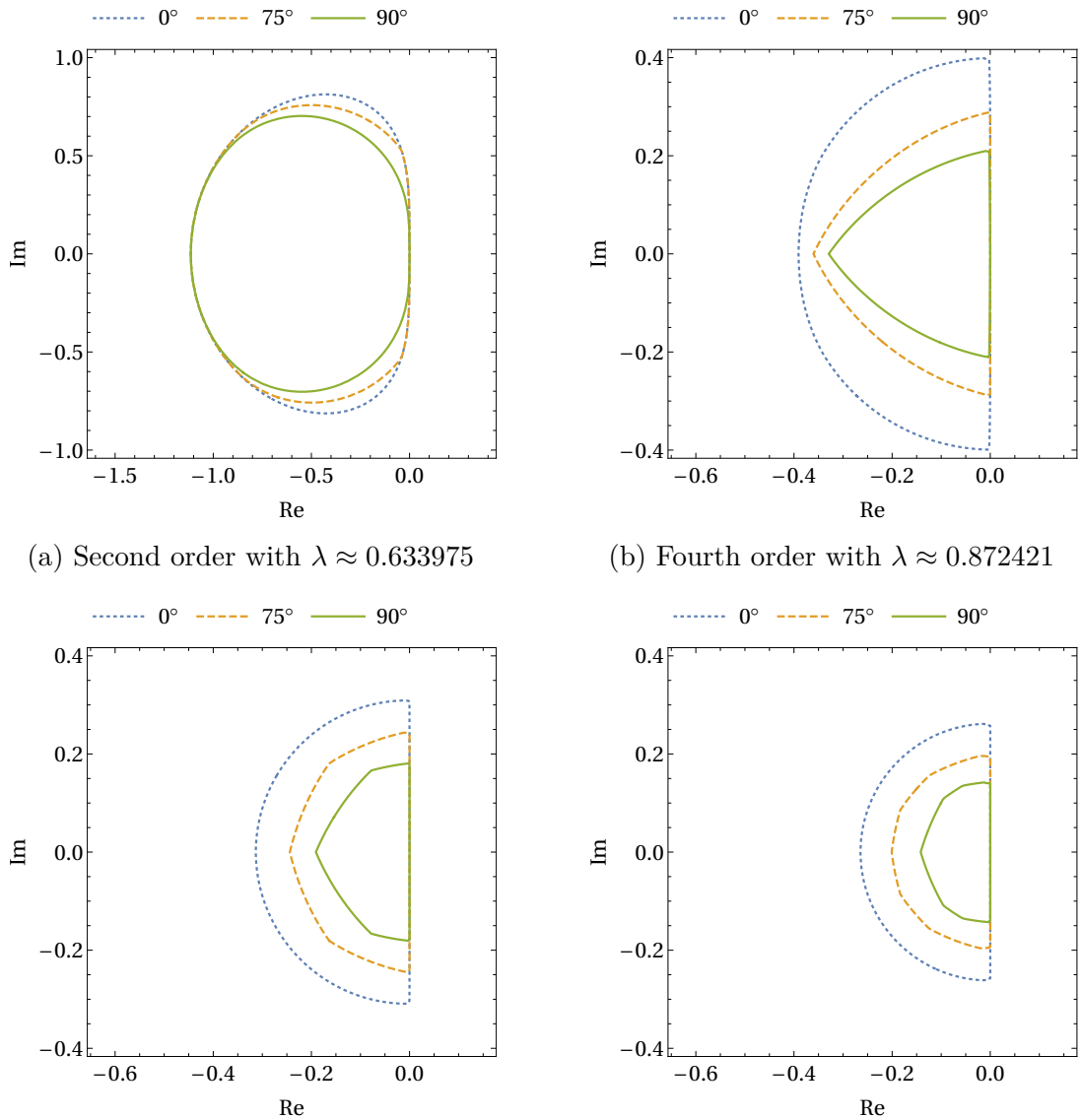
While (4.23) ensures $\rho(\widehat{\mathbf{M}}(\infty)) = 0$, it is not a sufficient condition for L-stability of a type 4 DIMSIM. In [31], appropriate values of λ for L-stability are provided for orders two to ten, excluding nine. If the weaker condition of L(α)-stability is acceptable, smaller values of λ may be used as well.

With $\mathbf{c}$ available as free parameters, it is natural to see if they can be used to optimize the stability of parallel IMEX DIMSIMs. It is easy to verify that stability is, in fact, independent of $\mathbf{c}$:

$$
\begin{aligned}
\mathbf{T}^{-1}\,\mathbf{M}(w,\widehat{w})\,\mathbf{T} = \overline{\mathbf{V}} &+ \frac{w}{1-\lambda\,\widehat{w}}\left(\widehat{\mathbf{E}}_s - \overline{\mathbf{V}}\,\mathbf{K}_s^T\right) \\
&+ \frac{\widehat{w}}{1-\lambda\,\widehat{w}}\left(\widehat{\mathbf{E}}_s - \lambda\,\mathbf{E}_s + \overline{\mathbf{V}}\left(\lambda\,I_{s\times s} - \mathbf{K}_s^T\right)\right).
\end{aligned}
$$

The stability matrix is similar to a matrix completely independent of $\mathbf{c}$; thus $\mathbf{c}$ has no effect on the power boundedness of $\mathbf{M}$.

Plots of the constrained nonstiff stability region for several methods appear in fig. 4.1. Roughly speaking, the area of the stability region shrinks as the order increases. Further, the smaller values of λ satisfying (4.23a) tend to provide larger stability regions for a fixed order.



(a) Second order with $\lambda \approx 0.633975$

(b) Fourth order with $\lambda \approx 0.872421$

(c) Seventh order with $\lambda \approx 1.35220$

(d) Tenth order with $\lambda \approx 1.69561$

Figure 4.1: Stability regions $\mathcal{S}_\alpha$ with $\alpha = 0°, 75°, 90°$ for parallel IMEX DIMSIMs. Note the scale for (a) is different than for the other plots.

4.5 Parallel Ensemble IMEX Euler Methods

If one seeks to minimize communication costs for parallel IMEX GLMs, the choice $\mathbf{U} = \mathbf{V} = I_{s\times s}$ is attractive, as it eliminates the need to share external stages among parallel processes. As we will show in this section, this choice of coefficients also leads to particularly favorable structures for the order conditions and stability matrix.

Theorem 4.7 (Parallel ensemble IMEX Euler order conditions)**.** *A nonconfluent parallel ensemble IMEX Euler method, which starts with the structural assumptions*

$$\mathbf{A} = 0_s, \quad \widehat{\mathbf{A}} = \lambda\, I_{s\times s}, \quad \mathbf{U} = \mathbf{V} = I_{s\times s},$$

has $p = q = r = s$ if and only if the remaining method coefficients are

$$\mathbf{W} = \mathbf{C}_{s+1}, \qquad \widehat{\mathbf{W}} = \mathbf{C}_{s+1} - \lambda\, \mathbf{C}_{s+1}\, \mathbf{K}_{s+1}, \tag{4.24a}$$

$$\mathbf{B} = \mathbf{C}_s\, \mathbf{F}_s\, \mathbf{C}_s^{-1}, \quad \widehat{\mathbf{B}} = \mathbf{C}_s\, \mathbf{F}_s\, (I_{s\times s} - \lambda\, \mathbf{K}_s)\, \mathbf{C}_s^{-1}, \tag{4.24b}$$

where

$$\mathbf{F}_n = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{6} & \cdots & \frac{1}{n!} \\ & 1 & \frac{1}{2} & \cdots & \frac{1}{(n-1)!} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \frac{1}{2} \\ & & & & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{4.25}$$

Remark 4.8. An alternative representation for (4.25) is $\mathbf{F}_n = \varphi_1(\mathbf{K}_n)$, where φ_1 is the entire function

$$\varphi_1(z) = \sum_{k=0}^{\infty} \frac{z^k}{(k+1)!} = \frac{e^z - 1}{z}.$$

Proof. With proposition 4.4, we need only show the explicit base method for parallel ensemble IMEX Euler has $p = q = r = s$ and $\mathbf{B}$ and $\widehat{\mathbf{B}}$ are related by (4.14b). By proposition 4.3, the internal stage order condition for the explicit method, given in (4.8a), holds. For the external stage order conditions:

$$\begin{aligned} \mathbf{W}\, \mathbf{E}_{s+1} - \mathbf{B}\, \mathbf{C}_{s+1}\, \mathbf{K}_{s+1} - \mathbf{V}\, \mathbf{W} &= \mathbf{C}_{s+1}\, \mathbf{E}_{s+1} - \mathbf{C}_s\, \mathbf{F}_s\, \mathbf{C}_s^{-1}\, \mathbf{C}_{s+1}\, \mathbf{K}_{s+1} - \mathbf{C}_{s+1} \\ &= \mathbf{C}_{s+1}\, \mathbf{E}_{s+1} - \mathbf{C}_{s+1}\, \mathbf{F}_{s+1}\, \mathbf{K}_{s+1} - \mathbf{C}_{s+1} \\ &= \mathbf{C}_{s+1}\, \big(\mathbf{E}_{s+1} - \mathbf{F}_{s+1}\, \mathbf{K}_{s+1} - I_{(s+1)\times(s+1)}\big) \\ &= 0_{s\times(s+1)}. \end{aligned}$$

Therefore, the explicit method satisfies all order conditions and has $p = q = r = s$. Finally,

$$\begin{aligned} \mathbf{B} - \lambda\, \mathbf{C}_s\, \mathbf{E}_s\, \mathbf{C}_s^{-1} + \lambda\, \mathbf{V} &= \mathbf{C}_s\, \mathbf{F}_s\, \mathbf{C}_s^{-1} - \lambda\, \mathbf{C}_s\, \mathbf{E}_s\, \mathbf{C}_s^{-1} + \lambda\, I_{s\times s} \\ &= \mathbf{C}_s\, (\mathbf{F}_s - \lambda\, \mathbf{E}_s + \lambda\, I_{s\times s})\, \mathbf{C}_s^{-1} \\ &= \mathbf{C}_s\, (\mathbf{F}_s - \mathbf{F}_s\, \mathbf{K}_s)\, \mathbf{C}_s^{-1} \\ &= \mathbf{C}_s\, \mathbf{F}_s\, (I_{s\times s} - \lambda\, \mathbf{K}_s)\, \mathbf{C}_s^{-1} \\ &= \widehat{\mathbf{B}}, \end{aligned}$$

which completes the proof. □

While the parallel IMEX DIMSIMs of section 4.4 require symbolic tools to derive and have coefficients that can be expressed as roots of polynomials, ensemble methods have simple, rational coefficients that can be derived with basic matrix multiplication. The following parallel ensemble IMEX Euler method, for example, is second order:

$$\begin{array}{c|cc|cc|cc}
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 \\
\hline
 & \frac{1}{2} & \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} & 1 & 0 \\
 & -\frac{1}{2} & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 1
\end{array}.$$

A third order method is given by

$$\begin{array}{c|ccc|ccc|ccc}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{7}{6} & \frac{2}{3} & -\frac{5}{6} & 1 & 0 & 0 \\
 & \frac{1}{6} & -\frac{1}{3} & \frac{7}{6} & -\frac{5}{6} & \frac{11}{3} & -\frac{11}{6} & 0 & 1 & 0 \\
 & \frac{7}{6} & -\frac{10}{3} & \frac{19}{6} & -\frac{11}{6} & \frac{14}{3} & -\frac{11}{6} & 0 & 0 & 1
\end{array},$$

and a fourth order method is given by

$$\begin{array}{c|cccc|cccc|cccc}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
\frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & \frac{9}{8} & \frac{3}{8} & \frac{3}{8} & -\frac{7}{8} & 1 & 0 & 0 & 0 \\
 & -\frac{1}{8} & \frac{5}{8} & -\frac{3}{8} & \frac{7}{8} & \frac{7}{8} & -\frac{19}{8} & \frac{45}{8} & -\frac{25}{8} & 0 & 1 & 0 & 0 \\
 & -\frac{7}{8} & \frac{27}{8} & -\frac{37}{8} & \frac{25}{8} & \frac{25}{8} & -\frac{93}{8} & \frac{131}{8} & -\frac{55}{8} & 0 & 0 & 1 & 0 \\
 & -\frac{25}{8} & \frac{93}{8} & -\frac{123}{8} & \frac{63}{8} & \frac{55}{8} & -\frac{195}{8} & \frac{237}{8} & -\frac{89}{8} & 0 & 0 & 0 & 1
\end{array}.$$

When the order of the method increases, so does the magnitude of the method coefficients: a phenomenon previously described for parallel IMEX DIMSIMs. Similarly, the distribution of abscissae can limit the growth of coefficients, and thus, the floating-point errors associated with them. Table 4.2 lists these maximum coefficients for $\mathbf{c}$'s evenly space between $[0, 1]$, as well as $[2 - s, 1]$.

Method order	$c_i = \frac{i-1}{s-1}$	$c_i = 1 - s + i$
2	1.50	1.50
3	4.67	1.92
4	29.62	3.54
5	203.87	6.37
6	1380.73	13.07
7	9868.32	23.62
8	69256.88	47.97
9	506662.23	87.98
10	3639853.98	177.82

Table 4.2: Approximate values for the largest coefficient in absolute value from $\mathbf{B}$ and $\widehat{\mathbf{B}}$ for parallel ensemble IMEX Euler methods of orders two to ten with $\lambda = 1$.

4.5.1 Stability

An interesting property of parallel ensemble IMEX Euler methods is that $\mathbf{B}$, $\widehat{\mathbf{B}}$, $\mathbf{A}$, and $\widehat{\mathbf{A}}$ all simultaneously triangularize. The stability matrix (4.17) can therefore be put into an upper triangular form with a simple similarity transformation:

$$\mathbf{C}_s^{-1}\,\mathbf{M}(w,\widehat{w})\,\mathbf{C}_s = I_{r\times r} + \frac{w}{1-\lambda\,\widehat{w}}\,\mathbf{F}_s + \frac{\widehat{w}}{1-\lambda\,\widehat{w}}\,\mathbf{F}_s\,(I_{s\times s} - \lambda\,\mathbf{K}_s)\,. \tag{4.26}$$

The diagonal entries of (4.26) are all $1+(w+\widehat{w})/(1-\lambda\,\widehat{w})$ and identically are the eigenvalues of the stability matrix. Note the geometric multiplicity of this repeated eigenvalue is r when $w = \widehat{w} = 0$ and 1 otherwise. In order to ensure L-stability of the implicit base method as well as $\rho(\mathbf{M}(w,\infty)) = 0$, we set $\lambda = 1$. In this case, the eigenvalues simplify to $(1+w)/(1-\widehat{w})$ matching the stability of the IMEX Euler scheme

$$y_n = y_{n-1} + h\,f(t_{n-1}, y_{n-1}) + h\,g(t_n, y_n).$$

There are several other interesting stability features for parallel ensemble IMEX Euler methods. First, stability is independent of the order and the choice of abscissae, allowing a systematic approach to develop stable methods of arbitrary order. The constrained nonstiff stability region has the simple form

$$\mathcal{S}_\alpha = \mathcal{S} = \{w \in \mathbb{C} : |1+w| < 1 \vee w = 0\}\,,$$

when $s > 1$. Except for the origin, the boundary of this circular stability region is carefully excluded because the 1 eigenvalue of $\mathbf{M}$ is defective at those points. This family of methods is stability decoupled in the sense that linear stability of the base methods for their respective partitions implies linear stability of the IMEX scheme.

We note that aside from the origin, $\mathcal{S}_\alpha$ does not contain any of the imaginary axis, indicating potential stability issues when f is oscillatory. This analysis is a bit pessimistic, however,

as $\mathcal{S}_\alpha$ represents the explicit stability when $\widehat{w}$ is chosen in a worst-case scenario. Only when $\widehat{w} = 0$ is there instability for all purely imaginary w. As the modulus of $\widehat{w}$ grows, the range of imaginary w for which the IMEX method is stable also grows.

4.6 Numerical Experiments

We provide numerical experiments to confirm the order of convergence and to study the performance of our methods compared to other IMEX methods. We use the CUSP and Allen–Cahn problems in our experiments.

4.6.1 CUSP Problem

The CUSP problem [72, Chapter IV.10] is associated with the equations

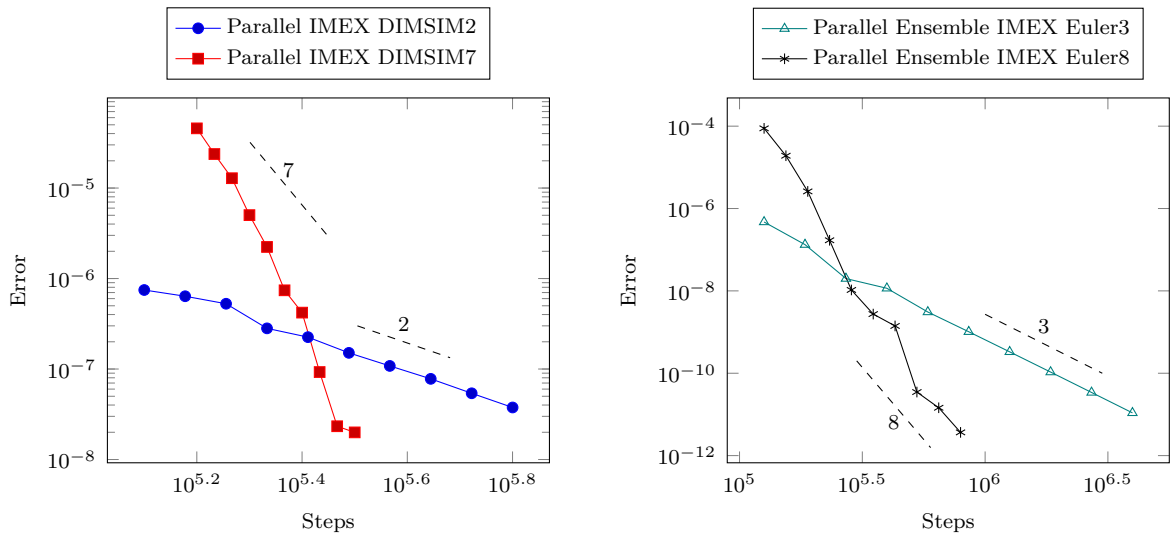
$$\begin{aligned}
\frac{\partial y}{\partial t} &= -\frac{1}{\varepsilon}\left(y^3 + a\,y + b\right) + \sigma\,\frac{\partial^2 y}{\partial x^2},\\
\frac{\partial a}{\partial t} &= b + 0.07\,v + \sigma\,\frac{\partial^2 a}{\partial x^2},\\
\frac{\partial b}{\partial t} &= b\,(1 - a^2) - a - 0.4\,y + 0.035\,v + \sigma\,\frac{\partial^2 b}{\partial x^2},
\end{aligned} \tag{4.27}$$

where $v = \frac{u}{u+0.1}$ and $u = (y - 0.7)\,(y - 1.3)$. The timespan is $t \in [0, 1.1]$, the spatial domain is $x \in [0, 1]$, and the parameters are chosen as $\sigma = \frac{1}{144}$ and $\varepsilon = 10^{-4}$. Spatial derivatives are discretized using second order central finite differences on a uniform mesh with $N = 32$ points and periodic boundary conditions. The initial conditions are

$$y_i(0) = 0, \qquad a_i(0) = -2\cos\left(\frac{2\pi i}{N}\right), \qquad b_i(0) = 2\sin\left(\frac{2\pi i}{N}\right),$$

for $i = 1, \ldots, N$. Note that the problem is singularly perturbed in the y component and the stiffness of the system can be controlled using ε. Following the splitting used in [83], the diffusion terms and the term scaled by ε^{-1} form g, while the remaining terms form f. The MATLAB implementation of the CUSP problem is available in [119].

We performed a fixed time-stepping convergence study of the new methods. Figure 4.2 shows the error of the final solution versus number of timesteps. Error is computed in the ℓ^2 sense using a high-accuracy reference solution. In all cases, the parallel IMEX GLMs converge at least at the same rate as the theoretical order of accuracy.



(a) Error versus steps for parallel IMEX DIMSIMs of orders two and seven. For this problem, the seventh order method converges faster than the nominal order.

(b) Error versus steps for parallel ensemble IMEX Euler methods of orders three and eight.

Figure 4.2: Convergence of parallel IMEX DIMSIM and parallel ensemble IMEX Euler methods for the CUSP problem (4.27).

4.6.2 Allen–Cahn Problem

We also consider the two-dimensional Allen–Cahn problem described in [165]. It is a reaction-diffusion system governed by the equation

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + \beta \left(u - u^3\right) + s, \tag{4.28}$$

where $\alpha = 0.1$ and $\beta = 3$. The time-dependent Dirichlet boundary conditions and source term $s(t, x, y)$ are derived using method of manufactured solutions such that the exact solution is

$$u(t, x, y) = 2 + \sin(2\pi\,(x - t))\cos(3\pi\,(y - t)).$$

We discretize the PDE on a unit square domain using degree two Lagrange finite elements and a uniform triangular mesh with $N = 32$ points in each direction. The diffusion term and forcing associated with the boundary conditions are treated implicitly, while the reaction and source term are treated explicitly.

The problem is implemented using the FEniCS package [5] leveraging OpenMP parallelism to speed up f and g evaluations, as well as MPI parallelism of stage computations made possible by the structure of the parallel IMEX GLMs. All tests were run on the Cascades cluster maintained by Virginia Tech's Advance Research Computing center (ARC). Parallel

experiments were performed on $p = q = r = s$ nodes, each using 12 cores. Serial experiments were done on a single node with the same number of cores. The error was computed using the ℓ_2 norm by comparing the nodal values of the numerical solution against a high-accuracy reference solution.

Figure 4.3 summarizes the results of this experiment by comparing several additive Runge–Kutta (ARK) methods and IMEX DIMSIMs with Parallel IMEX GLMs derived in this paper. At order three, serial methods are ARK3(2)4L[2]SA from [85] and IMEX-DIMSIM3 from [36]. At order four, comparisons are done against ARK4(3)7L[2]SA$_1$ from [88] and IMEX-DIMSIM4 from [165]. Order five serial methods are ARK5(4)8L[2]SA$_2$ from [88] and IMEX-DIMSIM5 from [165]. Finally, the order six baseline is IMEX-DIMSIM6($\mathcal{S}_{\pi/2}$) from [83]. The results show parallel ensemble IMEX Euler methods are the most efficient in all cases. Parallel IMEX DIMSIMs are competitive at orders three and six and surpass the efficiency of serial schemes at orders four and five.

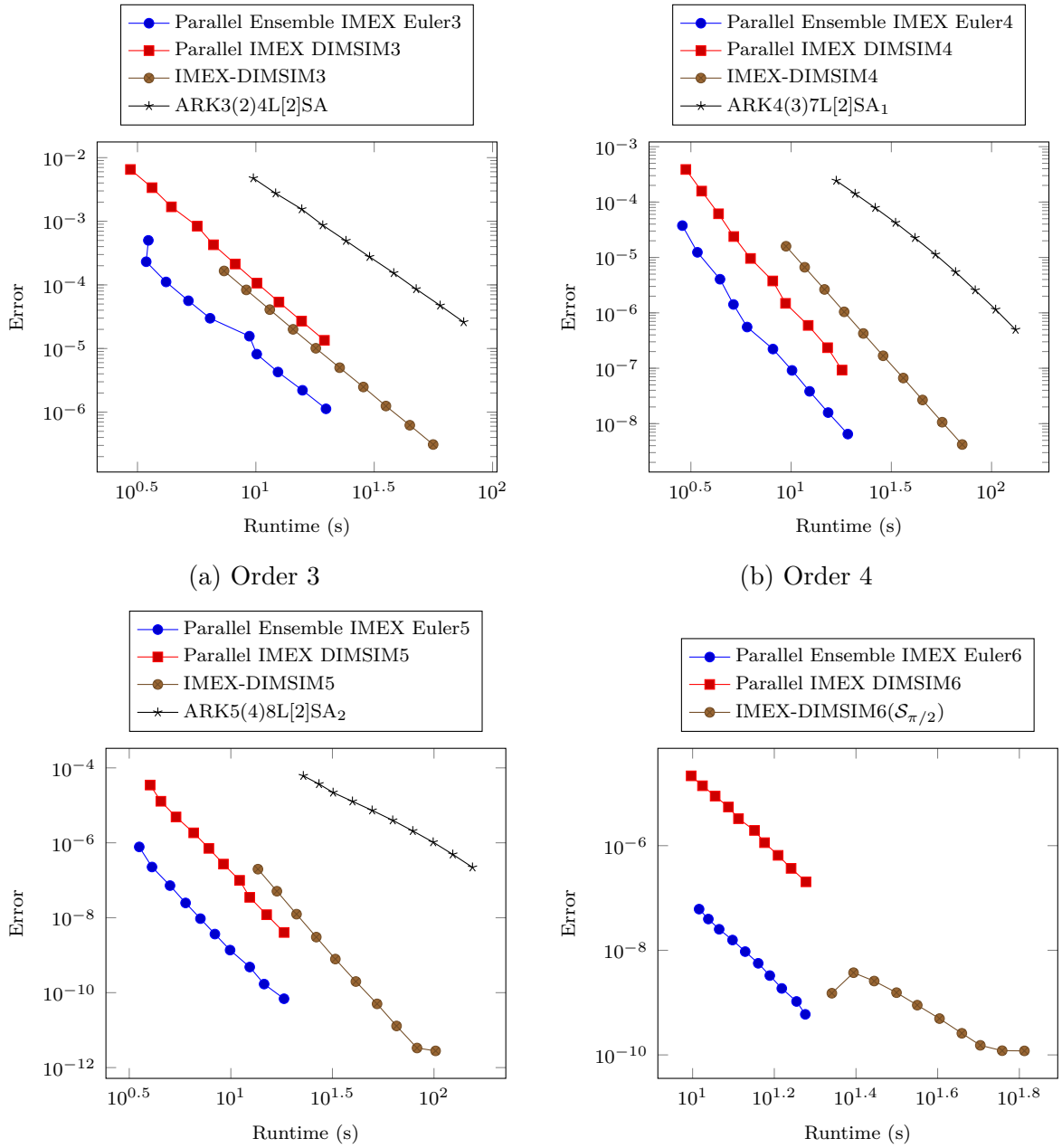
Figure 4.4 plots convergence of the methods used in the experiment. We can see the ARK methods exhibit order reduction for this problem, which explains their poor efficiency results. All other methods achieve the expected order of accuracy. For a fixed number of steps, the parallel IMEX GLMs are less accurate than the serial IMEX GLMs, which indicates parallel methods have larger error constants and are not the most efficient when limited to serial execution. This is to be expected given parallel methods have a more restrictive structure and less coefficients available for optimizing the principal error and stability.

4.7 Conclusion

This paper studies parallel IMEX GLMs and provides a methodology to derive and solve simple order conditions for methods of arbitrary order. Using this framework, we construct two families of methods, based on existing DIMSIMs and on IMEX Euler, and provide linear stability analyses for them.

Our numerical experiments show that parallel IMEX GLMs can outperform existing serial IMEX schemes. Between parallel IMEX DIMSIMs and parallel ensemble IMEX Euler methods, the latter proved to be the most competitive. The error for the ensemble methods is generally smaller than that of the DIMSIMs, due in part to the improved ending procedure. Moreover, the magnitude of method coefficients grows slower for ensemble methods as documented in tables 4.1 and 4.2, reducing the impact of accumulated floating-point errors. For orders five and higher we have to carefully select the method and distribution of the abscissae to control these errors. In addition, one notes that parallel ensemble IMEX Euler methods tend to have smaller values of λ, which improves convergence of iterative linear solvers used in the Newton iterations.

Owing to their excellent stability properties, the ensemble family shows great potential for constructing other types of partitioned GLMs. Of particular interest are alternating



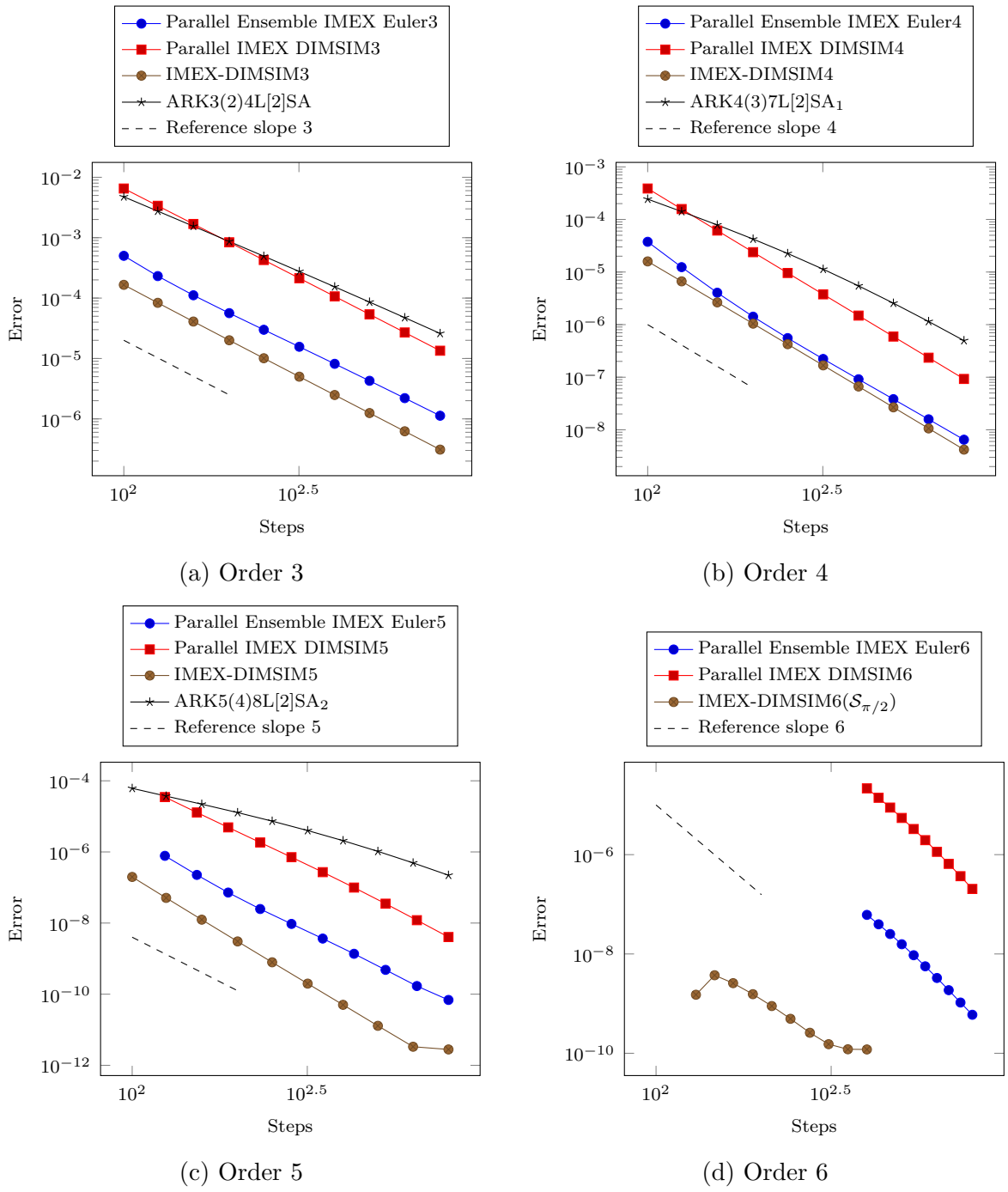
(a) Order 3

(b) Order 4

(c) Order 5

(d) Order 6

Figure 4.3: Work-precision diagrams for the Allen–Cahn problem (4.28).



(a) Order 3

(b) Order 4

(c) Order 5

(d) Order 6

Figure 4.4: Convergence diagrams for the Allen–Cahn problem (4.28).

directions implicit (ADI) GLMs [136], as well as multirate GLMs. The authors hope to study these in future works.

Chapter 5

A Fast Time-Stepping Strategy for Dynamical Systems Equipped with a Surrogate Model

Material from: Steven Roberts, Andrey A Popov, Arash Sarshar, and Adrian Sandu. A fast time-stepping strategy for dynamical systems equipped with a surrogate model. *arXiv preprint arXiv:2011.03688*, 2020

5.1 Introduction

This paper will consider the system of ordinary differential equations (ODEs)

$$y' = f(t, y), \quad y(t_0) = y_0, \quad y \in \mathbb{C}^{\mathrm{N}}, \tag{5.1}$$

equipped with a surrogate model approximating the dynamics of (5.1):

$$y'_{\mathrm{sur}} = f_{\mathrm{sur}}(t, y_{\mathrm{sur}}), \qquad y_{\mathrm{sur}} \in \mathbb{C}^{\mathrm{S}}. \tag{5.2}$$

This surrogate model is assumed to be much less expensive to solve than the full model (5.1), possibly by evolving in a lower dimensional space ($\mathrm{S} < \mathrm{N}$). Moreover, we assume that projections between the full and surrogate model spaces are realized by the matrices $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{\mathrm{N}\times\mathrm{S}}$:

$$y_{\mathrm{sur}} = \mathbf{W}^* y, \quad y \approx \mathbf{V} y_{\mathrm{sur}}, \quad \mathbf{W}^*\mathbf{V} = I_{\mathrm{S}\times\mathrm{S}}. \tag{5.3}$$

Here, the "$*$" symbol denotes the conjugate transpose of a matrix.

This paper presents a new technique to supplement the numerical integration of (5.1) using the surrogate model (5.2). For an appropriate choice of surrogate model, our method can be significantly more efficient than using either the full or the surrogate model alone. This is accomplished by applying a multirate time integration scheme to the following ODE with fast dynamics $f^{\{\mathrm{f}\}}$ and slow dynamics $f^{\{\mathrm{s}\}}$:

$$y' = \underbrace{\mathbf{V} f_{\mathrm{sur}}(t, \mathbf{W}^* y)}_{f^{\{\mathrm{f}\}}(t,y)} + \underbrace{f(t, y) - \mathbf{V} f_{\mathrm{sur}}(t, \mathbf{W}^* y)}_{f^{\{\mathrm{s}\}}(t,y)}. \tag{5.4}$$

Multirate methods are characterized by using different stepsizes for different parts of an ODE, as opposed to a single, global timestep [9, 41, 43, 60, 61, 66, 70, 118, 123, 129, 130, 133]. For problems with partitions exhibiting vastly different time scales, stiffnesses, evaluation costs, or amounts of nonlinearity, multirate methods can be more efficient that their single rate counterparts. In (5.4), the fast partition $f^{\{\mathfrak{f}\}}$ is treated with a small timestep, and the slow partition $f^{\{\mathfrak{s}\}}$ is treated with a large timestep. Note that (5.4) has the same solution as (5.1). It is rewritten and partitioned, however, in such a way that $f^{\{\mathfrak{f}\}}$ contains the surrogate model dynamics and $f^{\{\mathfrak{s}\}}$ represents the error in the surrogate model. Ideally, this error will be small, so a large timestep would be acceptable. Moreover, this means expensive evaluations of the full model occur infrequently compared to the inexpensive surrogate model evaluations.

With about 60 years of development [118], the multirating strategy has been applied to numerous classes of time integration methods. Conceivably any multirate method suitable for additively partitioned systems could be applied to (5.4). This paper builds upon multirate infinitesimal methods [17, 92, 141, 146, 161], as they offer a particularly flexible and elegant approach to multirate integration. In particular, we use the multirate infinitesimal general-structure additive Runge–Kutta (MRI-GARK) framework proposed in [121, 129]. MRI-GARK is an appealing choice as it generalizes many types of multirate infinitesimal methods, allows for the construction of high order methods, and supports implicit stages. We note, however, that this paper primarily focuses on explicit methods where (5.1) is nonstiff.

There are other instances in the literature where multirate methods and surrogate models are used in conjunction. In [69], an implicit multirate scheme is used to simulate a fast-evolving electric circuit coupled with a slow-evolving electric field. Model order reduction is then applied to the electric field problem to further reduce the computational cost. A multirate Runge–Kutta–Chebyshev method is proposed in [3] using a time-averaged right-hand side, which can be viewed as a surrogate model with better stiffness properties. From the multiscale modeling community, there are several related ideas including heterogeneous multiscale methods and projective integration[2, 156]. While these examples are not limited to time integration problems, they exhibit similarities to multirate methods when used in the time integration context. They seek to efficiently approximate macroscopic quantities of interest given a microscopic model. In contrast, the methods of this paper seek to approximate the full, microscopic state with the help of macroscopic approximations. High-order/low-order methods [37] are another example in this category but are designed to solve coupled, nonlinear systems of equations. Finally, surrogate models have been used extensively to speed up optimization problems when objective function evaluations are expensive [53, 102, 112].

This remainder of this paper is structured as follows. In section 5.2, we review the MRI-GARK framework and extensions. Section 5.3 then specializes MRI-GARK-type methods to the special ODE (5.4). An error analysis is performed in section 5.4, and various approaches for constructing surrogate models are discussed in section 5.5. Convergence and performance experiments can be found in section 5.6. Finally, section 5.7 summarizes the results of the paper and proposes future extensions.

5.2 Multirate Infinitesimal General-Structure Additive Runge-Kutta Methods

In this section, we will briefly review the MRI-GARK framework, as it serves as the foundation for the remainder of the paper. These methods numerically solve ODEs that are additively partitioned into fast and slow dynamics:

$$y' = f(t, y) = f^{\{\mathfrak{f}\}}(t, y) + f^{\{\mathfrak{s}\}}(t, y). \tag{5.5}$$

The defining characteristic of multirate infinitesimal methods is that the slow dynamics $f^{\{\mathfrak{s}\}}$ is propagated with a discrete method, most commonly a Runge–Kutta method, while the fast dynamics $f^{\{\mathfrak{f}\}}$ is propagated continuously through modified fast ODEs. An MRI-GARK scheme [129] advances the solution of (5.5) from time t_n to $t_{n+1} = t_n + H$ via the following computational process:

$$Y_1^{\{\mathfrak{s}\}} = y_n, \tag{5.6a}$$

$$\begin{cases} v_i(0) = Y_i^{\{\mathfrak{s}\}}, \\ \quad T_i = t_n + c_i^{\{\mathfrak{s}\}} H, \\ v_i'(\theta) = \Delta c_i^{\{\mathfrak{s}\}} f^{\{\mathfrak{f}\}}\big(T_i + \Delta c_i^{\{\mathfrak{s}\}}\theta, v_i(\theta)\big) + \displaystyle\sum_{j=1}^{i+1} \gamma_{i,j}\big(\tfrac{\theta}{H}\big) f^{\{\mathfrak{s}\}}\big(T_j, Y_j^{\{\mathfrak{s}\}}\big), \\ \qquad \text{for } \theta \in [0, H], \\ Y_{i+1}^{\{\mathfrak{s}\}} = v_i(H), \qquad i = 1, \ldots, s^{\{\mathfrak{s}\}}, \end{cases} \tag{5.6b}$$

$$y_{n+1} = Y_{s^{\{\mathfrak{s}\}}+1}^{\{\mathfrak{s}\}}. \tag{5.6c}$$

There are $s^{\{\mathfrak{s}\}}$ modified fast ODEs (5.6b) solved between the abscissae of a Runge–Kutta method. The distances between consecutive abscissae are

$$\Delta c_i^{\{\mathfrak{s}\}} = \begin{cases} c_{i+1}^{\{\mathfrak{s}\}} - c_i^{\{\mathfrak{s}\}}, & i = 1, \ldots, s^{\{\mathfrak{s}\}} - 1, \\ 1 - c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}}, & i = s^{\{\mathfrak{s}\}}. \end{cases}$$

Equation (5.6b) also contains time-dependent, polynomial forcing terms for the slow dynamics which satisfy

$$\gamma_{i,j}(t) := \sum_{k \geq 0} \gamma_{i,j}^k t^k, \quad \widetilde{\gamma}_{i,j}(t) := \int_0^t \gamma_{i,j}(\tau) \mathrm{d}\tau = \sum_{k \geq 0} \gamma_{i,j}^k \frac{t^{k+1}}{k+1}, \quad \overline{\gamma}_{i,j} := \widetilde{\gamma}_{i,j}(1). \tag{5.7}$$

Continuing with the notation of [129], capitalized versions of (5.7) denote matrices of coefficients. If $\overline{\Gamma}$ is lower triangular, (5.6) is explicit, and otherwise, it is implicit.

For general $f^{\{\mathfrak{f}\}}$, one cannot expect to find a closed-form solution to the modified fast ODE (5.6b); however, one can determine $v_i(H)$ by time-stepping with "infinitesimally" small

timesteps. For the sake of simplifying the analysis, we assume this integration is exact, but computationally, it is not possible to take infinitely many steps. In practice, (5.6b) is integrated to an accuracy that is negligible compared to the other source of error: treating $f^{\{\mathfrak{s}\}}$ with a discrete Runge–Kutta method. This is analogous to assuming integrations within a Strang splitting are exact, but using sufficiently accurate approximations in practice. Multirate infinitesimal methods enjoy great flexibility in that *any* consistent time integration method can be used to solve the modified fast ODEs. This offers the freedom to choose implicit or explicit methods, as well as methods of any order. Equation (5.6) is multirate in the sense that there are $s^{\{\mathfrak{s}\}}$ evaluations of $f^{\{\mathfrak{s}\}}$ over a timestep H, while (5.6b) is generally integrated with a smaller timestep and requires many more evaluations of $f^{\{\mathfrak{f}\}}$.

5.2.1 Coupled MRI-GARK Schemes

In [121] two new types of MRI-GARK methods were proposed: step predictor-corrector MRI-GARK (SPC-MRI-GARK) and internal predictor-corrector MRI-GARK (IPC-MRI-GARK). In order to improve the linear stability of MRI-GARK, both include coupled, discrete prediction stages that are subsequently corrected by solving modified fast ODEs. In comparing the two MRI-GARK variants, SPC-MRI-GARK was identified to have simpler order conditions, better stability, and a simpler implementation. For these reasons, we will only consider SPC-MRI-GARK in the construction of surrogate model timestepping techniques.

An SPC-MRI-GARK method based on the "slow" Runge–Kutta method $(\mathcal{A}^{\{\mathfrak{s},\mathfrak{s}\}}, \mathcal{b}^{\{\mathfrak{s}\}}, \mathcal{c}^{\{\mathfrak{s}\}})$ solves (5.5) with steps of the form:

$$Y_i = y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \mathcal{a}_{i,j}^{\{\mathfrak{s}\}} f\big(t_n + \mathcal{c}_j^{\{\mathfrak{s}\}} H, Y_j\big), \qquad i = 1, \ldots, s^{\{\mathfrak{s}\}}, \tag{5.8a}$$

$$\begin{cases} v(0) = y_n, \\ \quad v' = f^{\{\mathfrak{f}\}}(t_n + \theta, v) + \displaystyle\sum_{j=1}^{s^{\{\mathfrak{s}\}}} \gamma_j\big(\tfrac{\theta}{H}\big) f^{\{\mathfrak{s}\}}\big(t_n + \mathcal{c}_j^{\{\mathfrak{s}\}} H, Y_j\big), \quad \text{for } \theta \in [0, H], \\ y_{n+1} = v(H). \end{cases} \tag{5.8b}$$

Similar to a traditional Runge–Kutta scheme, SPC-MRI-GARK starts by computing stage values that approximate the solution at the times $t_n + \mathcal{c}_i^{\{\mathfrak{s}\}} H$. In this prediction step (5.8a), the fast and slow dynamics are treated together with the timestep H. The fast part of the stage values is inaccurate from this large timestep and is discarded. A single ODE (5.8b) is solved from t_n to t_{n+1} to correct the fast dynamics and produce the next step y_{n+1}. Like MRI-GARK, there are time-dependent, polynomial forcing terms for the slow tendencies. Following [121, Definition 2.2], the definitions of $\gamma_i(t)$, $\widetilde{\gamma}_i(t)$, $\overline{\gamma}_i$ are identical to (5.7), except they are vector quantities with a single subscript index. While base methods of any structure

are admitted in the SPC-MRI-GARK framework, only diagonally implicit methods were considered in [121]. As the focus of this paper is explicit methods, we have derived new explicit methods of orders two and three and present their coefficients in appendix D.

5.3 Method Formulation

Following the process outlined in the introduction, we seek to apply the MRI-GARK method (5.6) to the partitioned problem (5.4) in order to construct a solution process that can incorporate surrogate model information. We start with the modified fast ODE (5.6b):

$$\begin{aligned} v_i'(\theta) = \Delta c_i^{\{s\}} \mathbf{V} f_{\text{sur}}\big(T_i + \Delta c_i^{\{s\}}\theta, \mathbf{W}^* v_i(\theta)\big) \\ + \sum_{j=1}^{i+1} \gamma_{i,j}\big(\tfrac{\theta}{H}\big)\big(f(T_j, Y_j) - \mathbf{V} f_{\text{sur}}(T_j, \mathbf{W}^* Y_j)\big). \end{aligned} \tag{5.9}$$

In (5.9), the full model f only appears in the forcing terms, and for explicit methods, it is evaluated at previously computed stage values. Only the surrogate model f_{sur} needs to be evaluated at each of the infinitesimal steps to compute $v_i(H)$.

Note that in formulation (5.9), $v_i \in \mathbb{C}^{\text{N}}$ evolves in the full space, and there is no benefit from choosing a surrogate model that evolves in a lower dimensional space. To resolve this, we first split v_i into the parts residing inside and outside the surrogate model space:

$$v_i(\theta) = \mathbf{V}\mathbf{W}^* v_i(\theta) + \left(I_{\text{N}\times\text{N}} - \mathbf{V}\mathbf{W}^*\right) v_i(\theta).$$

More formally, $\mathbf{V}\mathbf{W}^*$ and $(I_{\text{N}\times\text{N}} - \mathbf{V}\mathbf{W}^*)$ define oblique projections onto the range of $\mathbf{V}$ and the nullspace of $\mathbf{W}^*$, respectively. We now consider the solution to (5.9) from the perspective of these two, complementary subspaces.

Starting outside the surrogate model space, we have that

$$\begin{aligned} \left(I_{\text{N}\times\text{N}} - \mathbf{V}\mathbf{W}^*\right) v_i(H) &= \left(I_{\text{N}\times\text{N}} - \mathbf{V}\mathbf{W}^*\right)\left(Y_i + \int_0^H v_i'(\theta)\,\mathrm{d}\theta\right) \\ &= \left(I_{\text{N}\times\text{N}} - \mathbf{V}\mathbf{W}^*\right)\left(Y_i + H\sum_{j=1}^{i} \overline{\gamma}_{i,j} f(T_j, Y_j)\right), \end{aligned} \tag{5.10}$$

where we have used (5.7) and the fact that $(I_{\text{N}\times\text{N}} - \mathbf{V}\mathbf{W}^*)\,\mathbf{V} = 0_{\text{N}\times\text{S}}$ by (5.3). Equation (5.10) reveals that inside the nullspace of $\mathbf{W}^*$ the solution to equation (5.9) becomes a Runge–Kutta stage. Therefore, we exclude the components in this subspace during the infinitesimal step integration of (5.9).

We now consider the dynamics (5.9) projected onto the surrogate model space:

$$\begin{aligned} z_i(\theta) &\coloneqq \mathbf{W}^* v_i(\theta) \in \mathbb{C}^{\mathrm{S}}, \\ z_i'(\theta) &= \Delta c_i^{\{\mathfrak{s}\}} f_{\mathrm{sur}}\big(T_i + \Delta c_i^{\{\mathfrak{s}\}}\theta, z_i(\theta)\big) \\ &\quad + \sum_{j=1}^{i+1} \gamma_{i,j}\big(\tfrac{\theta}{H}\big)\left(\mathbf{W}^* f(T_j, Y_j) - f_{\mathrm{sur}}(T_j, \mathbf{W}^* Y_j)\right). \end{aligned} \tag{5.11}$$

The solution to (5.9) can be expressed as the sum of its two components:

$$v_i(H) = \mathbf{V} z_i(H) + (I_{\mathrm{N}\times\mathrm{N}} - \mathbf{V}\mathbf{W}^*)\left(Y_i + H\sum_{j=1}^{i+1} \overline{\gamma}_{i,j} f(T_j, Y_j)\right).$$

Thus to compute an MRI-GARK stage, we only need to solve an ODE of dimension S instead of N. We summarize the simplifications of (5.6) in proposition 5.1 and provide a graphical illustration in fig. 5.1.

Definition 5.1 (SM-MRI-GARK)**.** A surrogate model MRI-GARK (SM-MRI-GARK) method solves the ODE (5.1) with the help of the surrogate model (5.2) using steps of the form

$$Y_1 = y_n, \tag{5.12a}$$

$$\begin{cases} z_i(0) = \mathbf{W}^* Y_i, \\ \quad T_i = t_n + c_i^{\{\mathfrak{s}\}} H, \\ z_i'(\theta) = \Delta c_i^{\{\mathfrak{s}\}} f_{\mathrm{sur}}\big(T_i + \Delta c_i^{\{\mathfrak{s}\}}\theta, z_i(\theta)\big) \\ \qquad + \sum_{j=1}^{i+1} \gamma_{i,j}\big(\tfrac{\theta}{H}\big)\left(\mathbf{W}^* f(T_j, Y_j) - f_{\mathrm{sur}}(T_j, \mathbf{W}^* Y_j)\right), \quad \text{for } \theta \in [0, H], \\ Y_{i+1} = \mathbf{V} z_i(H) + (I_{d\times d} - \mathbf{V}\mathbf{W}^*)\left(Y_i + H\sum_{j=1}^{i+1} \overline{\gamma}_{i,j} f(T_j, Y_j)\right), \\ \qquad i = 1, \dots, s^{\{\mathfrak{s}\}}, \end{cases} \tag{5.12b}$$

$$y_{n+1} = Y_{s^{\{\mathfrak{s}\}}+1}. \tag{5.12c}$$

Remark 5.2 (SM-MRI-GARK for hierarchical surrogate models)**.** For certain applications, an entire hierarchy of surrogate models $f, f_{\mathrm{sur}}^{[1]}, f_{\mathrm{sur}}^{[2]}, \dots, f_{\mathrm{sur}}^{[m]}$ may be available. In such scenarios, one can take advantage of all models by applying SM-MRI-GARK recursively. First, the highest fidelity models f and $f_{\mathrm{sur}}^{[1]}$ are applied to (5.12). The ODE (5.12b), is then solved using an SM-MRI-GARK method with $\Delta c_i^{\{\mathfrak{s}\}} f_{\mathrm{sur}}^{[2]}$ as the surrogate model. Its ODEs are solved using an SM-MRI-GARK method with $\Delta c_i^{\{\mathfrak{s}\}\times 2} f_{\mathrm{sur}}^{[3]}$ as the surrogate model. This is continued until we reach $\Delta c_i^{\{\mathfrak{s}\}\times(m-1)} f_{\mathrm{sur}}^{[m]}$ as the lowest fidelity surrogate model. Note the power of $\Delta c_i^{\{\mathfrak{s}\}}$ scaling the surrogate models can be eliminated by applying the change of variable $\theta \to \theta/\Delta c_i^{\{\mathfrak{s}\}}$ to (5.12b).

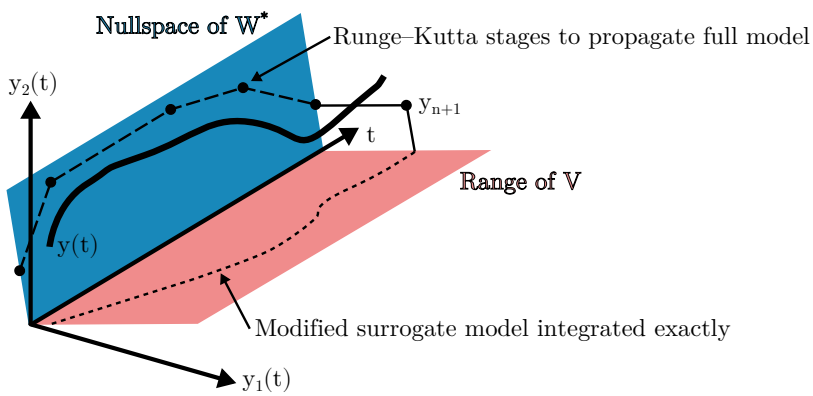


Figure 5.1: Illustration of SM-MRI-GARK for a two-variable ODE and a surrogate model that evolves in a one-dimensional subspace.

Remark 5.3 (Connection to EPIRK-K methods)**.** In [129], Sandu showed that exponential Runge–Kutta methods are special case of MRI-GARK methods when the linear-nonlinear partitioning

$$y' = \underbrace{\mathbf{L}\,y}_{f^{\{\mathrm{f}\}}(t,y)} + \underbrace{f(t,y) - \mathbf{L}\,y}_{f^{\{\mathrm{s}\}}(t,y)}$$

is used. Conversely, MRI-GARK can be viewed as a generalization of exponential Runge–Kutta methods where the piece integrated exactly can be linear or nonlinear.

An analogous connection can be made between EPIRK-K [100] and SM-MRI-GARK. EPIRK-K is an extension of exponential propagation iterative methods of Runge-Kutta type (EPIRK) [155] that approximates the Jacobian $\mathbf{J}_n = \frac{\partial f}{\partial y}(t_n, y_n)$ appearing in matrix exponentials and related matrix functions with a Krylov-subspace approximation $\mathbf{U}\mathbf{U}^*\mathbf{J}_n\mathbf{U}\mathbf{U}^*$. SM-MRI-GARK with $\mathbf{V} = \mathbf{W} = \mathbf{U}$ and $f_{\mathrm{sur}}(t, y_{\mathrm{sur}}) = \mathbf{U}^*\mathbf{J}_n\mathbf{U}y_{\mathrm{sur}}$ closely resembles the class of EPIRK-K schemes.

Remark 5.4 (Stepsize adaptivity)**.** Even with fixed timesteps H, adaptive time-steppers can be used to integrate (5.12b) so that f_{sur} is called just enough to meet tolerances. It is also possible to independently and adaptively select H and thus the frequency at which f is evaluated. Following [129, Remark 2.5], (5.12) can be equipped with an embedded method to estimate the local truncation error, and then standard error controllers can be used [74, Section II.4]. Each method in appendix D includes an embedded method that is one order lower than the main method.

As a simple example of an SM-MRI-GARK method, consider forward Euler as the base method and the internally consistent [129, Section 3.1.1] coupling $\gamma_{1,1}(t) = 1$:

$$\begin{aligned} z(0) &= \mathbf{W}^* y_n \\ z'(\theta) &= f_{\mathrm{sur}}(t_n + \theta, z(\theta)) + \mathbf{W}^* f(t_n, y_n) - f_{\mathrm{sur}}(t_n, \mathbf{W}^* y_n), \qquad \text{for } \theta \in [0, H], \\ y_{n+1} &= \mathbf{V} z(H) + (I_{d\times d} - \mathbf{V}\mathbf{W}^*)\,(y_n + H f(t_n, y_n))\,. \end{aligned} \tag{5.13}$$

Equation (5.13) is only first order accurate; however, second and third order SM-MRI-GARK schemes can be constructed using the coefficients provided in appendix D. In order to implement these efficiently, we have provided pseudocode in proposition 5.5. It minimizes the number of full and surrogate model evaluations, as well as the number of matrix-vector products involving $\mathbf{V}$ and $\mathbf{W}^*$.

Algorithm 5.5. Pseudocode for explicit SM-MRI-GARK (5.12).

procedure SM-MRI-GARK$(f, f_{\text{sur}}, \mathbf{V}, \mathbf{W}^*, t_0, t_{\text{end}}, y_0, N_{\text{steps}})$
  $y = y_0$
  $\widehat{y} = \mathbf{W}^* y_0$
  $H = (t_{\text{end}} - t_0)/N_{\text{steps}}$
  for $n = 1, \ldots, N_{\text{steps}}$ **do**
    for $i = 1, \ldots, s^{\{\mathfrak{s}\}}$ **do**
      $T_i = t_0 + (n + c_i^{\{\mathfrak{s}\}})H$
      $k_i = f(T_i, y)$
      $\widehat{k}_i = \mathbf{W}^* k_i$
      $\widehat{\ell}_i = \widehat{k}_i - f_{\text{sur}}(T_i, \widehat{y})$
      $\widehat{w} = \widehat{y} + H \sum_{j=1}^{i} \overline{\gamma}_{i,j} \widehat{k}_j$
      $y = y + H \sum_{j=1}^{i} \overline{\gamma}_{i,j} k_j$
      $\widehat{y} = \text{OdeSolve}\Bigg(z_i'(\theta) = \Delta c_i^{\{\mathfrak{s}\}} f_{\text{sur}}(T_i + \Delta c_i^{\{\mathfrak{s}\}}\theta, z_i(\theta)) + \sum_{j=1}^{i} \gamma_{i,j} \left(\frac{\theta}{H}\right) \widehat{\ell}_j,$
      $\text{timespan} = [0, H], \text{initial\_condition} = \widehat{y} \Bigg)$
      $y = y + \mathbf{V}(\widehat{y} - \widehat{w})$
    end for
  end for
  return y
end procedure

5.3.1 Formulation for SPC-MRI-GARK

A similar process can be used to apply SPC-MRI-GARK (5.8) to the ODE (5.4). For brevity, we avoid replicating the simplifications (5.10) and (5.11) and directly present the method formulation in the following definition.

Definition 5.6 (SM-SPC-MRI-GARK)**.** A surrogate model SPC-MRI-GARK (SM-SPC-MRI-GARK) method solves the ODE (5.1) with the help of the surrogate model (5.2) using

steps of the form

$$Y_i = y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s}\}} f\big(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j\big), \qquad i = 1, \dots, s^{\{\mathfrak{s}\}}, \tag{5.14a}$$

$$\begin{cases} z(0) = \mathbf{W}^* y_n, \\ z(\theta)' = f_{\text{sur}}(t_n + \theta, z(\theta)) \\ \qquad\qquad + \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \gamma_j\big(\frac{\theta}{H}\big) \Big(\mathbf{W}^* f\big(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j\big) - f_{\text{sur}}\big(t_n + c_j^{\{\mathfrak{s}\}} H, \mathbf{W}^* Y_j\big)\Big), \\ \qquad\qquad \text{for } \theta \in [0, H], \\ y_{n+1} = \mathbf{V} z(H) + (I_{d\times d} - \mathbf{V}\mathbf{W}^*) \left(y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} b_j f\big(t_n + c_j^{\{\mathfrak{s}\}} H, Y_j\big) \right). \end{cases} \tag{5.14b}$$

The SM-MRI-GARK Euler method from (5.13) also happens to be an SM-SPC-MRI-GARK method. The base method is again forward Euler, and the coupling is $\gamma_1(t) = 1$. Methods with more than one stage will not coincide, however, as SM-SPC-MRI-GARK has one modified fast ODE per step while SM-MRI-GARK has $s^{\{\mathfrak{s}\}}$. Another distinction, which can be seen in proposition 5.7, is that SM-SPC-MRI-GARK only requires one matrix-vector product with $\mathbf{V}$ per step, while SM-MRI-GARK requires $s^{\{\mathfrak{s}\}}$.

Algorithm 5.7. Pseudocode for explicit SM-SPC-MRI-GARK (5.14).

procedure SM-SPC-MRI-GARK($f, f_{\text{sur}}, \mathbf{V}, \mathbf{W}^*, t_0, t_{\text{end}}, y_0, N_{\text{steps}}$)
  $y = y_0$
  $\widehat{y} = \mathbf{W}^* y_0$
  $H = (t_{\text{end}} - t_0)/N_{\text{steps}}$
  for $n = 1, \dots, N_{\text{steps}}$ **do**
    $t_n = t_0 + nH$
    for $i = 1, \dots, s^{\{\mathfrak{s}\}}$ **do**
      $k_i = f\big(t_n + c_i^{\{\mathfrak{s}\}} H, y + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} k_j\big)$
      $\widehat{k}_i = \mathbf{W}^* k_i$
      $\widehat{\ell}_i = \widehat{k}_i - f_{\text{sur}}\big(t_n + c_i^{\{\mathfrak{s}\}} H, \widehat{y} + H \sum_{j=1}^{i-1} a_{i,j}^{\{\mathfrak{s}\}} \widehat{k}_j\big)$
    end for
    $\widehat{w} = \widehat{y} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} b_j^{\{\mathfrak{s}\}} \widehat{k}_j$
    $y = y + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} b_j^{\{\mathfrak{s}\}} k_j$
    $\widehat{y} = \text{OdeSolve}\Big(z'(\theta) = f_{\text{sur}}(t_n + \theta, z(\theta)) + \sum_{j=1}^{s^{\{\mathfrak{s}\}}} \gamma_j\big(\frac{\theta}{H}\big) \widehat{\ell}_j,$
      $\text{timespan} = [0, H], \text{initial\_condition} = \widehat{y}\Big)$
    $y = y + \mathbf{V}(\widehat{y} - \widehat{w})$
  end for
  return y

18: **end procedure**

5.4 Error Analysis

Consider the edge case where the surrogate model is taken to be the full model and $\mathbf{V} = \mathbf{W}^* = I_{\mathrm{N}\times\mathrm{N}}$. The stages of (5.12) simplify to

$$\begin{aligned} &Y_1 = y_n, \\ &\begin{cases} z_i(0) = Y_i, \\ \quad z_i' = \Delta c_i^{\{\mathfrak{s}\}} f_{\mathrm{sur}}\big(t_n + c_i^{\{\mathfrak{s}\}} H + \Delta c_i^{\{\mathfrak{s}\}}\theta, z_i\big), \qquad \text{for } \theta \in [0, H], \\ Y_{i+1} = z_i(H), \qquad i = 1, \ldots, s^{\{\mathfrak{s}\}}, \end{cases} \\ &y_{n+1} = Y_{s^{\{\mathfrak{s}\}}+1}. \end{aligned}$$

This is simply the original ODE (5.1) solved between the abscissae of a Runge–Kutta method. With the infinitesimal step assumption, the error is zero. At the other extreme, take the surrogate model to be $f_{\mathrm{sur}}(t, y_{\mathrm{sur}}) = 0_{\mathrm{N}}$. Again, (5.12) can be simplified, but now into a traditional, unpartitioned Runge–Kutta method:

$$\begin{aligned} Y_1 &= y_n, \\ Y_{i+1} &= Y_i + H \sum_{j=1}^{i} \overline{\gamma}_{i,j} f\big(t_n + c_i^{\{\mathfrak{s}\}} H, Y_j\big), \qquad i = 1, \ldots, s^{\{\mathfrak{s}\}}, \\ y_{n+1} &= Y_{s^{\{\mathfrak{s}\}}+1}. \end{aligned}$$

This error will be nonzero in general and can be analyzed using classical Runge–Kutta order condition and convergence theory.

While neither of these two choices for the surrogate model are particularly interesting, they illustrate how the accuracy of SM-MRI-GARK and SM-SPC-MRI-GARK depends on several factors including the accuracy of the surrogate model, the projection matrices $\mathbf{V}$ and $\mathbf{W}$, and the method coefficients. In this section, we analyze the structure of local truncation error

$$\mathrm{LTE}_{n+1} = y(t_{n+1}) - y_{n+1} = C\, H^{p+1} + \mathcal{O}\big(H^{p+2}\big) \tag{5.15}$$

to better understand the influence of these factors. To simplify our error analysis, we use the autonomous form of (5.4), which comes at no loss of generality for internally consistent methods.

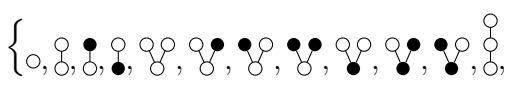
The new families of methods in this paper are derived as special cases of MRI-GARK and SPC-MRI-GARK. Interestingly, it is also possible to view MRI-GARK and SPC-MRI-GARK as special cases of the surrogate model methods. When

$$\mathbf{V} = \mathbf{W}^* = I_{\mathrm{N}\times\mathrm{N}}, \quad f(t, y) = f^{\{\mathfrak{f}\}}(t, y) + f^{\{\mathfrak{s}\}}(t, y), \quad f_{\mathrm{sur}}(t, y) = f^{\{\mathfrak{f}\}}(t, y),$$

we identically recover (5.6) and (5.8). Thanks to these properties, MRI-GARK and SPC-MRI-GARK order conditions are both necessary and sufficient for the surrogate model versions. Interested readers can refer to [129, Section 3.1] and [121, Section 2.2] for these order conditions, which are derived using N-tree theory [12]. Thus, the order of the surrogate-model-based integration only depends on the coefficients of the underlying MRI-GARK method. Assuming that f_{sur} is sufficiently smooth and has a moderate Lipschitz constant, one can use *any* surrogate model, even a very inaccurate one, without concern of losing the order.

We now discuss the leading error constant C in (5.15). N-Trees provide an elegant way to quantify the terms of which it is composed. For our multirate methods, there are two partitions, so we use the set of two-trees $\mathbb{T}_2$. Tree vertices are "colored" either fast (black circle) or slow (white circle). Let $\mathbb{T}_2^{\{\mathfrak{f}\}}$ be the set of trees in $\mathbb{T}_2$ with all vertices fast-colored (including the empty tree), and $\mathbb{T}_2 \setminus \mathbb{T}_2^{\{\mathfrak{f}\}}$ the set of trees with at least one slow-colored node. We have that

$$C = \sum_{\substack{\mathfrak{u}\in\mathbb{T}_2\setminus\mathbb{T}_2^{\{\mathfrak{f}\}}\\ \rho(\mathfrak{u})=p+1}} \frac{1}{\sigma(\mathfrak{u})}\left(\Phi(\mathfrak{u}) - \frac{1}{\gamma(\mathfrak{u})}\right) F(\mathfrak{u})(y), \tag{5.16}$$

$\mathbb{T}_2 \setminus \mathbb{T}_2^{\{\mathfrak{f}\}} =$ 

where $\rho(\mathfrak{u})$, $\sigma(\mathfrak{u})$, $\gamma(\mathfrak{u})$, are the order, symmetries, and density, respectively [12]. The elementary weight $\Phi(\mathfrak{u})$ depends only on the method and its coefficients. For $\mathfrak{u} \in \mathbb{T}_2^{\{\mathfrak{f}\}}$, $\Phi(\mathfrak{u}) = 1/\gamma(\mathfrak{u})$ by the infinitesimal step assumption, which allows us to exclude this subset of trees from the summation in (5.16). Finally, the elementary differentials have the recursive, tensorial definition:

$$F(\emptyset)(y) = y, \tag{5.17a}$$

$$F(\bullet)(y) = \mathbf{V} f_{\text{sur}}(\mathbf{W}^* y), \tag{5.17b}$$

$$F(\circ)(y) = f(y) - \mathbf{V} f_{\text{sur}}(\mathbf{W}^* y), \tag{5.17c}$$

$$F\Big([\mathfrak{u}_1, \dots, \mathfrak{u}_m]^{\{\mathfrak{f}\}}\Big)(y) = \mathbf{V} f_{\text{sur}}^{(m)}(\mathbf{W}^* y)\big(\mathbf{W}^* F(\mathfrak{u}_1)(y), \dots, \mathbf{W}^* F(\mathfrak{u}_m)(y)\big), \tag{5.17d}$$

$$F\Big([\mathfrak{u}_1, \dots, \mathfrak{u}_m]^{\{\mathfrak{s}\}}\Big)(y) = f^{(m)}(y)\big(F(\mathfrak{u}_1)(y), \dots, F(\mathfrak{u}_m)(y)\big) - F\Big([\mathfrak{u}_1, \dots, \mathfrak{u}_m]^{\{\mathfrak{f}\}}\Big)(y). \tag{5.17e}$$

A tree $\mathfrak{u} \in \mathbb{T}_2$ can be expressed as $[\mathfrak{u}_1, \dots, \mathfrak{u}_m]^{\{\sigma\}}$ where σ is the color of the root vertex and $\mathfrak{u}_1, \dots, \mathfrak{u}_m$ are the nonempty subtrees left when the root is removed. Table 5.1 provides examples of terms appearing in (5.16).

Using (5.16), the local truncation error of SM-MRI-GARK Euler from (5.13) is

$$\text{LTE}_{n+1}^{\text{Euler}} = \frac{1}{2}(f'(y_n) - \mathbf{V} f'_{\text{sur}}(\mathbf{W}^* y_n)\mathbf{W}^*) f(y_n) H^2 + \mathcal{O}\big(H^3\big).$$

$\mathfrak{u}$	$F(\mathfrak{u})(y)$	$\rho(\mathfrak{u})$	$\sigma(\mathfrak{u})$	$\gamma(\mathfrak{u})$
$\circ$	$f(y) - \mathbf{V} f_{\text{sur}}(\mathbf{W}^* y)$	1	1	1
$\bullet\!-\!\circ$	$f'(y)\mathbf{V} f_{\text{sur}}(\mathbf{W}^* y) - \mathbf{V} f'_{\text{sur}}(\mathbf{W}^* y) f_{\text{sur}}(\mathbf{W}^* y)$	2	1	2
$\bullet\!-\!\circ\!-\!\bullet$	$f''(y)\big(f(y), f(y)\big) - f''(y)\big(f(y), \mathbf{V} f_{\text{sur}}(\mathbf{W}^* y)\big)$	3	1	3

Table 5.1: Examples of trees, elementary differentials, and other tree properties.

Note that each tree in the summation in (5.16) contains at least one slow node. From (5.17e), a slow-colored vertex corresponds to the surrogate model error or one of its derivatives appearing in an elementary differential. In order to quantify the error in the surrogate model, suppose there exists a (small) constant ε such that:

$$\|f(y_n) - \mathbf{V} f_{\text{sur}}(\mathbf{W}^* y_n)\| \leq \varepsilon,$$
$$\big\|f^{(m)}(y_n)(x_1, \dots, x_m) - \mathbf{V} f^{(m)}_{\text{sur}}(\mathbf{W}^* y_n)(\mathbf{W}^* x_1, \dots, \mathbf{W}^* x_m)\big\| \leq \varepsilon \|x_1\| \cdots \|x_m\|,$$

for $m = 1, \dots p$. Here $\|\cdot\|$ denotes an arbitrary norm on $\mathbb{C}^{\mathrm{N}}$. This yields the bounds

$$\big\|F\big(\bullet\big)(y_n)\big\| \leq d_0 + \varepsilon, \qquad \Big\|F\Big([\mathfrak{u}_1, \dots, \mathfrak{u}_m]^{\{\mathfrak{f}\}}\Big)(y_n)\Big\| \leq (d_m + \varepsilon) \prod_{i=1}^{m} \|F(u_i)(y_n)\|,$$
$$\big\|F\big(\circ\big)(y_n)\big\| \leq \varepsilon, \qquad \Big\|F\Big([\mathfrak{u}_1, \dots, \mathfrak{u}_m]^{\{\mathfrak{s}\}}\Big)(y_n)\Big\| \leq \varepsilon \prod_{i=1}^{m} \|F(u_i)(y_n)\|,$$

where $d_0 = \|f(y_n)\|$ and d_m is the operator norm of $f^{(m)}(y_n)$. Now we have that

$$\|C\| \leq \sum_{\substack{\mathfrak{u} \in \mathbb{T}_2 \setminus \mathbb{T}_2^{\{\mathfrak{f}\}} \\ \rho(\mathfrak{u}) = p+1}} \left| \frac{1}{\sigma(\mathfrak{u})} \left(\Phi(\mathfrak{u}) - \frac{1}{\gamma(\mathfrak{u})} \right) \right| \varepsilon^{\rho^{\{\mathfrak{s}\}}(\mathfrak{u})} \left(\varepsilon + \max_{i=0,\dots,p} d_i \right)^{\rho^{\{\mathfrak{f}\}}(\mathfrak{u})} = \sum_{i=1}^{p+1} c_i \varepsilon^i,$$

where $\rho^{\{\sigma\}}(\mathfrak{u})$ is the number of σ-colored vertices in $\mathfrak{u}$. The constants c_i only depend on f and the order condition residuals; they are independent of f_{sur}, $\mathbf{V}$, $\mathbf{W}$, H and ε. As one might expect, the local truncation error decreases as the accuracy of the surrogate model and its derivatives increase.

5.5 Construction of Surrogate Models for Accelerating Time Integration

This section discusses several techniques to construct a surrogate model f_{sur} and the associated linear operators $\mathbf{V}$ and $\mathbf{W}^*$ with a computationally favorable balance of accuracy and evaluation cost.

5.5.1 Reduced-Order Models (ROMs)

There are numerous techniques from the reduced-order modeling community that may be used to generate the surrogate models including proper orthogonal decomposition (POD) [147], Krylov-subspace methods [63], reduced basis methods [109], discrete empirical interpolation method [38], and dynamic mode decomposition [144, 154]. Other multi-resolution methods based on Fourier or wavelet transformation [27] could be used to only capture coarse information about the model.

5.5.2 Multimesh Models

Consider a numerical approach to discretize a partial differential equation (PDE) in space over a computational mesh using, for example, the finite element, difference, or volume method. A surrogate model can come from solving the PDE on a coarser mesh or using a lower order spatial discretization. This provides an approximation to capture the "shape" of the solution that is cheaper and also enjoys a less strict Courant–Friedrichs–Lewy (CFL) condition. If the meshes are nested, $\mathbf{W}$ is a subset of columns of the identity matrix, and $\mathbf{V}$ is a sparse interpolation matrix. We note that this strategy closely resembles multigrid methods[19]. The relaxation and prolongation operators, however, have to be chosen such as $\mathbf{W}^*\mathbf{V} = I_{\mathrm{N}\times\mathrm{N}}$, such as the 1D relaxation stencil $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ and the 1D prolongation stencil $\begin{bmatrix} 1/2 & 1 & 1/2 \end{bmatrix}$.

5.5.3 Applying Simplifying Approximations to the Full Model

By linearizing, filtering, averaging, simply ignoring, or otherwise approximating certain terms in f, a computationally inexpensive f_{sur} can be produced. In [37, Section 6], for example, a surrogate model of z-level ocean model is produced by a vertical averaging of barotropic velocities. Consider also a direct N-body simulation with a Barnes-Hut or fast multipole method used as the surrogate model. For this example, the surrogate models happen to have the same state representation as the full model, and therefore, $\mathrm{S} = \mathrm{N}$ and $\mathbf{V} = \mathbf{W} = I_{\mathrm{N}\times\mathrm{N}}$.

5.5.4 Data-Driven Surrogate Models

When sufficient input and output data is available, supervised learning approaches are a viable option to generate approximate models for a system. A variety of techniques have been developed, some depending on system identification and sparse dictionary learning [126], others using neural networks to discover operators and right-hand side functions [113]. Patch data, both in time and space [90] have been used to train machine learning models that can reproduce crude or partial dynamics of the full system. In some cases, the data

driven dynamics reside in the full space, so $\mathrm{S} = \mathrm{N}$ and $\mathbf{V} = \mathbf{W} = I_{\mathrm{N}\times\mathrm{N}}$. In other cases, the dynamics could reside in the dominant modes of the data, in which case $\mathrm{S} < \mathrm{N}$ and $\mathbf{V}$ and $\mathbf{W}$ are determined by the method.

5.6 Numerical Experiments

In order to evaluate the new surrogate model time-steppers, we apply them to a diverse set of ODE test problems equipped with surrogate models. Two of the test problems are used to verify convergence properties (error versus steps) and another two are used to measure the integrators' performance (error versus runtime). The methods considered in the experiments are SM-MRI-GARK Euler from (5.13) and the four methods from appendix D. We compare these with their base methods, which are traditional Runge–Kutta methods, when only the full or only the surrogate model is used. In all the experiments, the error is computed by comparing a numerical solution to a high-accuracy reference solution at the final time. The solution using only the surrogate model resides in the surrogate model subspace, so it is multiplied by $\mathbf{V}$ before computing error.

5.6.1 Quasi-Geostrophic Model and Quadratic ROM

The quasi-geostrophic (QG) equations [52, 54, 55], are a well-studied set of approximations to both atmospheric and ocean flow. The QG equations have a wide range of well-studied lower dimensional approximations. For our use case, a quadratic POD-Galerkin ROM makes an excellent surrogate model.

We follow the formulations of [99, 127], and the same setup as utilized in [110]:

$$\begin{aligned} \frac{\partial \omega}{\partial t} + J(\psi,\omega) - \mathrm{Ro}^{-1}\frac{\partial \psi}{\partial x} &= \mathrm{Re}^{-1}\nabla^2\omega + \mathrm{Ro}^{-1}F, \\ \omega &= -\nabla^2\psi, \\ J(\psi,\omega) &\equiv \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y}. \end{aligned} \tag{5.18}$$

Here ω represents the vorticity, ψ the streamfunction, $\mathrm{Re} = 450$ the Reynolds number, and $\mathrm{Ro} = 0.0036$ the Rossby number. The forcing term is selected to be a symmetric double Gyre to simulate flow in the ocean [99, 127]:

$$F = \sin(\pi(y-1)).$$

The domain for the problem is $\Omega = [0,1]\times[0,2]$. Homogeneous Dirichlet,

$$\psi(x,y) = 0, \quad \forall (x,y) \in \partial\Omega,$$

boundary conditions are used for all time.

For the spatial discretization, a second order finite difference is performed, with the canonical Arakawa approximation [11] performed on the Jacobian term, J. The domain is discretized using 63 points in the x direction and 127 points in the y direction. The discretization begets the discrete stream function variable ψ. The relation between the streamfunction and vorticity is used to define the time derivative of the streamfunction variable $\frac{\partial \psi}{\partial t}$, which we will take as the PDE of interest.

Following [99], and using the method of snapshots [147], we construct basis transformation operators that capture the time-averaged dominant linear modes of the system. The basis transformations are represented by the orthogonal, in L^2, linear operators $\mathbf{V}$, and $\mathbf{W}^*$, where $y_{\text{sur}} = \mathbf{W}^*\psi$ represents the basis transformation from streamfunction space into ROM space. Details as to how these are derived are found in [110].

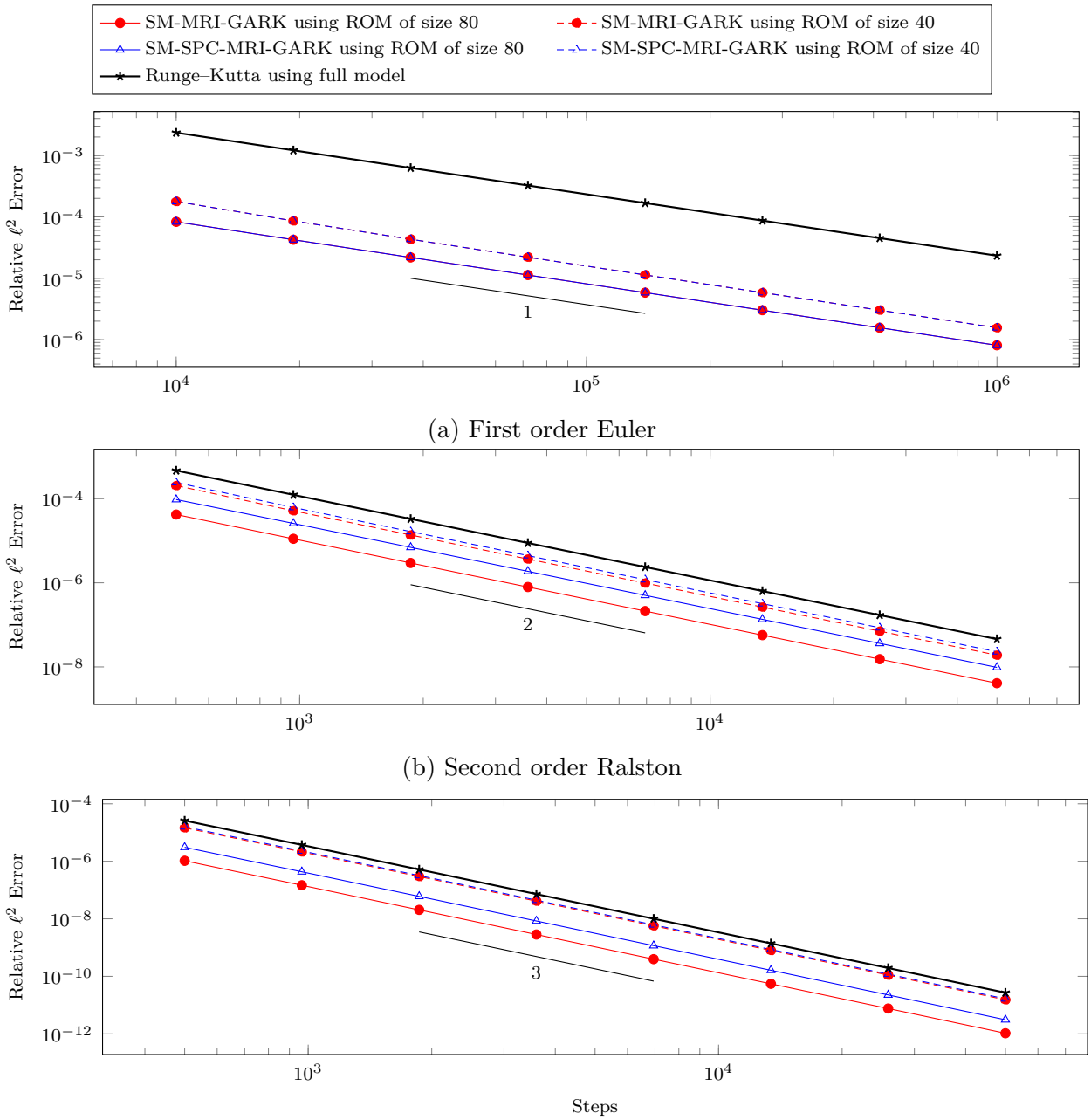
In order to construct a reduced-order model we have to optimally approximate the time derivative of the reduced-order state. As (5.18) is quadratic in nature, we can construct a quadratic POD-Galerkin reduced-order model, of the form

$$y'_{\text{sur}} = f_{\text{sur}}(t, y_{\text{sur}}) = \mathbf{W}^* f(t, \mathbf{V} y_{\text{sur}}) = b + \mathbf{A} y_{\text{sur}} + y_{\text{sur}}^T \mathcal{B} y_{\text{sur}},$$

where y_{sur} is the state in the POD basis, f is the solution to the Poisson equation in (5.18) for the streamfunction derivative $\frac{\partial \psi}{\partial t}$, $b \in \mathbb{R}^{\text{S}}$ corresponds to the forcing term $\text{Ro}^{-1}F$, $\mathbf{A} \in \mathbb{R}^{\text{S}\times\text{S}}$ corresponds to the linear term $\text{Ro}^{-1}\frac{\partial \psi}{\partial x} + \text{Re}^{-1}\nabla^2\omega$, and $\mathcal{B} \in \mathbb{R}^{\text{S}\times\text{S}\times\text{S}}$ corresponds to the Jacobian term $-J(\psi, \omega)$. We take S, the dimension of the reduced-order state space, to be significantly smaller than the dimension of the space of the discretized streamfunction, N. It is known from [110] that for this specific spatial discretization, utilizing $\text{N} = 8001$ points, $\text{S} = 40$ modes corresponds to 98.1% of the total kinetic energy of the system, while $\text{S} = 80$ modes corresponds to 99.4% of the total energy, meaning that this problem should be extremely well-suited to dimensional reduction.

The timespan for which we run the system is a ten-day forecast, which for the given parameters is approximately $t \in [0, 0.109]$. The ROM was built in the interpolatory regime on the test timespan using 2000 evenly-spaced snapshots. The model and the reduced-order model implementations for the experiments have been taken from [119]. As they are written in MATLAB, we use `ode45` with absolute and relative tolerances of 10^{-11} to accurately solve the modified fast ODEs (5.12b) and (5.14b).

Figure 5.2 shows the convergence of the QG equations with respect to a ROM. We have not plotted the results of the Runge–Kutta methods using only the surrogate model as they produced errors orders of magnitude larger than the other methods. For a fixed number of timesteps, SM-MRI-GARK and SM-SPC-MRI-GARK consistently have less error than Runge–Kutta methods using only the full model. At order three, however, the more accurate ROM of size $\text{S} = 80$ is needed to achieve a substantial decrease in error.



(a) First order Euler

(b) Second order Ralston

(c) Third order Ralston

Figure 5.2: Convergence plots for the Quasi-Geostrophic equations (5.18).

5.6.2 Lorenz '96 with a Machine Learning Surrogate Model

The Lorenz '96 problem [98] is given by

$$X_k' = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - X_k + F, \qquad k = 1, 2, \ldots, 40, \tag{5.19}$$

with periodic boundary conditions, and a forcing term $F = 8$. Due to its chaotic dynamics, this problem is prominently used in data assimilation and atmospheric research. Readers interested in current literature on machine learning surrogate models based on Lorenz '96 may refer to [58, 117].

In the following experiments, a neural network model with fully-connected layers was trained to approximate the right-hand side function in (5.19). Therefore, the projections are $\mathbf{V} = \mathbf{W}^* = I_{d\times d}$ and the trained network acts as f_{sur}. The network consists of input and output layers and a hidden layer of dimension 80. The input and hidden layers use the softplus activation function while the output layer does not have any activations. Again, this experiment is implemented in MATLAB and uses `ode45` with tight tolerances to solve ODEs involving the surrogate model.

Similar to [98], the initial condition $X_{20}(0) = 8.008$ and $X_i(0) = 8$ for $i \neq 20$ is propagated for 2 units of time to expel transient effects and the developed solution is used as the true initial condition. The training data is taken from 5000 equally-spaced solutions of the full model within the interval $t \in [2, 10]$ and, the convergence tests are performed over timespan $t \in [4, 8]$.

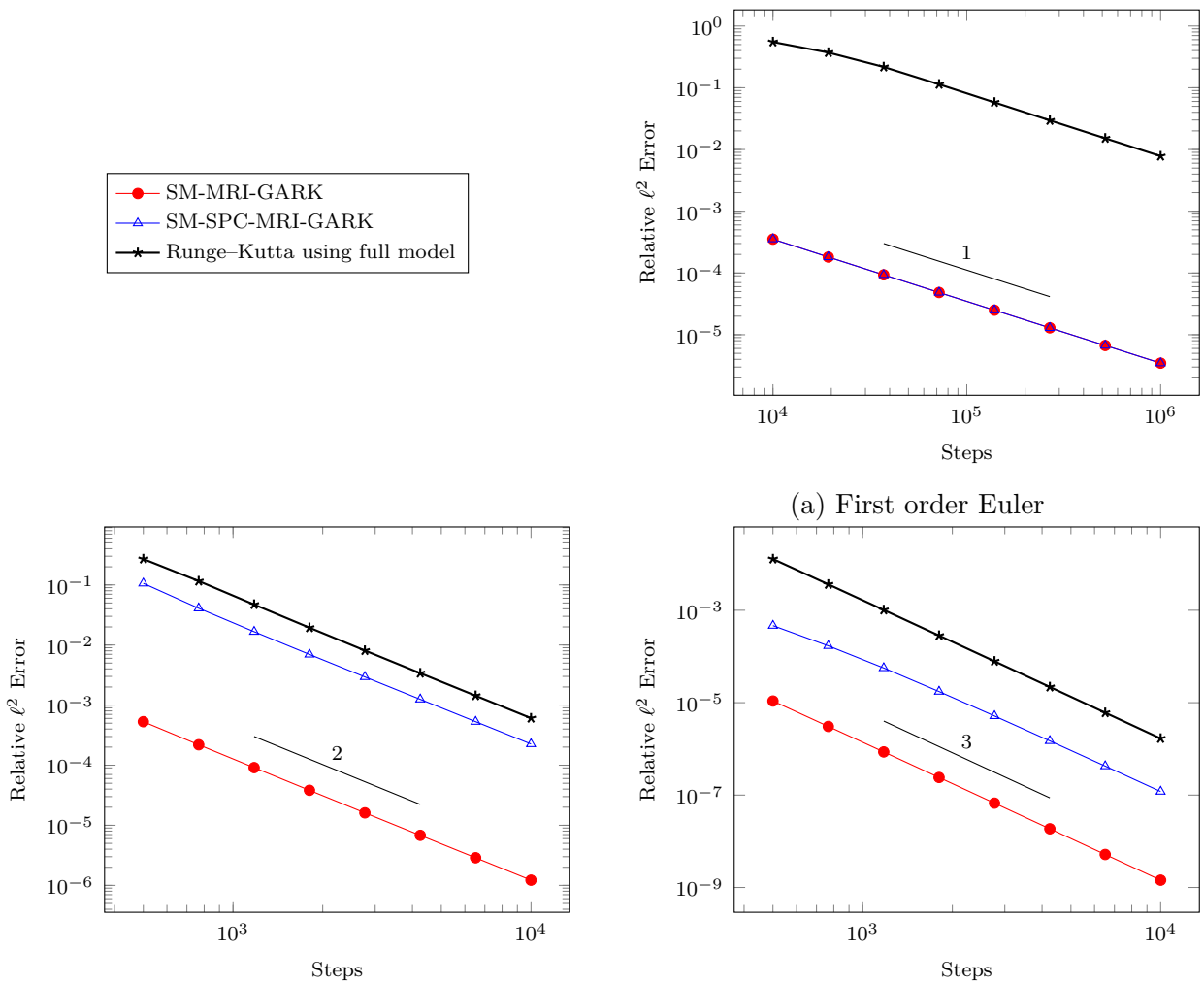
The convergence results are shown in fig. 5.3. We note that merely integrating the surrogate model did not produce a stable solution in any of the experiments. On the other hand, when used together with the full model in SM-MRI-GARK and SM-SPC-MRI-GARK methods, the accuracy of the solution increases significantly compared to the full model solution.

Stepsize Adaptivity

Outside of proposition 5.4, we have only considered fixed values of H across an entire timespan of interest. Some adaptivity has been introduced by using adaptive methods like `ode45` to solve modified fast ODEs. For this section, we present H adaptivity results for SM-MRI-GARK and SM-SPC-MRI-GARK methods applied to the Lorenz '96 problem. We use the error controller from [74, Section II.4] to select H such that the local truncation error is within specified absolute and relative tolerances (`AbsTol` and `RelTol`, respectively).

For the first experiment, we test the second order SM-SPC-MRI-GARK and third order SM-MRI-GARK from appendix D on the Lorenz '96 problem with its machine-learning-based surrogate. Figure 5.4 plots the adaptively chosen H for each step. We can see the value of H varies slightly throughout the timespan and there are reasonably few rejected steps.

For the second experiment we test all methods in appendix D using `AbsTol` = `RelTol` = $10^{-3}, 10^{-4}, \ldots, 10^{-10}$. The (global) error at the end of the timespan is recorded for each tolerance and plotted in fig. 5.5. For reference, we also include traditional, adaptive Runge–Kutta methods using only the full model. As expected, cutting the local error tolerance by a factor of ten causes the global error to decrease by a factor of approximately ten.



(a) First order Euler

(b) Second order Ralston

(c) Third order Ralston

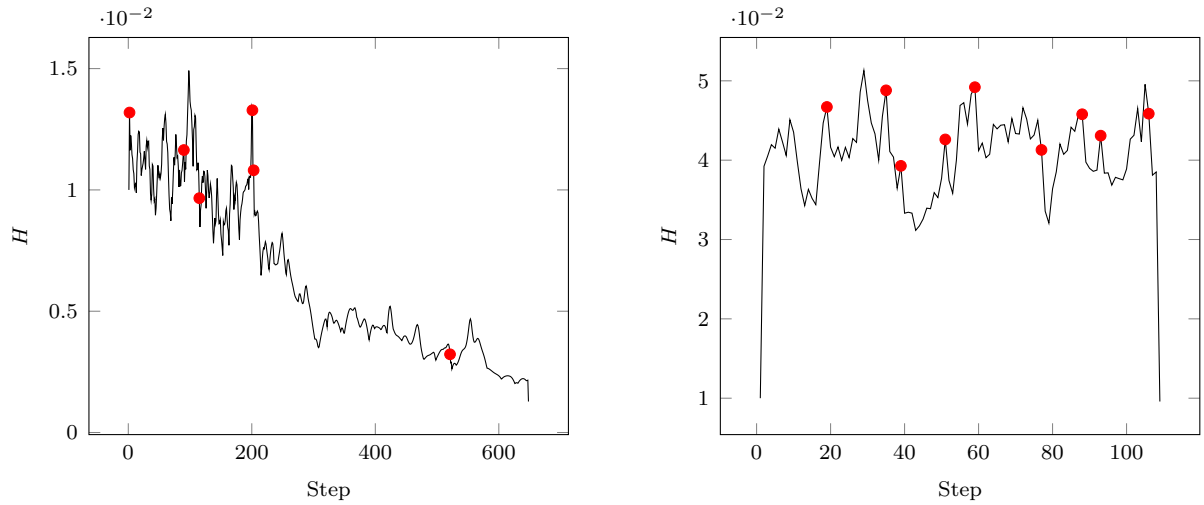
Figure 5.3: Convergence plots for the Lorenz '96 problem (5.19).

5.6.3 Brusselator PDE

The test problem BRUS from [74, Section II.10] is based on the Brusselator reaction-diffusion PDE

$$\begin{aligned} \frac{\partial u}{\partial t} &= \alpha \nabla^2 u + 1 + u^2 v - 4.4u, \\ \frac{\partial v}{\partial t} &= \alpha \nabla^2 v + 3.4u - u^2 v, \end{aligned} \tag{5.20}$$

where $\alpha = 0.002$. The spatial domain is the unit square $0 \leq x, y \leq 1$ with zero, Neumann boundary conditions for both u and v. Using the method of lines, this is discretized with second order central finite difference on a uniform $P \times P$ grid. Starting at the initial



(a) Second order Ralston SM-SPC-MRI-GARK (b) Third order Ralston SM-MRI-GARK

Figure 5.4: Adaptivity selected stepsize H for each step taken to solve the Lorenz '96 problem (5.19) with $\texttt{AbsTol} = \texttt{RelTol} = 10^{-4}$. Rejected steps shown with red markers.

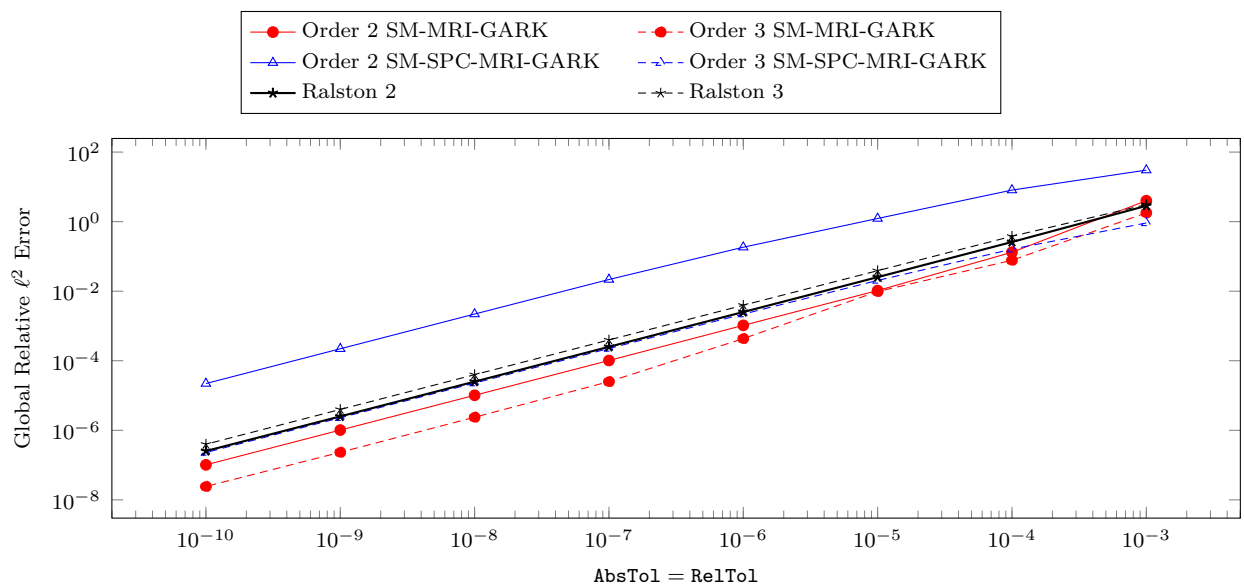


Figure 5.5: Global error versus stepsize controller tolerances for the Lorenz '96 problem (5.19).

conditions

$$u(t = 0, x, y) = 0.5 + y, \qquad v(t = 0, x, y) = 1 + 5x,$$

we seek the solution at $t = 7.5$.

In our experiments, $P = 257$, and thus, $N = 2P^2 = 132098$. For the surrogate model,

we take the approach described in section 5.5.2 where we use the same finite difference discretization of (5.20) but on a coarser mesh. In particular, coarse meshes of size $P = 129$ and $P = 65$ are used, which nest inside the fine mesh. The modified fast ODEs (5.12b) and (5.14b) are solved with *one* step of a Runge–Kutta method one or two orders higher than the base method. We found that this uses the fewest evaluations of f_{sur} while keeping the ODE solution errors negligible.

Comparing the performance of the integrators is our primary goal with the Brusselator problem, and we implemented the tests in C. For each integrator and surrogate model size, the runtime and error were recorded for a range of eight stepsizes. The results are plotted in fig. 5.6. At orders one and two, SM-MRI-GARK and SM-SPC-MRI-GARK show clear speedups over Runge–Kutta solutions using only the full model. In addition, the performance increases as the surrogate model mesh becomes finer. Unfortunately, the results are reversed at order three. Profiling of the code helped to identify why SM-MRI-GARK and SM-SPC-MRI-GARK performed poorly. On a 129×129 grid, evaluations of f_{sur} are about 25% as expensive as evaluation f. The linear operators $\mathbf{V}$ and $\mathbf{W}^*$ have efficient, matrix-free implementations, but are still 75% as expensive as f. These two factors caused one step of SM-MRI-GARK and SM-SPC-MRI-GARK to take approximately twice as long as a traditional Runge–Kutta step. At order three, the reduction in error is not enough to overcome this large overhead.

5.6.4 Advection PDE

Finally, we consider the following 2D advection problem with zero, Dirichlet boundary conditions:

$$\begin{aligned} \frac{\partial u}{\partial t} + a \cdot \nabla u &= 0, \qquad \text{on } \Omega = [0,1] \times [0,1], \\ u(t,x,y) &= 0, \qquad \text{on } \partial\Omega. \end{aligned}$$

The velocity field corresponds to the Molenkamp-Crowley problem:

$$a(x,y) = \left[2\pi\left(y - \tfrac{1}{2}\right), -2\pi\left(x - \tfrac{1}{2}\right)\right].$$

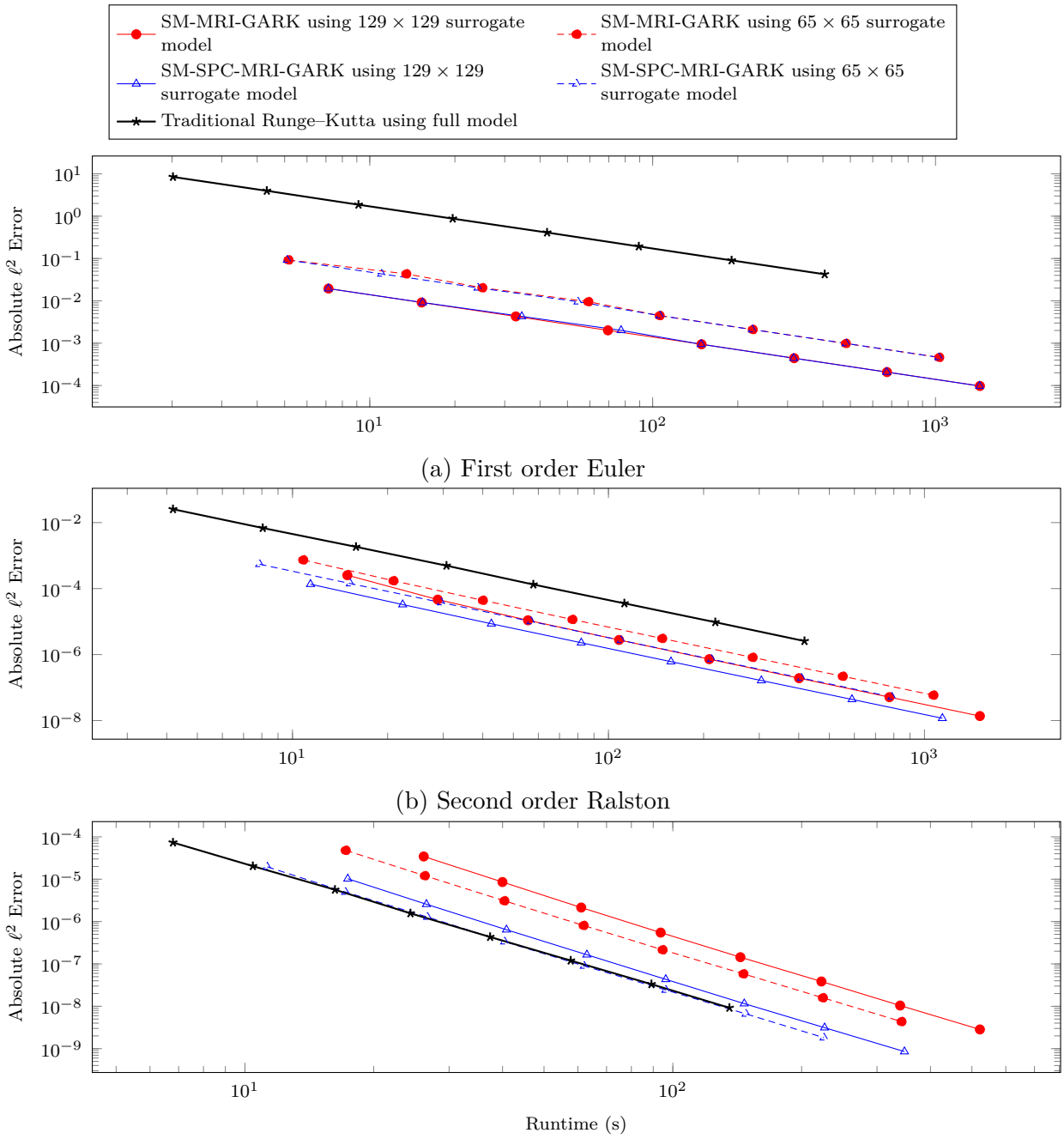
We start with the initial condition

$$u(t=0,x,y) = \exp\!\left(-100\left((x-0.35)^2 + (y-0.35)^2\right)\right)$$

and stop at $t = 2$. This allows the initial profile to rotate clockwise twice about the center of the domain.

For the method of lines discretization in space, we represent Ω with a $P \times P$ uniform, triangular mesh and apply a nodal discontinuous Galerkin (DG) method. The order in space is chosen to match the order in time. This yields the linear ODE

$$y' = \mathbf{M}^{-1}\,\mathbf{K}\,y, \tag{5.21}$$



(a) First order Euler

(b) Second order Ralston

(c) Third order Ralston

Figure 5.6: Work precision diagrams for BRUS (5.20).

where $\mathbf{M}$ and $\mathbf{K}$ are mass and advection matrices, respectively. We use the C++ library MFEM [8] for this DG discretization, and in particular, base it on the serial version of example 9 provided in MFEM version 4.1.

Again, we try the multimesh approach by using a mesh size of $P = 100$ for f and a mesh size of $P = 50$ for f_{sur}. We use the same technique discussed in section 5.6.3 for integrating modified fast ODEs involving f_{sur}. The size of the ODE (5.21) for each experiment is listed in table 5.2.

Order of time and space discretization	Dimension of full model N	Dimension of surrogate model S
1	6×10^4	1.5×10^4
2	1.2×10^5	3×10^4
3	2×10^5	5×10^4

Table 5.2: Dimensions of the full and surrogate models used in the advection experiment.

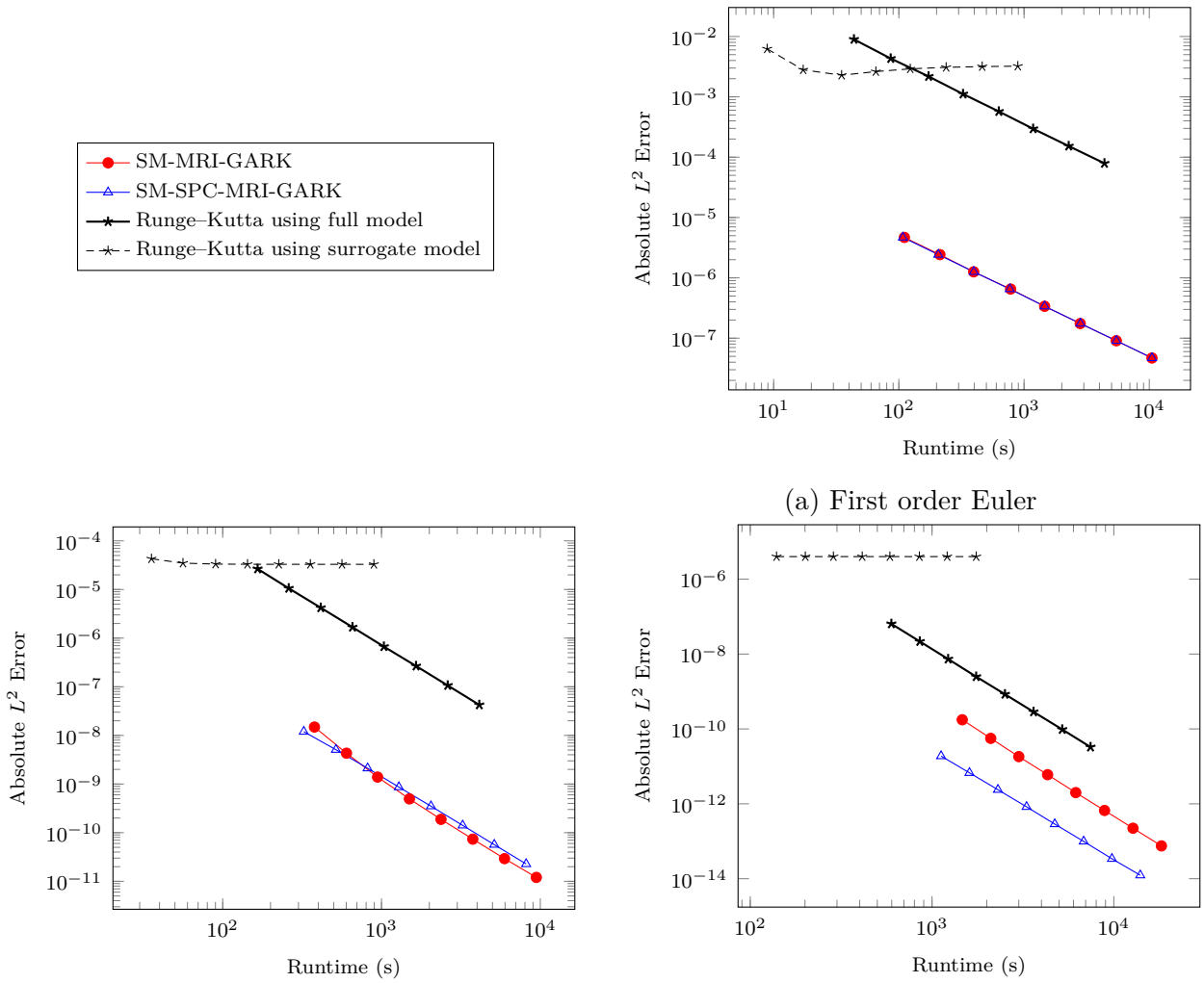
In contrast to the Brusselator problem (5.20), the advection problem is linear and hyperbolic. Moreover, profiling reveals an evaluation of the RHS of (5.21) is much more expensive than an interpolation between meshes due, in part, to the linear solve with the mass matrix. Based on the discussion in the previous subsection, we may expect better performance results at order three, and indeed, that is the case.

Figure 5.7 shows the error and timing for the integrators over a range of eight stepsizes. All SM-MRI-GARK and SM-SPC-MRI-GARK methods outperform Runge–Kutta methods with speedups ranging from approximately 3 to 725. The error for the Runge–Kutta methods using the surrogate model remains nearly constant, which indicates that spatial errors dominate temporal errors.

5.7 Conclusions

This work develops a new methodology to accelerate the time integration of large ODEs using surrogate models. Specifically, surrogate information is incorporated into the numerical solution of the original ODEs using multirate methods. We derive and analyze two implementations of this new time-stepping technique: SM-MRI-GARK and SM-SPC-MRI-GARK. Both combine continuous evolution of the surrogate model with discrete Runge–Kutta steps of the full ODE. There are numerous ways to generate surrogate models that offer flexible trade-offs among accuracy, computational cost, and stability. The new methods are designed such that the overall order of accuracy is independent of the surrogate model. The more accurate the surrogate models are, the smaller the local error constants are, which leads to a smaller global error.

Numerical experiments reveal that, at low orders of accuracy, it is possible to achieve orders of magnitude speedups over traditional Runge–Kutta methods. As the order increases, overheads associated with evaluations of $\mathbf{V}$, $\mathbf{W}^*$, and the surrogate model are penalized more when measuring efficiency, and speedups tend to decrease. There is not a clear winner



(a) First order Euler

(b) Second order Ralston

(c) Third order Ralston

Figure 5.7: Work precision diagrams for advection problem (5.21).

between SM-MRI-GARK and SM-SPC-MRI-GARK; their relative performance appears to depend heavily on the properties of the full and surrogate models.

The methods considered in this paper are based on explicit multirate infinitesimal methods, although the surrogate integration can be done implicitly. Even so, the explicit treatment of the surrogate model error can lead to stepsize restrictions. Instability can occur if a surrogate model does not capture the stiffness of the full model. The authors plan to explore implicit and implicit-explicit methods [35, 164, 165, 167] for stiff problems and differential-algebraic equations. We note that stiffness can appear in the full model, surrogate model, and even the surrogate model error, and each scenario brings different considerations.

Chapter 6

Eliminating Order Reduction on Linear, Time-Dependent ODEs with GARK Methods

6.1 Introduction

Consider the linear, inhomogeneous system of ordinary differential equations

$$y' = f(t, y) = Ly + g(t), \quad y(t_0) = y_0, \quad t \in [t_0, t_f], \tag{6.1}$$

where $y \in \mathbb{C}^d$. Problems of this form frequently arise from the spatial discretization of linear partial differential equations (PDEs). In this case, L approximates differential operators and $g(t)$ accounts for source terms and boundary conditions.

Runge–Kutta methods are some of the most widely used to integrate (6.1). One step of an s-stage Runge–Kutta method using timestep h is given by

$$Y_i = y_n + h \sum_{j=1}^{s} a_{i,j} f(t_n + c_j h, Y_j), \qquad i = 1, \ldots, s, \tag{6.2a}$$

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_i f(t_n + c_j h, Y_j), \tag{6.2b}$$

and its coefficients are concisely represented by the Butcher tableau:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}. \tag{6.3}$$

An important special case of (6.1) is the Prothero–Robinson (PR) test problem [111]:

$$y' = \lambda(y - \phi(t)) + \phi'(t), \qquad y(0) = \phi(0). \tag{6.4}$$

In their seminal work, Prothero and Robinson analyzed the error and stability of Runge–Kutta methods applied to (6.4) as $\mathrm{Re}(h\lambda) \to -\infty$ and $h \to 0$. For this seemingly innocuous problem, A-stability is not sufficient to guarantee a stable numerical solution to (6.4).

Moreover, the order of convergence for a Runge–Kutta method may be lower than what is predicted by classical order condition theory: a phenomenon referred to as *order reduction*. Classical order condition theory typically requires f to have a moderate Lipschitz constant that is independent of the timestep h, and for the PR problem, these assumptions do not hold. The analysis of order reduction has been extended to many other classes of problems including linear PDEs [103, 134, 160] and general, nonlinear problems [28, 57].

It is well-known that the Runge–Kutta simplifying assumptions

$$B(p): \quad b^T c^{k-1} = \frac{1}{k}, \quad k = 1, \ldots, p, \tag{6.5a}$$

$$C(q): \quad Ac^{k-1} = \frac{c^k}{k}, \quad k = 1, \ldots, q, \tag{6.5b}$$

mitigate the order reduction phenomenon [72, Section IV.15]. A method satisfying $B(p)$ and $C(q)$ with $p \geq q$ is said to have stage order q. Ideally, a method would have the stage order equal to the classical order, but in many cases, this cannot be achieved. Explicit Runge–Kutta methods, for example, have a maximum stage order of one, while diagonally implicit methods have a maximum stage order of two. The concept of weak stage order (WSO) has been explored in [89]. As the name suggests, it considers weaker but sufficient conditions to avoid order reduction. Unfortunately, diagonally implicit methods cannot have weak stage order greater than three. In [103, 104], the authors derive a rigorous error expansion and order conditions for stiff, parabolic PDEs. Similar results have been derived for the PR problem in [115, 116].

An approach used to address order reduction in initial boundary value problems is a modified treatment of the boundary conditions in the stages (6.2a) [1, 6, 7, 106]. Many of these utilize time derivatives of the boundary conditions which would not be required in a traditional Runge–Kutta stage. One can interpret this as a composite method where a Runge–Kutta method is used to treat the differential operators, and a multi-derivative scheme is used to treat the boundary conditions.

In this work, we propose integrating (6.1) using a general-structure additive Runge–Kutta (GARK) method [132] to eliminate order reduction. For an arbitrary Runge–Kutta method used to treat the linear term Ly, it can be paired with a different, fully implicit Runge–Kutta scheme used to treat the forcing $g(t)$. This approach does not require time derivatives of $g(t)$, and in certain cases, requires fewer evaluations of $g(t)$ per step than a traditional Runge–Kutta step. Further, unlike (weak) stage order conditions, there are no restrictions on the order for explicit or diagonally implicit method structures.

The remainder of this paper is organized as follows. In section 6.2, the new GARK-based method for (6.1) is derived. Section 6.3 contains the error analysis and order condition results. Sections 6.4 to 6.6 provide three numerical experiments to test convergence properties for Runge–Kutta methods and their GARK extensions. Finally, we summarize our findings of the paper in section 6.7.

6.2 Method Formulation

We will consider a splitting of (6.1) into the linear term and the time-dependent forcing term:

$$y' = \underbrace{Ly}_{f^{\{1\}}(t,y)} + \underbrace{g(t)}_{f^{\{2\}}(t,y)} . \tag{6.6}$$

When a general, two-partitioned GARK scheme is applied to (6.6), we arrive at the following procedure:

$$\begin{aligned}
Y_i^{\{1\}} &= y_n + h\sum_{j=1}^{s^{\{1\}}} a_{i,j}^{\{1,1\}} LY_j^{\{1\}} + h\sum_{j=1}^{s^{\{2\}}} a_{i,j}^{\{1,2\}} g\Big(t_n + c_j^{\{2\}}h\Big), \qquad i = 1,\dots s^{\{1\}},\\
Y_i^{\{2\}} &= y_n + h\sum_{j=1}^{s^{\{1\}}} a_{i,j}^{\{2,1\}} LY_j^{\{1\}} + h\sum_{j=1}^{s^{\{2\}}} a_{i,j}^{\{2,2\}} g\Big(t_n + c_j^{\{2\}}h\Big), \qquad i = 1,\dots s^{\{2\}},\\
y_{n+1} &= y_n + h\sum_{j=1}^{\{1\}} b_j^{\{1\}} LY_j^{\{2\}} + h\sum_{j=1}^{s^{\{2\}}} b_j^{\{2\}} g\Big(t_n + c_j^{\{2\}}h\Big).
\end{aligned} \tag{6.7}$$

In contrast (6.2), the stages are partitioned, there are four A coefficient matrices used in the stages, and two sets of b coefficients. Note that the computation of y_{n+1} in (6.7) does not involve $Y_i^{\{2\}}$. With $Y_i^{\{1\}}$ serving as the only useful stages, it may appear that (6.7) degenerates into an additive Runge–Kutta (ARK) method which does not have partitioned stages. This is not the case, however, as the GARK formalism allows the additional flexibility of treating the linear term and forcing terms with a different number of stages. That is, $\mathbf{A}^{\{1,2\}}$ can be rectangular.

We can rewrite (6.7) into the compact form

$$Y^{\{1\}} = \mathbb{1}_{s^{\{1\}}} \otimes y_n + \left(\mathbf{A}^{\{1,1\}} \otimes Z\right) Y^{\{1\}} + h\left(\mathbf{A}^{\{1,2\}} \otimes I_{d\times d}\right) g\big(t_n + \mathbf{c}^{\{2\}}h\big), \tag{6.8a}$$

$$y_{n+1} = y_n + \left(\mathbf{b}^{\{1\}T} \otimes Z\right) Y^{\{1\}} + h\left(\mathbf{b}^{\{2\}T} \otimes I_{d\times d}\right) g\big(t_n + \mathbf{c}^{\{2\}}h\big), \tag{6.8b}$$

where $\otimes$ denotes the Kronecker product, $\mathbb{1}_{s^{\{1\}}}$ is a vector of ones of dimension $s^{\{1\}}$, and $Z = hL$. We also use the notation

$$Y^{\{1\}} = \begin{bmatrix} Y_1^{\{1\}} \\ \vdots \\ Y_{s^{\{1\}}}^{\{1\}} \end{bmatrix}, \qquad g\big(t_n + \mathbf{c}^{\{2\}}h\big) = \begin{bmatrix} g\Big(t_n + c_1^{\{2\}}h\Big) \\ \vdots \\ g\Big(t_n + c_{s^{\{2\}}}^{\{2\}}h\Big) \end{bmatrix}.$$

For the simplified method (6.8), we will represent it with the tableau

$$\begin{array}{c|c}
\mathbf{c}^{\{1\}T} & \mathbf{c}^{\{2\}T} \\ \hline
\mathbf{A}^{\{1,1\}} & \mathbf{A}^{\{1,2\}} \\ \hline
\mathbf{b}^{\{1\}T} & \mathbf{b}^{\{2\}T}
\end{array} . \tag{6.9}$$

In this paper, we define $\mathbf{c}^{\{1\}} = \mathbf{A}^{\{1,1\}}\mathbb{1}_{s^{\{1\}}}$ and refer to $(\mathbf{A}^{\{1,1\}}, \mathbf{b}^{\{1\}}, \mathbf{c}^{\{1\}})$ as the *base method*.

The implicitness of (6.8) is entirely determined by the structure of $\mathbf{A}^{\{1,1\}}$. With $f^{\{2\}}$ only a function of time, we can make $\mathbf{A}^{\{1,2\}}$ a dense matrix without incurring additional function evaluations.

6.3 Order Conditions

6.3.1 Classical Order Conditions

For sufficiently small h, we can rely on traditional, tree-based GARK order condition theory to determine the local truncation error of the new method (6.8). We present these order conditions in the following theorem.

Theorem 6.1 (Classical order conditions)**.** *The method* (6.8) *applied to* (6.1) *has classical order p if and only if*

$$\mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,1\}\times(k-1)}\mathbb{1}_{s^{\{1\}}} = \frac{1}{k!}, \qquad 1 \le k \le p \tag{6.10a}$$

$$\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times(k-1)} = \frac{1}{k}, \qquad 1 \le k \le p \tag{6.10b}$$

$$\mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,1\}\times(k-1)}\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times(\ell-1)} = \frac{(\ell-1)!}{(\ell+k)!}, \qquad 1 \le k, \ell \text{ and } k+\ell \le p. \tag{6.10c}$$

Proof. We will start by converting (6.6) into the autonomous form

$$\widetilde{y}' = \underbrace{\begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix}\widetilde{y}}_{\widetilde{f}^{\{1\}}(\widetilde{y})} + \underbrace{\begin{bmatrix} g(t) \\ 1 \end{bmatrix}}_{\widetilde{f}^{\{2\}}(\widetilde{y})}, \qquad \widetilde{y} = \begin{bmatrix} y \\ t \end{bmatrix} \in \mathbb{R}^{d+1}, \tag{6.11}$$

which comes at no loss of generality. From N-tree order condition theory [12, 132], a GARK method is order p if and only if

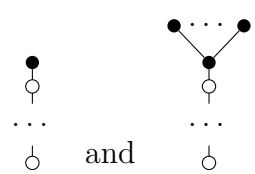
$$\sum_{\substack{\mathfrak{t}\in 2T \\ \rho(\mathfrak{t})\le p}} \left(\Phi(\mathfrak{t}) - \frac{1}{\gamma(\mathfrak{t})}\right) F(\mathfrak{t})(\widetilde{y}) = 0,$$

where $2T$ is the set of 2-trees, and ρ, Φ, γ are the order, elementary weight, and density of a tree, respectively. In trees, we use white vertices ($\circ$) for partition 1 and black vertices ($\bullet$)

for partition 2. The elementary differentials for (6.11) simplify to

$$F(\mathfrak{t})(\widetilde{y}) = \begin{cases} \widetilde{y}, & \text{if } \mathfrak{t} = \emptyset, \\ \begin{bmatrix} g(t) \\ 1 \end{bmatrix}, & \text{if } \mathfrak{t} = \bullet, \\ \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} F(\mathfrak{u})(\widetilde{y}), & \text{if } \mathfrak{t} = \overset{\mathfrak{u}}{\circ}, \\ \begin{bmatrix} g^{(m)}(t) \\ 0 \end{bmatrix}, & \text{if } \mathfrak{t} = \overbrace{\bullet \cdots \bullet}^{m}, \\ 0_{d+1}, & \text{otherwise.} \end{cases}$$

For the elementary differentials that do not vanish, we split their corresponding trees into three sets. The first are trees where all vertices are white and singly-branched. By considering their GARK elementary weights and densities, we recover the order conditions (6.10a). Similarly, bushy trees with all black vertices give (6.10b). Finally, trees of the form



correspond to (6.10c). □

6.3.2 Stiff Order Conditions

Before the asymptotic regime is reached, a method satisfying (6.10) may exhibit an order of convergence less than p. In order to characterize this behavior, we will reexamine the local truncation error produced at each step and the accumulated global error. We define the global error at step n in the stages (6.8a) and step (6.8b) to be

$$E_n = y\big(t_n + \mathbf{c}^{\{1\}}h\big) - Y^{\{1\}} = \begin{bmatrix} y(t_n + c_1 h) - Y_1^{\{1\}} \\ \vdots \\ y(t_n + c_{s^{\{1\}}} h) - Y_{s^{\{1\}}}^{\{1\}} \end{bmatrix} \quad \text{and} \quad e_n = y(t_n) - y_n,$$

respectively.

In (6.8b), we replace the initial condition y_n with $y(t_n)$ and replace the stages Y with

$y\big(t_n + \mathbf{c}^{\{1\}}h\big)$:

$$\begin{aligned} y\big(t_n + \mathbf{c}^{\{1\}}h\big) &= \mathbb{1}_{s^{\{1\}}} \otimes y(t_n) + \big(\mathbf{A}^{\{1,1\}} \otimes Z\big)\, y\big(t_n + \mathbf{c}^{\{1\}}h\big) \\ &\quad + h\,\big(\mathbf{A}^{\{1,2\}} \otimes I_{d\times d}\big)\, g\big(t_n + \mathbf{c}^{\{2\}}h\big) + \Delta_n \\ &= \mathbb{1}_{s^{\{1\}}} \otimes y(t_n) + \big(\mathbf{A}^{\{1,1\}} \otimes Z\big)\, y\big(t_n + \mathbf{c}^{\{1\}}h\big) \\ &\quad + h\,\big(\mathbf{A}^{\{1,2\}} \otimes I_{d\times d}\big)\, y'\big(t_n + \mathbf{c}^{\{2\}}h\big) - \big(\mathbf{A}^{\{1,2\}} \otimes Z\big)\, y\big(t_n + \mathbf{c}^{\{2\}}h\big) + \Delta_n. \end{aligned} \tag{6.12}$$

The stage defect Δ_n has the expansion

$$\begin{aligned} \Delta_n = &\sum_{k=1}^{\infty} \frac{\mathbf{c}^{\{1\}k} - k\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times(k-1)}}{k!} \otimes h^k y^{(k)}(t_n) \\ &+ \sum_{k=0}^{\infty} \frac{\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times k} - \mathbf{A}^{\{1,1\}}\mathbf{c}^{\{1\}\times k}}{k!} \otimes \big(h^k Z y^{(k)}(t_n)\big)\,. \end{aligned}$$

The global error in the stages follows by subtracting (6.12) from (6.8a):

$$\begin{aligned} E_n &= \mathbb{1}_{s^{\{1\}}} \otimes e_n + \big(\mathbf{A}^{\{1,1\}} \otimes Z\big)\, E_n + \Delta_n \\ &= \big(I_{ds^{\{1\}}\times ds^{\{1\}}} - \mathbf{A}^{\{1,1\}} \otimes Z\big)^{-1} \big(\mathbb{1}_{s^{\{1\}}} \otimes e_n + \Delta_n\big)\,. \end{aligned}$$

Similar to (6.12), the exact solution also satisfies

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + \big(\mathbf{b}^{\{1\}T} \otimes Z\big)\, y\big(t_n + \mathbf{c}^{\{1\}}h\big) + h\,\big(\mathbf{b}^{\{2\}T} \otimes I_{d\times d}\big)\, y'\big(t_n + \mathbf{c}^{\{2\}}h\big) \\ &\quad - \big(\mathbf{b}^{\{2\}T} \otimes Z\big)\, y\big(t_n + \mathbf{c}^{\{2\}}h\big) + \delta_n, \\ \delta_n &= \sum_{k=1}^{\infty} h^k \frac{1 - k\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times(k-1)}}{k!} y^{(k)}(t_n) + \sum_{k=0}^{\infty} h^k \frac{\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times k} - \mathbf{b}^{\{1\}T}\mathbf{c}^{\{1\}\times k}}{k!} Z y^{(k)}(t_n). \end{aligned}$$

Finally, the global error recurrence is

$$e_{n+1} = e_n + \big(\mathbf{b}^{\{1\}T} \otimes Z\big)\, E_n + \delta_n = R^{\{1\}}(Z) e_n + \mathrm{lte}_n, \tag{6.13}$$

where

$$R^{\{1\}}(z) = 1 + z\mathbf{b}^{\{1\}} \left(I_{s^{\{1\}}\times s^{\{1\}}} - z\mathbf{A}^{\{1,1\}}\right) \mathbb{1}_{s^{\{1\}}} \tag{6.14}$$

is the linear stability function of the base method, and the local truncation error at step n is

$$\mathrm{lte}_n = \big(\mathbf{b}^{\{1\}T} \otimes Z\big) \big(I_{ds^{\{1\}}\times ds^{\{1\}}} - \mathbf{A}^{\{1,1\}} \otimes Z\big)^{-1} \Delta_n + \delta_n \tag{6.15a}$$

$$= \sum_{k=0}^{\infty} W_k(Z) \frac{h^k}{k!} y^{(k)}(t_n). \tag{6.15b}$$

The local error residuals at order k are defined as

$$\begin{aligned} W_0(z) &= z\left(\mathbf{b}^{\{2\}T}\mathbb{1}_{s^{\{2\}}} - \mathbf{b}^{\{1\}T}\mathbb{1}_{s^{\{1\}}}\right) + z^2\mathbf{b}^{\{1\}T}\left(I_{s^{\{1\}}\times s^{\{1\}}} - z\mathbf{A}^{\{1,1\}}\right)^{-1}\left(\mathbf{A}^{\{1,2\}}\mathbb{1}_{s^{\{2\}}} - \mathbf{A}^{\{1,1\}}\mathbb{1}_{s^{\{1\}}}\right), \\ W_k(z) &= 1 + \left(\mathbf{b}^{\{2\}T} + z\mathbf{b}^{\{1\}T}\left(I_{s^{\{1\}}\times s^{\{1\}}} - z\mathbf{A}^{\{1,1\}}\right)^{-1}\mathbf{A}^{\{1,2\}}\right)\left(z\mathbf{c}^{\{2\}\times k} - k\mathbf{c}^{\{2\}\times(k-1)}\right), \qquad k \geq 1. \end{aligned} \tag{6.16}$$

For simplicity, both (6.14) and (6.16) have been written in a scalar form but are rational, matrix functions of Z.

We can expand (6.15b) further by expressing it as a multivariate series in h and Z:

$$\mathrm{lte}_n = \sum_{k,\ell=0}^{\infty} w_{k,\ell}\frac{h^k Z^\ell}{k!}y^{(k)}(t_n). \tag{6.17}$$

The coefficients $w_{k,\ell}$ are found by taking a Maclaurin series of $W_k(z)$:

$$w_{k,\ell} = \begin{cases} 0, & k=0, \ell=0, \\ \mathbf{b}^{\{2\}T}\mathbb{1}_{s^{\{2\}}} - \mathbf{b}^{\{1\}T}\mathbb{1}_{s^{\{1\}}}, & k=0, \ell=1, \\ \mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,1\}\times(\ell-2)}\left(\mathbf{A}^{\{1,2\}}\mathbb{1}_{s^{\{2\}}} - \mathbf{A}^{\{1,1\}}\mathbb{1}_{s^{\{1\}}}\right), & k=0, \ell>1, \\ 1 - k\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times(k-1)}, & k>0, \ell=0, \\ \mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times k} - k\mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times(k-1)}, & k>0, \ell=1, \\ \mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,1\}\times(\ell-2)}\left(\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times k} - k\mathbf{A}^{\{1,1\}}\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times(k-1)}\right), & k>0, \ell>1. \end{cases} \tag{6.18}$$

For $Z = \mathcal{O}(h)$, i.e., a nonstiff problem, the order condition results are summarized in the following diagram:

$$\begin{array}{ccc} \mathrm{lte}_n = \mathcal{O}(h^{p+1}) & \Longleftrightarrow & W_k(z) = \mathcal{O}\big(h^{p+1-k}\big) \text{ for } 0 \leq k \leq p \\ \Updownarrow & & \Updownarrow \\ (6.10) & \Longleftrightarrow & w_{k,\ell} = 0 \text{ for } k,\ell \geq 0 \text{ and } k+\ell \leq p \end{array}$$

As expected, we can recover the tree-based order conditions given in (6.10) from (6.17).

For stiff problems, however, h and Z can have more complex relations which necessitates more stringent order conditions. Instead of canceling low order terms of $W_k(z)$, it is possible to completely eliminate the term.

Theorem 6.2. *$W_k(z) \equiv 0$ if and only if $w_{k,\ell} = 0$ for $\ell = 0, \ldots, s^{\{1\}}+1$.*

Proof. Note that we can express (6.16) as

$$W_k(z) = \frac{n_{k,0} + n_{k,1}z + \cdots + n_{k,s^{\{1\}}+1}z^{s^{\{1\}}+1}}{d_{k,0} + d_{k,1}z + \cdots + d_{k,s^{\{1\}}}z^{s^{\{1\}}}} = \sum_{\ell=0}^{\infty} w_{k,\ell}z^\ell, \tag{6.19}$$

where $d_{k,0} \neq 0$.

($\Leftarrow$) Suppose that $w_{k,\ell} = 0$ for $\ell = 0, \ldots, s^{\{1\}}+1$. From (6.19), $n_{k,i} = \sum_{j=0}^{\min(i,s^{\{1\}})} w_{k,i-j} d_{k,j} = 0$ for $i = 0, \ldots, s^{\{1\}}+1$. Thus, $W_k(z) \equiv 0$.

($\Rightarrow$) For the order direction of the proof, it is clear that if $W_k(z) \equiv 0$, the Maclaurin series coefficients $w_{k,\ell} = 0$ for $\ell \geq 0$. $\square$

Following the idea of Prothero and Robinson, we can also examine the error when $Z \to -\infty$. In this limit, we cannot rely on the power series expansion in Z used in (6.17), and instead, we will consider a Laurent series in Z:

$$\mathrm{lte}_n = \sum_{k=0}^{\infty} \sum_{\ell=-1}^{\infty} x_{k,\ell} \frac{h^k Z^{-\ell}}{k!} y^{(k)}(t_n).$$

As an intermediate step in the expansion of (6.16), note that

$$z \left(I_{s^{\{1\}} \times s^{\{1\}}} - z\mathbf{A}^{\{1,1\}} \right)^{-1} \mathbf{A}^{\{1,2\}} = -\sum_{\ell=0}^{\infty} z^{-\ell} \mathbf{\Omega}^{\ell+1} \mathbf{A}^{\{1,2\}},$$

where

$$\mathbf{\Omega} = \begin{cases} \mathbf{A}^{\{1,1\}\,-1}, & \text{if } \mathbf{A}^{\{1,1\}} \text{ invertible,} \\ \begin{bmatrix} 0 & 0 \\ 0 & \left(\mathbf{A}^{\{1,1\}}_{2:s^{\{1\}},2:s^{\{1\}}} \right)^{-1} \end{bmatrix}, & \text{if } \mathbf{A}^{\{1,1\}}_{2:s^{\{1\}},2:s^{\{1\}}} \text{ invertible and } \mathbf{A}^{\{1,\sigma\}}_{1,i} = 0, \end{cases}$$

which accounts for methods with an explicit first stage like ESDIRK and Lobatto IIIA schemes. Substituting this into (6.16) yields

$$x_{k,\ell} = \begin{cases} \mathbf{b}^{\{1\}T} \mathbf{\Omega}^{\ell+2} \left(\mathbf{A}^{\{1,1\}} \mathbb{1}_{s^{\{1\}}} - \mathbf{A}^{\{1,2\}} \mathbb{1}_{s^{\{2\}}} \right), & k = 0, \ell \geq 0, \\ \left(\mathbf{b}^{\{2\}T} - \mathbf{b}^{\{1\}T} \mathbf{\Omega} \mathbf{A}^{\{1,2\}} \right) \mathbf{c}^{\{2\} \times k}, & k \geq 0, \ell = -1, \\ 1 - k \left(\mathbf{b}^{\{2\}T} - \mathbf{b}^{\{1\}T} \mathbf{\Omega} \mathbf{A}^{\{1,2\}} \right) \mathbf{c}^{\{2\} \times (k-1)} & k > 0, \ell = 0, \\ \quad - \mathbf{b}^{\{1\}T} \mathbf{\Omega}^2 \mathbf{A}^{\{1,2\}} \mathbf{c}^{\{2\} \times k}, & \\ \mathbf{b}^{\{1\}T} \mathbf{\Omega}^{\ell+2} \left(\mathbf{A}^{\{1,2\}} \mathbf{c}^{\{2\} \times k} - k \mathbf{A}^{\{1,1\}} \mathbf{A}^{\{1,2\}} \mathbf{c}^{\{2\} \times (k-1)} \right), & k > 0, \ell > 0. \end{cases} \tag{6.20}$$

Unless $x_{k,-1} = 0$ for $k \geq 0$, lte_n diverges as $|Z| \to \infty$. Equation (6.20) suggests the following sufficient condition to ensure $w_k(z)$ is bounded away from its poles:

$$\mathbf{b}^{\{2\}T} = \mathbf{b}^{\{1\}T} \mathbf{\Omega} \mathbf{A}^{\{1,2\}}. \tag{6.21}$$

If (6.8) is stiffly accurate [132, Definition 3.3], that is

$$e^T_{s^{\{1\}}} \mathbf{A}^{\{1,1\}} = \mathbf{b}^{\{1\}T} \quad \text{and} \quad e^T_{s^{\{1\}}} \mathbf{A}^{\{1,2\}} = \mathbf{b}^{\{2\}T}, \tag{6.22}$$

then (6.21) is automatically satisfied.

6.3.3 Simplifying Assumptions

Extensions of traditional Runge–Kutta simplifying assumptions (6.5) to the GARK framework have been proposed in [153]. Quadrature simplifying assumptions are defined as

$$B^{\{\sigma\}}(p): \quad \mathbf{b}^{\{\sigma\}T}\mathbf{c}^{\{\sigma\}\times(k-1)} = \frac{1}{k}, \quad k = 1, \dots, p.$$

A method satisfying $B^{\{\sigma\}}(1)$ for all σ is said to be consistent with (6.1). This condition is both necessary and sufficient for classical first order convergence. The stage order simplifying assumption in (6.5b) extends to

$$C^{\{\sigma,\mu\}}(q): \quad \mathbf{A}^{\{\sigma,\mu\}}\mathbf{c}^{\{\mu\}\times(k-1)} = \frac{\mathbf{c}^{\{\sigma\}\times k}}{k}, \quad k = 1, \dots, q.$$

The commonly-used internal consistency assumption [132, Definition 2.3] is equivalent to $C^{\{\sigma,\mu\}}(1)$ for all σ and μ.

Theorem 6.3. *Suppose the GARK method (6.8) has coefficients satisfying the simplifying assumptions $B^{\{\sigma\}}(p)$ and $C^{\{1,\sigma\}}(q)$ for $\sigma = 1, 2$. Then $W_k(z) \equiv 0$ for $k = 0, \dots, \min(p, q) - 1$.*

Proof. Assume $B^{\{\sigma\}}(p)$ and $C^{\{1,\sigma\}}(q)$ hold for $\sigma = 1, 2$. Since $W_0(z)$ has a different form than the other residuals in (6.16), we will treat is separately. One can easily verify that $W_0(z) \equiv 0$ when $p, q \geq 1$. For $k = 1, \dots, \min(p, q) - 1$,

$$\begin{aligned} W_k(z) &= \left(1 - k\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times(k-1)}\right) + \left(\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times k} - k\mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times(k-1)}\right) z \\ &\quad + \sum_{\ell=2}^{\infty} \mathbf{b}^{\{1\}T}\mathbf{A}^{\{1,1\}\times(\ell-2)} \left(\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times k} - k\mathbf{A}^{\{1,1\}}\mathbf{A}^{\{1,2\}}\mathbf{c}^{\{2\}\times(k-1)}\right) z^{\ell} \\ &= 0. \end{aligned}$$

□

The result in proposition 6.3 is slightly weaker than what can be achieved with unpartitioned Runge–Kutta methods. When we cast a Runge–Kutta method as a GARK method with (6.26), the result in proposition 6.3 can be sharpened by one order, i.e., $W_{\min(p,q)}(z) \equiv 0$. From proposition 6.3, we also see that the minimal conditions of consistency and internal consistency imply $W_0(z) \equiv 0$.

6.3.4 Global Error and Convergence

Following [72, Section IV.15], the accumulation of local truncation errors into the global error e_n is found by unrolling the error recurrence given in (6.13):

$$e_{n+1} = \left(R^{\{1\}}(Z)\right)^{n+1} e_0 + \sum_{j=0}^{n} \left(R^{\{1\}}(Z)\right)^{n-j} \mathrm{lte}_j.$$

Theorem 6.4. *Let $\langle \cdot, \cdot \rangle$ denote an inner product on $\mathbb{C}^d$ and $\|\cdot\|$ denote the induced norm. Assume the linear operator in (6.1) satisfies*

$$\mathrm{Re}\langle y, Ly \rangle \leq \mu \|y\|^2, \qquad \mu \leq 0. \tag{6.23}$$

If the GARK scheme (6.8) has an A-stable base method and satisfies

$$0 = w_{k,\ell}, \text{ for } k = 0, \ldots, p, \text{ and } \ell = 0, \ldots, s^{\{1\}} + 1, \tag{6.24a}$$

$$W_k(z) \text{ bounded for } z \in \mathbb{C}^- \text{ and } k > p, \tag{6.24b}$$

then there exists positive constants h_0 and C such that for $t_f = t_0 + nh$ fixed, the global error is bounded by

$$\|e_n\| \leq C h^p, \qquad \forall\, h \in (0, h_0), \tag{6.25}$$

where C depend on μ, the time span, the method coefficients, and bounds on the derivatives of y.

Remark 6.5. If the eigenvalue of $\mathbf{A}^{\{1,1\}}$ are positive, and $x_{k,-1} = 0$ for $k > p$, then (6.24b) holds.

Remark 6.6. When (6.1) is stiff, the exact solution can have an initial phase of rapid exponential decay. During this time, derivatives of the exact solution, and thus the error constant C in (6.25), can become disproportionally large. This is the case not just for our GARK methods but for B-convergent Runge–Kutta schemes. Outside of the initial transient phase, the derivatives of y can be bounded uniformly in time.

Proof. With the assumptions of the theorem, we can apply Theorem 4 from [73] to get

$$\left\|R^{\{1\}}(Z)\right\| \leq \sup_{\mathrm{Re}(z) \leq \mu} \left|R^{\{1\}}(z)\right| \leq 1 \quad \text{and} \quad \|W_k(Z)\| \leq \sup_{\mathrm{Re}(z) \leq \mu} |W_k(z)| \leq l_k < \infty.$$

Suppose that the derivatives of the exact solution satisfy the bound $\left\|y^{(k)}(t)\right\| \leq m_k$ for $t \in [t_0, t_f]$ and $k > p$. From (6.24a) and proposition 6.2, we have that

$$\|\mathrm{lte}_n\| \leq \sum_{k=p+1}^{\infty} \frac{h^k}{k!} \|W_k(Z)\| \left\|y^{(k)}(t_n)\right\| \leq \frac{h^{p+1}}{1-h} \max_{k \geq p} \frac{l_k m_k}{k!} \leq \frac{h^{p+1}}{1-h_0} \max_{k \geq p} \frac{l_k m_k}{k!},$$

where $0 < h < h_0 < 1$. Now we have that

$$\|e_n\| \leq \sum_{j=0}^{n-1} \left\|R^{\{1\}}(Z)\right\|^{n-1-j} \|\mathrm{lte}_j\| \leq \sum_{j=0}^{n-1} \frac{h^{p+1}}{1-h_0} \max_{k \geq p} \frac{l_k m_k}{k!} \leq \underbrace{\frac{t_f - t_0}{1-h_0} \max_{k \geq p} \frac{l_k m_k}{k!}}_{C} h^p.$$

□

6.3.5 Connections to WSO, Parabolic PDE, and PR Analyses

If we set

$$\mathbf{A}^{\{1,1\}} = \mathbf{A}^{\{1,2\}} = A, \quad \mathbf{b}^{\{1\}} = \mathbf{b}^{\{2\}} = b, \quad \mathbf{c}^{\{1\}} = \mathbf{c}^{\{2\}} = c, \tag{6.26}$$

the GARK method (6.8) degenerates into the traditional Runge–Kutta method (6.2). Assume (6.2) has classical order p. Our (6.16) simplifies to

$$\begin{aligned} W_0(z) &= 0 \\ W_k(z) &= 1 + \left(b^T + zb^T\left(I_{s\times s} - zA\right)^{-1} A\right)\left(zc^k - k\,c^{k-1}\right) \\ &= zb^T\left(I_{s\times s} - zA\right)^{-1}\left(c^k - kAc^{k-1}\right), \qquad k \geq 1. \end{aligned} \tag{6.27}$$

Now, we will show how existing analyses can be viewed as special cases of the GARK analysis in this work.

In the context of weak stage order, the functions

$$g^{(k)} = -\frac{1}{k} zb^T\left(I_{s\times s} - zA\right)^{-1}\left(c^k - kAc^{k-1}\right), \qquad k \geq 1,$$

are defined in [89, eq. 3] and match (6.27) up to an inconsequential scaling. Consider, for example, the following method from [89, page 458] that has order and weak stage order three:

$$\begin{array}{c|ccccc} 0.13756543551 & 0.13756543551 & 0 & 0 & 0 \\ 0.80179011576 & 0.56695122794 & 0.23483888782 & 0 & 0 \\ 2.33179673002 & -1.08354072813 & 2.96618223864 & 0.44915521951 & 0 \\ 0.59761291500 & 0.59761291500 & -0.43420997584 & -0.05305815322 & 0.88965521406 \\ \hline & 0.59761291500 & -0.43420997584 & -0.05305815322 & 0.88965521406 \end{array} . \tag{6.28}$$

One can verify that $g^{(k)} \equiv W_k(z) \equiv 0$ for $k = 1, 2, 3$. In fact, weak stage order $\widetilde{q}$ is equivalent to $g^{(k)} \equiv W_k(z) \equiv 0$ for $k = 1, \ldots, \widetilde{q}$.

Ostermann and Roche use the functions

$$W_k(z) = \frac{b^T(I_{s\times s} - zA)^{-1}(c^k - kAc^{k-1})}{1 - R(z)}, \qquad k \geq 1 \tag{6.29}$$

for the analysis of Runge–Kutta methods applied to linear, parabolic PDEs posed in Hilbert spaces [103]. Again, order reduction can be mitigated by setting $W_k(z) \equiv 0$ for an appropriate set of k. For small z, a series expansion of $W_k(z)$ is shown in [103, page 406] to yield the order conditions

$$b^T A^\ell c^k - kb^T A^{\ell+1} c^{k-1} = 0, \qquad 0 \leq \ell \leq p - k - 1, \text{ and } 1 \leq k \leq p - 1. \tag{6.30}$$

These correspond with our results in (6.10c) and (6.18) under the assumption (6.26). The slightly different scaling of (6.29) compared to (6.27) allows Ostermann and Roche to expand the global error in terms of $h^{k+1}W_k(Z)Zy^{(l)}(t)$ where $l = k, k+1$. This is in contrast to the $h^kW_k(Z)y^{(k)}(t)$ we use. Depending on the spectral properties of Z and the choice of norm, ν can be rational in $h^{k+1}\left\|W_k(Z)Zy^{(l)}(t)\right\| = \mathcal{O}(h^\nu)$. With some care, the approach of Ostermann and Roche can be extended to GARK methods and can explain fractional orders of convergence.

The nonstiff order conditions (6.30) also appear in the global error analysis of Runge–Kutta methods applied to the PR problem [115, page 108]. Further, Rang expands the global error about $z = \infty$ to derive stiff order conditions [115, equations 20 and 21]. These match our $x_{k,\ell}$ coefficients in (6.20) when (6.26) holds.

6.4 Empirical Prothero–Robinson Convergence

In this section, we will examine the error and convergence properties of singly diagonally implicit Runge–Kutta (SDIRK) methods applied to

$$y' = -200(y - \cos(t)) - \sin(t), \quad y(0) = 1, \quad t \in [0, 1]. \tag{6.31}$$

This is a special case of the PR test problem (6.4) with $\lambda = -200$ and $\phi(t) = \cos(t)$.

6.4.1 Order Two

First, we will start with the popular, second order, L-stable SDIRK method

$$\begin{array}{c|cc} 1-\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & 0 \\ 1 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\ \hline & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{array}, \tag{6.32}$$

which we will refer to as SDIRK2. Substituting these coefficients into (6.15) and (6.16) reveals that

$$\mathrm{lte}_n = \frac{(4-3\sqrt{2})Z}{2((\sqrt{2}-2)Z+2)^2}h^2y''(t_n) + \frac{(7-5\sqrt{2})Z - 3\sqrt{2}+4}{6((\sqrt{2}-2)Z+2)^2}h^3y^{(3)}(t_n) + \cdots. \tag{6.33}$$

When $Z = \mathcal{O}(h)$ like a nonstiff ODE, $\mathrm{lte}_n = \mathcal{O}(h^3)$. If we take $Z \to -\infty$ the differential equation becomes an algebraic equation and $\mathrm{lte}_n = 0$. Between these extremes, there are "moderately stiff" problems for which the leading term of (6.33) can cause order reduction.

In order to eliminate this problematic second order error, we extend SDIRK2 to a GARK method (6.8) such that $W_k(z) \equiv 0$ for $k = 0, 1, 2$. This introduces the new coefficients

$\mathbf{A}^{\{1,2\}}$, $\mathbf{b}^{\{2\}}$, and $\mathbf{c}^{\{2\}}$. We make the somewhat arbitrary choice $\mathbf{c}^{\{2\}} = [0, \frac{1}{2}, 1]$. To inherit the stiff accuracy property of the base method, $\mathbf{b}^{\{2\}T} = e_{s\{1\}}\mathbf{A}^{\{1,2\}}$. Using proposition 6.2, the unspecified coefficients in $\mathbf{A}^{\{1,2\}}$ are uniquely determined by the order conditions

$$w_{k,\ell} = 0, \text{ for } k = 0, 1, 2 \text{ and } \ell = 0, 1, 2, 3.$$

Finally, we arrive at the method SDIGARK2

$$\begin{array}{cc|ccc} 1-\frac{1}{\sqrt{2}} & 1 & 0 & \frac{1}{2} & 1 \\ \hline 1-\frac{1}{\sqrt{2}} & 0 & \frac{13}{2}-\frac{9}{\sqrt{2}} & 10\sqrt{2}-14 & \frac{17}{2}-6\sqrt{2} \\ \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & 2\sqrt{2}-\frac{5}{2} & 6-4\sqrt{2} & 2\sqrt{2}-\frac{5}{2} \\ \hline \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & 2\sqrt{2}-\frac{5}{2} & 6-4\sqrt{2} & 2\sqrt{2}-\frac{5}{2} \end{array}, \tag{6.34}$$

which has

$$\text{lte}_n = \frac{(3-2\sqrt{2})Z - 12\sqrt{2} + 16}{6((\sqrt{2}-2)Z+2)^2} h^3 y^{(3)}(t_n) + \cdots .$$

In fig. 6.1, we can see SDIRK2 suffers from order reduction when applied to (6.31), while SDIGARK2 maintains an order of convergence of at least two.

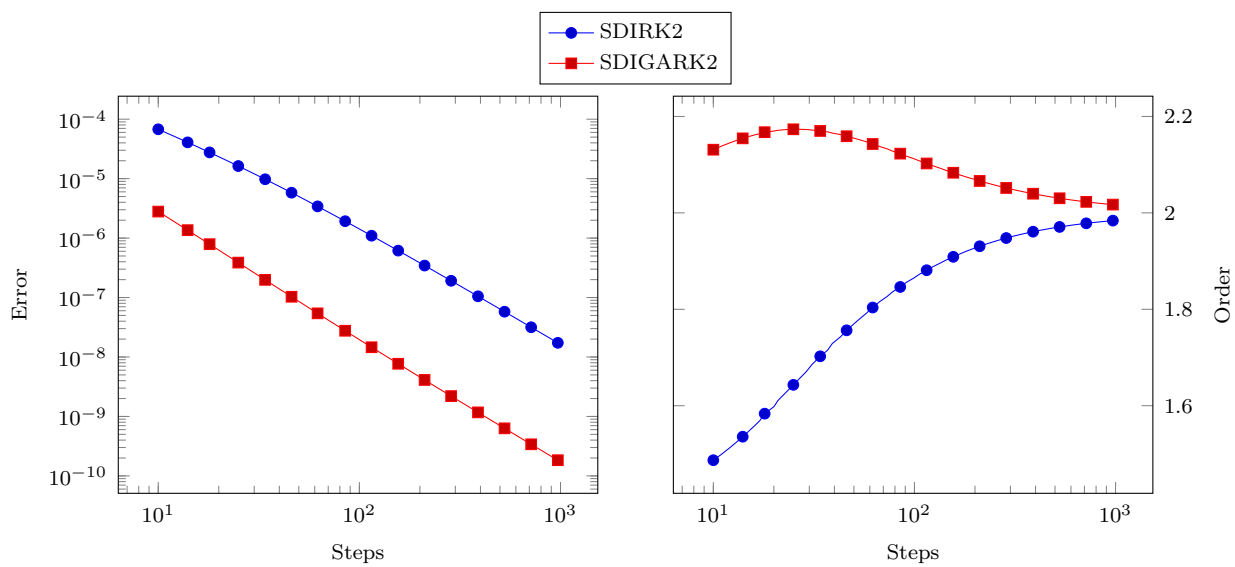


Figure 6.1: Convergence and order for the methods (6.32) and (6.34) when applied to the PR problem (6.31).

6.4.2 Order Three

In contrast to (6.32), the third order method we will consider next is neither L-stable nor stiffly accurate. The method SDIRK3 has the tableau

$$\begin{array}{c|cc} \frac{\sqrt{3}+3}{6} & \frac{\sqrt{3}+3}{6} & 0 \\ \frac{3-\sqrt{3}}{6} & -\frac{1}{\sqrt{3}} & \frac{\sqrt{3}+3}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} . \tag{6.35}$$

With a local truncation error of

$$\text{lte}_n = \frac{(2\sqrt{3}+3)Z^2}{2((\sqrt{3}+3)Z-6)^2} h^2 y''(t_n) + \frac{(3\sqrt{3}+5)Z^2}{6((\sqrt{3}+3)Z-6)^2} h^3 y^{(3)}(t_n) + \cdots ,$$

order reduction is expected outside of the $Z = \mathcal{O}(h)$ regime. We derive a GARK version of (6.35) following a similar methodology to how SDIGARK2 was derived in section 6.4.1 but select $\mathbf{c}^{\{2\}} = \begin{bmatrix} -2 & -1 & 0 & 1 \end{bmatrix}$. For constant stepsizes, this choice only requires one evaluation of g per step because $g(t_n - 2h), \ldots, g(t_n)$ were already computed in previous steps. One can view this as treating the linear term of (6.1) with SDIRK3 and the forcing term with a linear multistep method. Our new method SDIGARK3a, given by

$$\begin{array}{cc|cccc} \frac{\sqrt{3}+3}{6} & \frac{3-\sqrt{3}}{6} & -2 & -1 & 0 & 1 \\ \hline \frac{\sqrt{3}+3}{6} & 0 & \frac{-3\sqrt{3}-5}{36} & \frac{11\sqrt{3}+18}{36} & \frac{-13\sqrt{3}-15}{36} & \frac{11\sqrt{3}+20}{36} \\ -\frac{1}{\sqrt{3}} & \frac{\sqrt{3}+3}{6} & \frac{7\sqrt{3}+13}{36} & \frac{-25\sqrt{3}-48}{36} & \frac{29\sqrt{3}+75}{36} & \frac{-17\sqrt{3}-22}{36} \\ \hline \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}+3}{36} & \frac{-\sqrt{3}-4}{12} & \frac{\sqrt{3}+11}{12} & \frac{12-\sqrt{3}}{36} \end{array} , \tag{6.36}$$

has $W_k(z) \equiv 0$ for $k = 0, \ldots, 3$ so that

$$\text{lte}_n = \frac{\left(2\sqrt{3}+5\right) Z + 2\sqrt{3}+3}{2\left(\left(\sqrt{3}+3\right) Z - 6\right)^2} h^4 y^{(4)}(t_n) + \cdots .$$

Figure 6.2 confirms order reduction for SDIRK3, and interestingly, the convergence line for SDIGARK3a has a cusp around 250 steps. While the error is still consistent with the bounds from proposition 6.4, the instantaneous order of convergence dips below three following the cusp. This occurs because $W_4(z)$ has a root at $z \approx -0.7637$, and around this point, the leading error term no longer dominates the local truncation error. We note SDIGARK2 did not have this behavior because the root of $W_3(z)$ is positive.

One way to avoid the root is by choosing coefficients such that $W_4(z)$ is independent of z. With an additional stage ($s^{\{2\}} = 5$), it is possible to enforce the additional constraint $w_{4,\ell} = 0$ for $\ell \geq 1$, and thus, $W_4(z) \equiv \frac{1}{24} - \frac{1}{6}\mathbf{b}^{\{2\}T}\mathbf{c}^{\{2\}\times 3}$. Our updated method, SDIGARK3b, has a

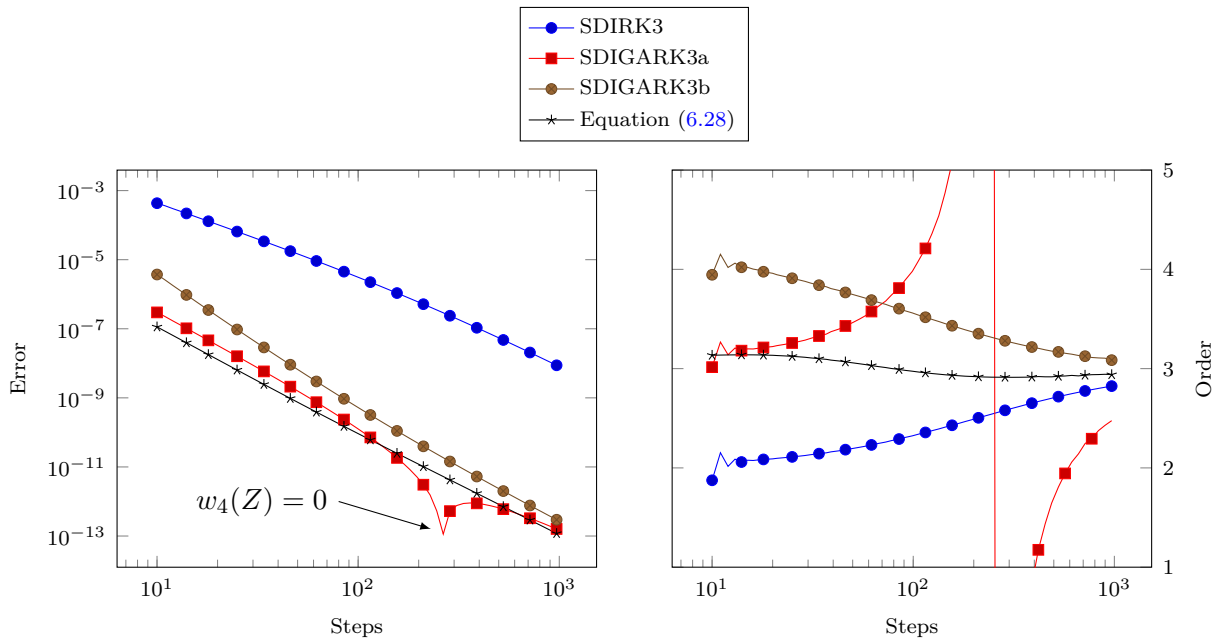


Figure 6.2: Convergence and order for third order DIRK schemes applied to the PR problem (6.31).

constant leading error term and the tableau

$$\begin{array}{cc|ccccc}
\frac{\sqrt{3}+3}{6} & \frac{3-\sqrt{3}}{6} & -3 & -2 & -1 & 0 & 1 \\
\hline
\frac{\sqrt{3}+3}{6} & 0 & \frac{17\sqrt{3}+29}{144} & \frac{-10\sqrt{3}-17}{18} & \frac{73\sqrt{3}+123}{72} & -\frac{11}{9}-\frac{5}{2\sqrt{3}} & \frac{61\sqrt{3}+109}{144} \\
-\frac{1}{\sqrt{3}} & \frac{\sqrt{3}+3}{6} & \frac{-137\sqrt{3}-243}{432} & \frac{79\sqrt{3}+141}{54} & \frac{-187\sqrt{3}-339}{72} & \frac{13}{3}+\frac{56}{9\sqrt{3}} & \frac{-341\sqrt{3}-507}{432} \\
\hline
\frac{1}{2} & \frac{1}{2} & \frac{-5(\sqrt{3}+2)}{72} & \frac{11\sqrt{3}+23}{36} & \frac{-3\sqrt{3}-7}{6} & \frac{13\sqrt{3}+53}{36} & \frac{-7(\sqrt{3}-2)}{72}
\end{array}.$$

It maintains an order of at least three in fig. 6.2. SDIGARK3a and SDIGARK3b have slightly larger errors than the WSO method from (6.28) but solve half as many linear systems and enjoy equal $\mathbf{A}_{i,i}^{\{1,1\}}$.

6.5 Space-Time Convergence on a Hyperbolic PDE

For a second numerical experiment, we will solve the following PDE used in [134]:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -\frac{\partial u}{\partial x} + \frac{t-x}{(1+t)^2}, && x,t \in [0,1], \\
u(t,0) &= \frac{1}{1+t}, && t \in [0,1], \\
u(0,x) &= 1+x, && x \in [0,1].
\end{aligned} \tag{6.37}$$

It possesses the simple, exact solution $u = (1+x)/(1+t)$. We discretize in space with a first order, upwind finite difference scheme on the uniform grid $x_i = ih$, where $i = 0, \ldots, d$ and $h = \frac{1}{d}$. Note that h is used as both the spatial grid size and the timestep in (6.8). The semidiscretized form of (6.37) is

$$y' = \begin{bmatrix} -\frac{1}{h} & & & \\ \frac{1}{h} & -\frac{1}{h} & & \\ & \ddots & \ddots & \\ & & \frac{1}{h} & -\frac{1}{h} \end{bmatrix} y + \begin{bmatrix} \frac{t-x_1}{(1+t)^2} + \frac{1}{h}\frac{1}{1+t} \\ \frac{t-x_2}{(1+t)^2} \\ \vdots \\ \frac{t-x_d}{(1+t)^2} \end{bmatrix} \in \mathbb{R}^d, \tag{6.38}$$

and is of the form (6.1). We will examine the convergence as space and time are simultaneously refined. With the exact solution linear in space, however, the finite differences in space are exact, and we will only measure the temporal error. Error is computed in the ℓ^∞ norm at the final timestep: $\|e_d\|_\infty$.

For the time discretization, we will use the classical fourth order Runge–Kutta method (RK4)

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}, \tag{6.39}$$

which by (6.15), has the local truncation error

$$\begin{aligned} \mathrm{lte}_n = \frac{Z^3}{96} h^2 y''(t_n) + \frac{Z^3 - 2Z^2}{576} h^3 y^{(3)}(t_n) + \frac{Z^3 - 4Z^2 + 8Z}{4608} h^4 y^{(4)}(t_n) \\ + \frac{Z^3 - 6Z^2 + 32Z - 16 I_{d\times d}}{46080} h^5 y^{(5)}(t_n) + \mathcal{O}\big(Z^3 h^6\big). \end{aligned}$$

If $Z = \mathcal{O}(h)$, we recover $\mathrm{lte}_n = \mathcal{O}(h^5)$ as expected. For (6.38), however, $Z = \mathcal{O}(1)$ and the local error is only $\mathcal{O}(h^2)$. Starting with (6.39) as the base method, we can construct a GARK method (6.8) that satisfies

$$w_{k,\ell} = 0, \text{ for } k = 0, \ldots, 4 \text{ and } \ell = 0, \ldots, 5.$$

to avoid order reduction. With $s^{\{2\}} = 5$ and abscissa like that of a linear multistep method,

we uniquely arrive at the following method which we will refer to as GARK4:

$$\begin{array}{c|cccc|ccccc}
0 & \frac{1}{2} & \frac{1}{2} & 1 & & -3 & -2 & -1 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & & 0 & 0 & 0 & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & 0 & 0 & & -\frac{1}{48} & \frac{1}{8} & -\frac{3}{8} & \frac{17}{24} & \frac{1}{16} \\
0 & 0 & 1 & 0 & & -\frac{1}{16} & \frac{1}{3} & -\frac{5}{8} & 1 & \frac{17}{48} \\
\hline
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & & -\frac{5}{144} & \frac{13}{72} & -\frac{5}{12} & \frac{67}{72} & \frac{49}{144}
\end{array} . \tag{6.40}$$

GARK4 has the local truncation error

$$\mathrm{lte}_n = \frac{3Z^3 + 17Z^2 + 41Z + 12I_{d\times d}}{1440} h^5 y^{(5)}(t_n) + \mathcal{O}\big(Z^3 h^6\big),$$

and therefore, should not exhibit order reduction when applied to (6.38). Indeed, this is verified in convergence results presented in fig. 6.3.

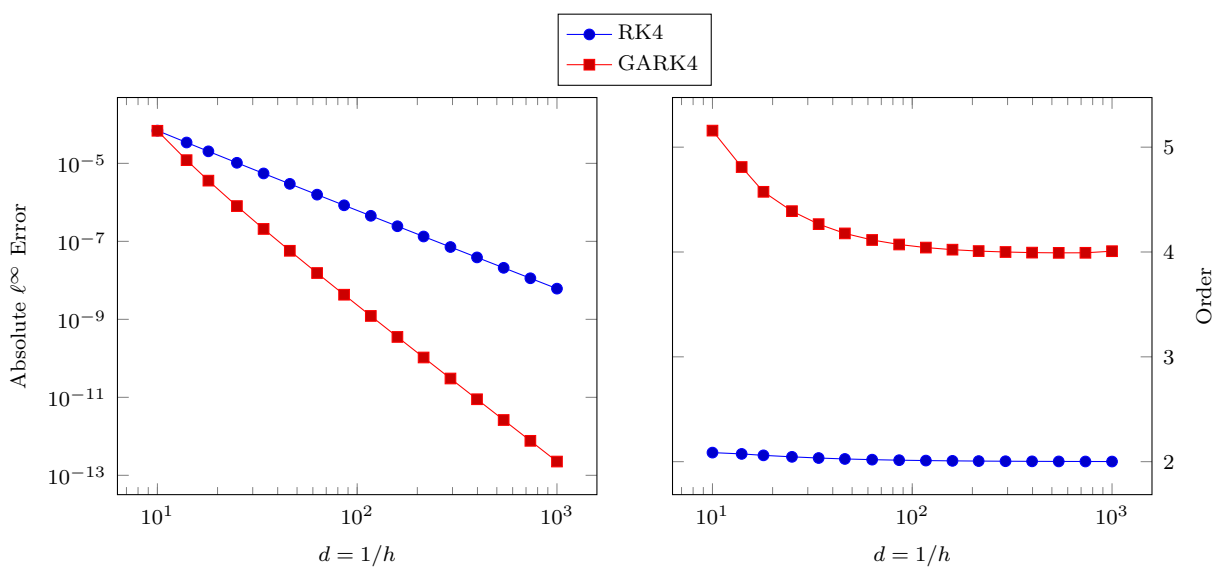


Figure 6.3: Convergence and order for the methods (6.39) and (6.40) when applied to the advection problem (6.38).

6.6 Time-Dependent Heat Equation Experiment

Our final experiment models the transient dynamics of heat in an aluminum heat sink via the PDE

$$\frac{\partial u}{\partial t}(t,\mathbf{x}) = \frac{k}{c_p\rho}\nabla^2 u(t,\mathbf{x}), \qquad \mathbf{x}\in\Omega\subset\mathbb{R}^3, \quad t\in[0,t_f], \tag{6.41a}$$

$$u(t,\mathbf{x}) = T_\infty\left(1+0.1\sin\left(\frac{\pi t}{2t_f}\right)\right), \qquad \mathbf{x}\in\partial\Omega_{\text{bottom}} \tag{6.41b}$$

$$\frac{\partial u}{\partial \mathbf{n}}(t,\mathbf{x}) = \frac{h_c}{k}\left(u(t,\mathbf{x})-T_\infty\right), \qquad \mathbf{x}\in\partial\Omega\setminus\partial\Omega_{\text{bottom}}, \tag{6.41c}$$

$$u(0,\mathbf{x}) = T_\infty, \qquad \mathbf{x}\in\Omega. \tag{6.41d}$$

The domain Ω and snapshots of the solution are plotted in fig. 6.4. The bottom face of the heat sink, Ω_{bottom}, is in contact with a CPU and has a temperature specified by a time-dependent, Dirichlet boundary condition. All other faces are in contact with the air and have convective, Robin boundary conditions. Finally, the model's parameters are listed in table 6.1.

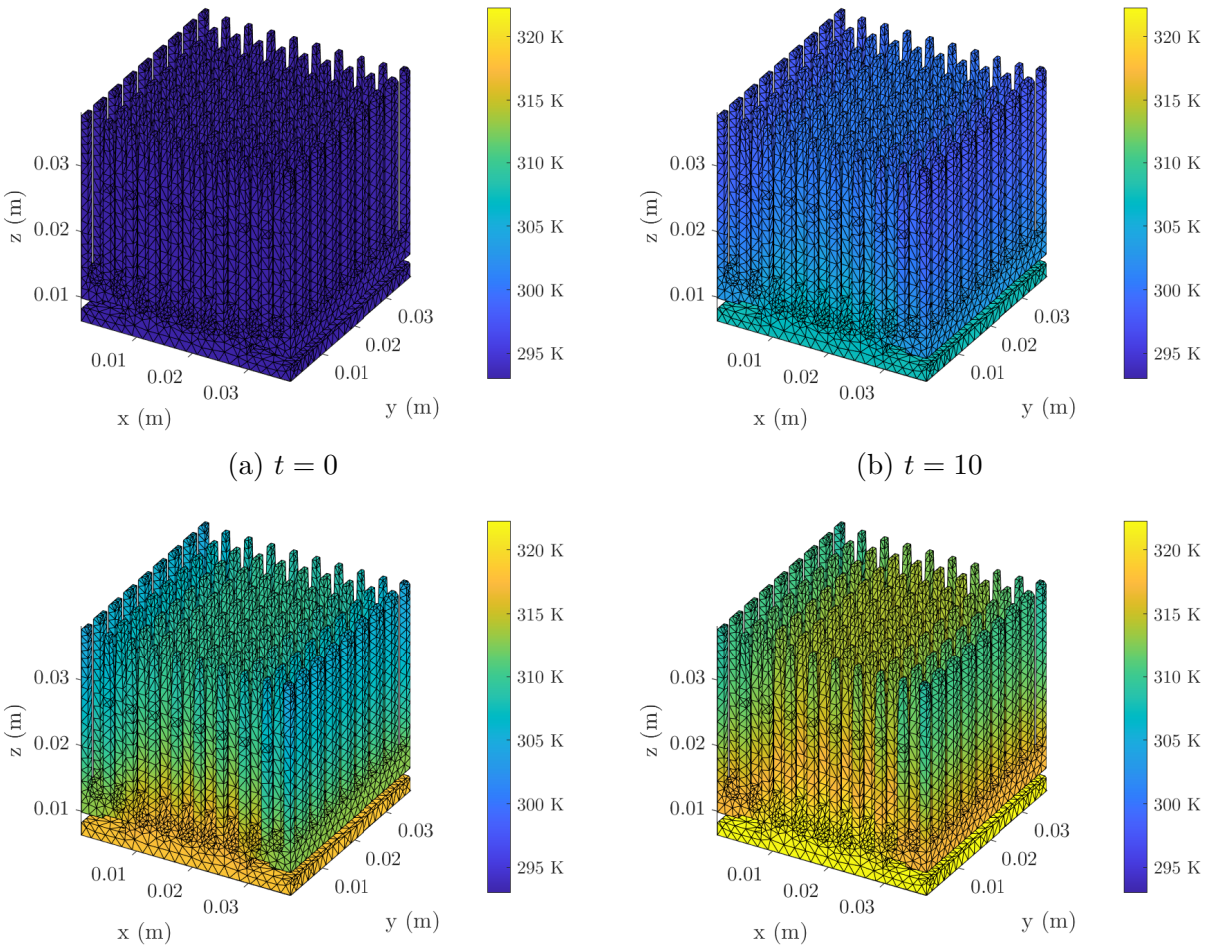
Variable	Description	Value
t_f	end time	$30\,\mathrm{s}$
T_∞	ambient air temperature	$293\,\mathrm{K}$
k	thermal conductivity	$225.94\,\mathrm{W\,m^{-1}\,K^{-1}}$
c_p	specific heat capacity	$900\,\mathrm{J\,K^{-1}\,kg^{-1}}$
ρ	mass density	$2698\,\mathrm{kg\,m^{-3}}$
h_c	convective heat transfer coefficient	$90\,\mathrm{W\,m^{-2}\,K^{-1}}$

Table 6.1: Parameters for heat equation (6.41).

Using MATLAB's PDE Toolbox, a second order, continuous finite element method is applied to the spatial dimensions of (6.41). The meshed heat sink contains 31139 elements and $d = 65570$ degrees of freedom. The resulting ODE is of the form (6.1) but with a mass matrix.

Fully implicit Runge–Kutta methods are some of the best-equipped to solve (6.1), but even these are susceptible to order reduction. For example, s-stage RadauIA methods have classical order $2s-1$ but stiff order $s-1$ for the PR problem [111, Table 1]. Consider the third order RadauIA method

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}, \qquad \mathrm{lte}_n = \frac{1}{6}Z^2(Z^2-4Z+6I_{d\times d})^{-1}h^2y''(t_n)+\cdots. \tag{6.42}$$



(a) $t = 0$ (b) $t = 10$

(c) $t = 20$ (d) $t = 30$

Figure 6.4: Mesh and solution snapshots for the heat equation (6.41).

Following the same strategy used to derive SDIGARK3b in section 6.4.2, we arrive at the following GARK extension to (6.42):

$$\begin{array}{cc|ccccc} 0 & \frac{2}{3} & -3 & -2 & -1 & 0 & 1 \\ \hline \frac{1}{4} & -\frac{1}{4} & -\frac{1}{81} & \frac{11}{162} & -\frac{17}{108} & \frac{53}{162} & -\frac{73}{324} \\ \frac{1}{4} & \frac{5}{12} & -\frac{37}{972} & \frac{95}{486} & -\frac{137}{324} & \frac{389}{486} & \frac{32}{243} \\ \hline \frac{1}{4} & \frac{3}{4} & -\frac{11}{216} & \frac{7}{27} & -\frac{5}{9} & \frac{28}{27} & \frac{67}{216} \end{array}, \qquad \mathrm{lte}_n = \frac{1}{72} h^4 y^{(4)}(t_n) + \cdots . \tag{6.43}$$

The eigenvalues of L for the discretized heat equation are all real and lie in the range $[-121314, -0.33]$. For the range of h used in our convergence experiments, the spectral radius of Z can be as large as 40438. The is problematic for the local error of (6.42), and indeed, order reduction is exhibited in the convergence results of fig. 6.4. With the

GARK Radau IA scheme having a leading error term independent of Z, the global order of convergence is consistently three.

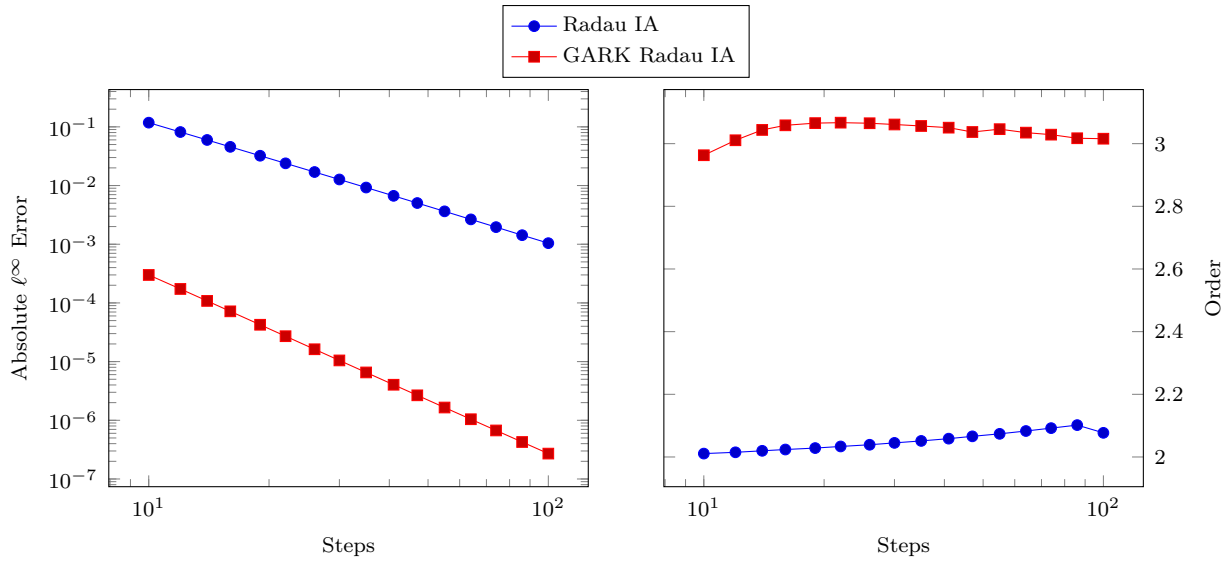


Figure 6.5: Convergence and order for the methods (6.42) and (6.43) when applied to the heat equation (6.41).

6.7 Conclusions

Even on simple, linear ODEs, stiffness can prove problematic for Runge–Kutta methods. In the last several decades, studies into B-convergence and stiff order conditions have addressed issues with order reduction but in ways that are often expensive. As opposed to resorting to additional stages or more coupling among stages, which increases the cost of linear solves, we have presented an inexpensive approach that only introduces additional forcing evaluations. The GARK framework has provided the necessary foundation to couple Runge–Kutta methods, possibly with differing numbers of stages, for the linear and forcing terms. We have presented an error analysis that makes no assumptions on the dependence of Z on h and derived conditions to ensure convergence independent of the stiffness. Finally, our numerical experiments have shown the effectiveness on simple problems like the scalar PR problem as well as more challenging PDEs. There are several possible extensions to this work including nonlinear problems and implicit-explicit (IMEX) methods.

Chapter 7

Design of implicit-explicit generalized additive Runge–Kutta methods for ODEs and DAEs

7.1 Introduction

A key component to many large-scale numerical simulations is solving systems of ordinary differential equations (ODEs) of the form

$$y' = f(y) = f^{\{E\}}(y) + f^{\{I\}}(y), \quad y(t_0) = y_0, \quad t \in [t_0, t_f], \tag{7.1}$$

where $y \in \mathbb{C}^d$. The right-hand side function $f : \mathbb{C}^d \to \mathbb{C}^d$ is additively partitioned into nonstiff dynamics $f^{\{E\}}(y)$ and stiff dynamics $f^{\{I\}}(y)$. An implicit-explicit (IMEX) method seeks to efficiently integrate (7.1) by using an inexpensive explicit method to treat $f^{\{E\}}(y)$ and limiting the application of an expensive implicit method to $f^{\{I\}}(y)$. Applications of IMEX methods range from atmospheric modeling [59] to core-collapse supernova simulations [95] to models of cardiac electrical activity [150].

The IMEX splitting approach has been applied to linear multistep methods [13], general linear methods [164], and linearly implicit methods [71]; however, the focus of this paper will be IMEX Runge–Kutta schemes. Historically, the additive Runge–Kutta (ARK) [44, 45] framework has served as one of the primary foundations for constructing these methods. In the most general form, an ARK method applied to (7.1) has steps of the form

$$Y_i = y_n + h \sum_{j=1}^{s} a_{i,j}^{\{E\}} f^{\{E\}}(Y_j) + h \sum_{j=1}^{s} a_{i,j}^{\{I\}} f^{\{I\}}(Y_j), \qquad i = 1, \ldots, s, \tag{7.2a}$$

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i^{\{E\}} f^{\{E\}}(Y_i) + h \sum_{i=1}^{s} b_i^{\{I\}} f^{\{I\}}(Y_i), \tag{7.2b}$$

where s is the number of stages and h is the timestep. The coefficients $(\mathbf{A}^{\{E\}}, \mathbf{b}^{\{E\}})$ and $(\mathbf{A}^{\{I\}}, \mathbf{b}^{\{I\}})$ define the explicit base method and implicit base method, respectively. IMEX ARK schemes up to order five have been highly-optimized over the last several decades [14, 23, 77, 85, 88]. Except in cases where specialized properties are required, there appears

to be little room for improvement over existing methods in the literature. For example, in a recent paper by Kennedy and Carpenter [88], they refine their fourth and fifth order methods originally proposed in [85], and state "it is unclear how these two methods could be substantially improved."

IMEX Runge–Kutta methods have also been constructed in the additive semi-implicit Runge–Kutta (ASIRK) framework [78, 168]. An ASIRK method is defined by

$$Y_i^{\{E\}} = y_n + h\sum_{j=1}^{s} a_{i,j}^{\{E\}} f^{\{E\}}\left(Y_j^{\{E\}}\right) + h\sum_{j=1}^{s} a_{i,j}^{\{E\}} f^{\{I\}}\left(Y_j^{\{I\}}\right), \quad i = 1,\ldots,s, \tag{7.3a}$$

$$Y_i^{\{I\}} = y_n + h\sum_{j=1}^{s} a_{i,j}^{\{I\}} f^{\{E\}}\left(Y_j^{\{E\}}\right) + h\sum_{j=1}^{s} a_{i,j}^{\{I\}} f^{\{I\}}\left(Y_j^{\{I\}}\right), \quad i = 1,\ldots,s, \tag{7.3b}$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f^{\{E\}}\left(Y_i^{\{E\}}\right) + h\sum_{i=1}^{s} b_i f^{\{I\}}\left(Y_i^{\{I\}}\right). \tag{7.3c}$$

In contrast to (7.2), the stages are partitioned, and each stage treats $f^{\{E\}}$ and $f^{\{I\}}$ with the same coefficients. In [132], this was referred to as "transposed" IMEX since the location of $\mathbf{A}^{\{E\}}$ and $\mathbf{A}^{\{I\}}$ is effectively transposed in the stage equations.

Despite offering complementary structures, both ARK and ASIRK share several limitations. First, the underlying implicit and explicit methods must have the same number of stages s. Otherwise, the method with fewer stages must be awkwardly padded with unused stages. Second, order conditions and common simplifying assumptions such as

$$\mathbf{b}^{\{E\}} = \mathbf{b}^{\{I\}} \quad \text{and} \quad \mathbf{c}^{\{E\}} = \mathbf{A}^{\{E\}}\mathbb{1}_s = \mathbf{c}^{\{I\}} = \mathbf{A}^{\{I\}}\mathbb{1}_s$$

are restrictive and tightly link the base methods together. Past order two, it is rarely possible to leverage existing, optimized Runge–Kutta methods from the literature. An optimal explicit method is unlikely to be compatible with an optimal implicit method. Typically, base methods are derived from scratch by solving a large, coupled system of IMEX order conditions.

Generalized additive Runge–Kutta (GARK) methods were introduced in [132] and encompass ARK, ASIRK, multirate [67], alternating direction implicit [62], and many other classes of methods. A GARK method equipped with an embedded method solves (7.1) via the

computational process

$$Y_i^{\{E\}} = y_n + h\sum_{j=1}^{s^{\{E\}}} a_{i,j}^{\{E,E\}} f^{\{E\}}\left(Y_j^{\{E\}}\right) + h\sum_{j=1}^{s^{\{I\}}} a_{i,j}^{\{E,I\}} f^{\{I\}}\left(Y_j^{\{I\}}\right), \quad i = 1,\ldots,s^{\{E\}}, \tag{7.4a}$$

$$Y_i^{\{I\}} = y_n + h\sum_{j=1}^{s^{\{E\}}} a_{i,j}^{\{I,E\}} f^{\{E\}}\left(Y_j^{\{E\}}\right) + h\sum_{j=1}^{s^{\{I\}}} a_{i,j}^{\{I,I\}} f^{\{I\}}\left(Y_j^{\{I\}}\right), \quad i = 1,\ldots,s^{\{I\}}, \tag{7.4b}$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s^{\{E\}}} b_i^{\{E\}} f^{\{E\}}\left(Y_i^{\{E\}}\right) + h\sum_{i=1}^{s^{\{I\}}} b_i^{\{I\}} f^{\{I\}}\left(Y_i^{\{I\}}\right), \tag{7.4c}$$

$$\widehat{y}_{n+1} = y_n + h\sum_{i=1}^{s^{\{E\}}} \widehat{b}_i^{\{E\}} f^{\{E\}}\left(Y_i^{\{E\}}\right) + h\sum_{i=1}^{s^{\{I\}}} \widehat{b}_i^{\{I\}} f^{\{I\}}\left(Y_i^{\{I\}}\right), \tag{7.4d}$$

which is represented by the tableau

$$\begin{array}{c|c} \mathbf{A}^{\{E,E\}} & \mathbf{A}^{\{E,I\}} \\ \hline \mathbf{A}^{\{I,E\}} & \mathbf{A}^{\{I,I\}} \\ \hline \mathbf{b}^{\{E\}T} & \mathbf{b}^{\{I\}T} \\ \hline \widehat{\mathbf{b}}^{\{E\}T} & \widehat{\mathbf{b}}^{\{I\}T} \end{array}. \tag{7.5}$$

The unifying GARK framework provides a more natural representation of IMEX Runge–Kutta methods as it reveals the "hidden" coupling coefficients $\mathbf{A}^{\{E,I\}}$ and $\mathbf{A}^{\{I,E\}}$. This roughly doubles the number of coefficients defining the method compared to ARK and ASIRK. Further, $\mathbf{A}^{\{E,I\}}$ and $\mathbf{A}^{\{I,E\}}$ need not be square, and the base methods can have a different number of stages, i.e., $s^{\{E\}} \neq s^{\{I\}}$. This additional flexibility allows us to improve upon existing IMEX methods and address the aforementioned limitations.

The goals of this paper are to explore the space of IMEX GARK methods, determine practical method structures, define important method properties, and present design criteria for the derivation of high-quality IMEX GARK methods. We present new methods up to order four that are suitable for ODEs and some that are suitable for index-1 differential algebraic equations (DAEs) as well. Our GARK-based analysis also sheds light on how to design high-order ARK methods for DAEs.

This paper builds upon several prior works on GARK methods. IMEX methods based on the linearly implicit GARK framework were proposed by the authors in [133]. In [153], Tanner presents a detailed investigation into GARK methods applied to system with two stiff partitions. This work also lays the foundation for GARK DAE order condition theory which we use later in this paper. Many design criteria for our IMEX methods are similar to that

of multirate GARK methods [123, 135]. Also related are the IMEX multirate infinitesimal GARK methods from [39].

The remaining sections are organized as follows. Two coefficient structures for IMEX GARK methods are discussed in section 7.2. Section 7.3 investigates the linear stability for these structures. We review classical order conditions for GARK methods in section 7.4 then move to DAE order conditions in section 7.5. The derivation of new IMEX GARK methods is detailed in section 7.6. Section 7.7 includes two numerical experiments that compare GARK IMEX methods to other IMEX schemes. Concluding remarks are found in section 7.8.

7.2 Practical IMEX GARK Structures

Most commonly, the implicit base method of an IMEX Runge–Kutta scheme is a singly diagonally implicit Runge–Kutta (SDIRK) method or an explicit first stage SDIRK (ESDIRK) method. These have the tableaus

$$\begin{array}{c|cccc} c_1^{\{I\}} & \gamma & & & \\ c_2^{\{I\}} & a_{2,1}^{\{I,I\}} & \gamma & & \\ \vdots & \vdots & & \ddots & \\ c_{s^{\{I\}}}^{\{I\}} & a_{s^{\{I\}},1}^{\{I,I\}} & a_{s^{\{I\}},2}^{\{I,I\}} & \cdots & \gamma \\ \hline & b_1^{\{I\}} & b_2^{\{I\}} & \cdots & b_{s^{\{I\}}}^{\{I\}} \end{array} \quad \text{and} \quad \begin{array}{c|cccc} 0 & 0 & & & \\ c_2^{\{I\}} & a_{2,1}^{\{I,I\}} & \gamma & & \\ \vdots & \vdots & & \ddots & \\ c_{s^{\{I\}}}^{\{I\}} & a_{s^{\{I\}},1}^{\{I,I\}} & a_{s^{\{I\}},2}^{\{I,I\}} & \cdots & \gamma \\ \hline & b_1^{\{I\}} & b_2^{\{I\}} & \cdots & b_{s^{\{I\}}}^{\{I\}} \end{array}, \tag{7.6}$$

respectively. These keep the size of nonlinear systems of equations small when solving for implicit stages and permits the reuse of Jacobian decompositions across stages. ESDIRK methods are motivated by the potential to have stage order two.

When the implicit method is paired with an explicit one, the GARK coupling matrices $\mathbf{A}^{\{E,I\}}$ and $\mathbf{A}^{\{I,E\}}$ must ensure that implicitness resides solely in the implicit method, and the $Y_i^{\{E\}}$ stages are explicit. That is, the GARK method must be decoupled [135, Definition 5.1]. It is necessary for $\mathbf{A}^{\{E,I\}}$ and $\mathbf{A}^{\{I,E\}}$ to be block lower triangular and satisfy the complementarity condition

$$\mathbf{A}^{\{E,I\}} \times \mathbf{A}^{\{I,E\}T} = 0_{s^{\{E\}} \times s^{\{I\}}}, \tag{7.7}$$

where $\times$ denotes an element-wise product.

Another condition that dictates the sparsity of the coupling is internal consistency [132, Definition 2.3]. This is given by

$$\mathbf{c}^{\{E\}} := \mathbf{c}^{\{E,E\}} = \mathbf{c}^{\{E,I\}} \quad \text{and} \quad \mathbf{c}^{\{I\}} := \mathbf{c}^{\{I,E\}} = \mathbf{c}^{\{I,I\}}, \tag{7.8}$$

where $\mathbf{c}^{\{\nu,\mu\}} = \mathbf{A}^{\{\nu,\mu\}} \mathbb{1}_{s^{\{\mu\}}}$. While not absolutely necessary, it is generally worth enforcing because it simplifies order conditions and provides many favorable properties.

The first class of methods we will consider, which we refer to as $\mathbb{M}1$ methods, use an SDIRK implicit method:

$$
\begin{array}{cccc|cccc}
0 & & & & 0 & & & \\
a_{2,1}^{\{E,E\}} & 0 & & & a_{2,1}^{\{E,I\}} & 0 & & \\
\vdots & & \ddots & & \vdots & & \ddots & \\
a_{s^{\{E\}},1}^{\{E,E\}} & a_{s^{\{E\}},2}^{\{E,E\}} & \cdots & 0 & a_{s^{\{E\}},1}^{\{E,I\}} & a_{s^{\{E\}},2}^{\{E,I\}} & \cdots & 0 \\
\hline
a_{1,1}^{\{I,E\}} & & & & \gamma & & & \\
a_{2,1}^{\{I,E\}} & a_{2,2}^{\{I,E\}} & & & a_{2,1}^{\{I,I\}} & \gamma & & \\
\vdots & & \ddots & & \vdots & & \ddots & \\
a_{s^{\{I\}},1}^{\{I,E\}} & a_{s^{\{I\}},2}^{\{I,E\}} & \cdots & a_{s^{\{I\}},s^{\{I\}}}^{\{I,E\}} & a_{s^{\{I\}},1}^{\{I,I\}} & a_{s^{\{I\}},2}^{\{I,I\}} & \cdots & \gamma \\
\hline
b_1^{\{E\}} & b_2^{\{E\}} & \cdots & b_{s^{\{E\}}}^{\{E\}} & b_1^{\{I\}} & b_2^{\{I\}} & \cdots & b_{s^{\{I\}}}^{\{I\}} \\
\hline
\widehat{b}_1^{\{E\}} & \widehat{b}_2^{\{E\}} & \cdots & \widehat{b}_{s^{\{E\}}}^{\{E\}} & \widehat{b}_1^{\{I\}} & \widehat{b}_2^{\{I\}} & \cdots & \widehat{b}_{s^{\{I\}}}^{\{I\}}
\end{array}
\quad . \tag{7.9}
$$

Internal consistency and (7.7) force $\mathbf{A}^{\{E,I\}}$ to be block strictly lower triangular and have a first row of zeros. Likewise, $\mathbf{A}^{\{I,E\}}$ must be block lower triangular. ARK methods that use an SDIRK implicit method must be padded with zero coefficients for internal consistency to hold. This is not necessary in (7.9) as the coupling coefficients act as a buffer between the base methods with potentially different abscissae.

The $\mathbb{M}2$ class of IMEX GARK methods is based on an ESDIRK method:

$$
\begin{array}{ccccc|ccccc}
0 & & & & & 0 & & & & \\
a_{2,1}^{\{E,E\}} & 0 & & & & a_{2,1}^{\{E,I\}} & a_{2,2}^{\{E,I\}} & & & \\
\vdots & & \ddots & & & \vdots & & \ddots & & \\
a_{s^{\{E\}}-1,1}^{\{E,E\}} & a_{s^{\{E\}}-1,2}^{\{E,E\}} & \cdots & 0 & & a_{s^{\{E\}}-1,1}^{\{E,I\}} & a_{s^{\{E\}}-1,2}^{\{E,I\}} & \cdots & a_{s^{\{E\}}-1,s^{\{E\}}-1}^{\{E,I\}} & \\
b_1^{\{E\}} & b_2^{\{E\}} & \cdots & b_{s^{\{E\}}-1}^{\{E\}} & 0 & b_1^{\{I\}} & b_2^{\{I\}} & \cdots & b_{s^{\{I\}}-1}^{\{I\}} & \gamma \\
\hline
0 & & & & & 0 & & & & \\
a_{2,1}^{\{I,E\}} & 0 & & & & a_{2,1}^{\{I,I\}} & \gamma & & & \\
\vdots & & \ddots & & & \vdots & & \ddots & & \\
a_{s^{\{I\}}-1,1}^{\{I,E\}} & a_{s^{\{I\}}-1,2}^{\{I,E\}} & \cdots & 0 & & a_{s^{\{I\}}-1,1}^{\{I,I\}} & a_{s^{\{I\}}-1,2}^{\{I,I\}} & \cdots & \gamma & \\
b_1^{\{E\}} & b_2^{\{E\}} & \cdots & b_{s^{\{E\}}-1}^{\{E\}} & 0 & b_1^{\{I\}} & b_2^{\{I\}} & \cdots & b_{s^{\{I\}}-1}^{\{I\}} & \gamma \\
\hline
b_1^{\{E\}} & b_2^{\{E\}} & \cdots & b_{s^{\{E\}}-1}^{\{E\}} & 0 & b_1^{\{I\}} & b_2^{\{I\}} & \cdots & b_{s^{\{I\}}-1}^{\{I\}} & \gamma \\
\hline
\widehat{b}_1^{\{E\}} & \widehat{b}_2^{\{E\}} & \cdots & \widehat{b}_{s^{\{E\}}-1}^{\{E\}} & \widehat{b}_{s^{\{E\}}}^{\{E\}} & \widehat{b}_1^{\{I\}} & \widehat{b}_2^{\{I\}} & \cdots & \widehat{b}_{s^{\{I\}}-1}^{\{I\}} & \widehat{b}_{s^{\{I\}}}^{\{I\}}
\end{array}
\quad . \tag{7.10}
$$

By construction, it is stiffly accurate [132, Definition 3.3] in both partitions since

$$\mathbf{b}^{\{E\}T} = e_{s^{\{E\}}}^T \mathbf{A}^{\{E,E\}}, \qquad \mathbf{b}^{\{I\}T} = e_{s^{\{E\}}}^T \mathbf{A}^{\{E,I\}}, \tag{7.11a}$$

$$\mathbf{b}^{\{E\}T} = e_{s^{\{I\}}}^T \mathbf{A}^{\{I,E\}}, \qquad \mathbf{b}^{\{I\}T} = e_{s^{\{I\}}}^T \mathbf{A}^{\{I,I\}}. \tag{7.11b}$$

Moreover, (7.10) has the first same as last (FSAL) property. This saves the two function evaluations $f^{\{E\}}(Y_1^{\{E\}})$ and $f^{\{I\}}(Y_1^{\{I\}})$ every step because these values are equivalent to $f^{\{E\}}(Y_{s^{\{E\}}}^{\{E\}})$ and $f^{\{I\}}(Y_{s^{\{I\}}}^{\{I\}})$ from the previous step. In contrast to (7.9), $\mathbf{A}^{\{E,I\}}$ is block lower triangular, and $\mathbf{A}^{\{I,E\}}$ is block strictly lower triangular. While the opposite structure is possible, it is incompatible with certain stability and DAE consistency conditions that will be discussed later. For the upcoming analysis of $\mathbb{M}2$ methods, the singular $\mathbf{A}^{\{I,I\}}$ will introduce special cases, and it will be helpful to define the notation

$$\mathbf{A}^{\{I,I\}} = \begin{bmatrix} 0 & 0_{s^{\{I\}}-1}^T \\ \mathbf{A}_1^{\{I,I\}} & \widehat{\mathbf{A}}^{\{I,I\}} \end{bmatrix}, \quad \mathbf{A}^{\{I,E\}} = \begin{bmatrix} 0_{s^{\{E\}}}^T \\ \widehat{\mathbf{A}}^{\{I,E\}} \end{bmatrix}, \quad \mathbf{A}^{\{E,I\}} = \begin{bmatrix} \mathbf{A}_1^{\{E,I\}} & \widehat{\mathbf{A}}^{\{E,I\}} \end{bmatrix},$$

where $\mathbf{A}_1^{\{E,I\}} \in \mathbb{R}^{s^{\{E\}}\times 1}$ and all other dimensions are evident.

7.3 Linear Stability

GARK methods applied to the linear test problem

$$y' = \lambda^{\{E\}} y + \lambda^{\{I\}} y, \tag{7.12}$$

where $\lambda^{\{E\}}, \lambda^{\{I\}} \in \mathbb{C}^-$, have been well-studied [123, 132]. In this section, we will specialize the analysis for the $\mathbb{M}1$ and $\mathbb{M}2$ classes of IMEX GARK methods. In both cases, the internal stability function is given by

$$\begin{aligned} R_{\text{int}}\left(z^{\{E\}}, z^{\{I\}}\right) &= \begin{bmatrix} R_{\text{int}}^{\{E\}}\left(z^{\{E\}}, z^{\{I\}}\right) \\ R_{\text{int}}^{\{I\}}\left(z^{\{E\}}, z^{\{I\}}\right) \end{bmatrix} \\ &= \begin{bmatrix} I - z^{\{E\}}\mathbf{A}^{\{E,E\}} & -z^{\{I\}}\mathbf{A}^{\{E,I\}} \\ -z^{\{E\}}\mathbf{A}^{\{I,E\}} & I - z^{\{I\}}\mathbf{A}^{\{I,I\}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{1}_{s^{\{E\}}} \\ \mathbb{1}_{s^{\{I\}}} \end{bmatrix}, \end{aligned} \tag{7.13}$$

where $z^{\{E\}} = h\lambda^{\{E\}}$ and $z^{\{I\}} = h\lambda^{\{I\}}$. For the test problem (7.12), this represents the amplification of errors for each of the $s^{\{E\}} + s^{\{I\}}$ GARK stages. Internal stability was found to be an important design property of IMEX ARK methods because it controls the "stiffness leakage" phenomenon [85]. Of particular interest is the case where $z^{\{I\}} \to -\infty$. For $\mathbb{M}1$ methods,

$$\begin{aligned} R_{\text{int}}\left(z^{\{E\}}, -\infty\right) &= \begin{bmatrix} \left(I + z^{\{E\}}\left(\mathbf{A}^{\{E,I\}}\mathbf{A}^{\{I,I\}\times -1}\mathbf{A}^{\{I,E\}} - \mathbf{A}^{\{E,E\}}\right)\right)^{-1} r \\ 0_{s^{\{I\}}} \end{bmatrix}, \\ r &= \mathbb{1}_{s^{\{E\}}} - \mathbf{A}^{\{E,I\}}\mathbf{A}^{\{I,I\}\times -1}\mathbb{1}_{s^{\{I\}}}. \end{aligned} \tag{7.14}$$

We can see that the implicit stages have perfect damping, but the damping of the explicit stages is determined by polynomials in $z^{\{E\}}$. Ideally, these internal stability polynomials will be less than one in magnitude for a large region of $z^{\{E\}}$ in the left half-plane. Equation (7.14) suggests

$$\mathbb{1}_{s^{\{E\}}} = \mathbf{A}^{\{E,I\}}\mathbf{A}^{\{I,I\}\times-1}\mathbb{1}_{s^{\{I\}}}, \tag{7.15a}$$

$$\mathbf{A}^{\{E,E\}} = \mathbf{A}^{\{E,I\}}\mathbf{A}^{\{I,I\}\times-1}\mathbf{A}^{\{I,E\}}, \tag{7.15b}$$

as possible simplifying assumptions: both of which also appear in [153, Section 2.5]. However, the former is impossible to satisfy for $\mathbb{M}1$ methods because the first element of r is always 1.

For $\mathbb{M}2$ methods, the additional explicit stage causes the degree of $z^{\{I\}}$ in the denominator of (7.13) to be one lower. Thus, the internal stability for the explicit stages can grow unbounded in $z^{\{I\}}$ unless an additional condition is enforced. To isolate the problematic, highest-degree term in the numerator, we examine the limit

$$\begin{aligned}\lim_{z^{\{I\}}\to-\infty}\frac{R_{\text{int}}^{\{E\}}\left(z^{\{E\}},z^{\{I\}}\right)}{z^{\{I\}}} &= \left(I + z^{\{E\}}\left(\widehat{\mathbf{A}}^{\{E,I\}}\widehat{\mathbf{A}}^{\{I,I\}\times-1}\widehat{\mathbf{A}}^{\{I,E\}} - \mathbf{A}^{\{E,E\}}\right)\right)^{-1}\widehat{r},\\ \widehat{r} &= \mathbf{A}_1^{\{E,I\}} - \widehat{\mathbf{A}}^{\{E,I\}}\widehat{\mathbf{A}}^{\{I,I\}\times-1}\mathbf{A}_1^{\{I,I\}}.\end{aligned} \tag{7.16}$$

We can see simplifying assumptions

$$\mathbf{A}_1^{\{E,I\}} = \widehat{\mathbf{A}}^{\{E,I\}}\widehat{\mathbf{A}}^{\{I,I\}\times-1}\mathbf{A}_1^{\{I,I\}}, \tag{7.17a}$$

$$\mathbf{A}^{\{E,E\}} = \widehat{\mathbf{A}}^{\{E,I\}}\widehat{\mathbf{A}}^{\{I,I\}\times-1}\widehat{\mathbf{A}}^{\{I,E\}}, \tag{7.17b}$$

arise analogous to (7.15). Fortunately, it is possible for $\mathbb{M}2$ methods to satisfy (7.17a) since $\mathbf{A}^{\{E,I\}}$ is block lower triangular, but not strictly so. Equation (7.17a) is both necessary and sufficient for the internal stability of an $\mathbb{M}2$ method to be bounded. For the implicit stages of an $\mathbb{M}2$ method,

$$R_{\text{int}}^{\{I\}}\left(z^{\{E\}},-\infty\right) = \begin{bmatrix} 1 \\ -\left(\left(\widehat{\mathbf{A}}^{\{I,I\}} + z^{\{E\}}\widehat{\mathbf{A}}^{\{I,E\}}\left(I - z^{\{E\}}\mathbf{A}^{\{E,E\}}\right)^{-1}\widehat{\mathbf{A}}^{\{E,I\}}\right)^{-1} \\ \cdot\left(\mathbf{A}_1^{\{I,I\}} + z^{\{E\}}\widehat{\mathbf{A}}^{\{I,E\}}\left(I - z^{\{E\}}\mathbf{A}^{\{E,E\}}\right)^{-1}\mathbf{A}_1^{\{E,I\}}\right)\right)\end{bmatrix}. \tag{7.18}$$

Using the internal stability function, we can express the linear stability of the GARK method (7.4) as

$$R\left(z^{\{E\}},z^{\{I\}}\right) = 1 + z^{\{E\}}\mathbf{b}^{\{E\}T}R_{\text{int}}^{\{E\}}\left(z^{\{E\}},z^{\{I\}}\right) + z^{\{I\}}\mathbf{b}^{\{I\}T}R_{\text{int}}^{\{I\}}\left(z^{\{E\}},z^{\{I\}}\right).$$

Visualizing the region of stability for this multivariate function is commonly simplified by considering the set

$$\mathcal{S}_{\infty,\alpha}^{\text{1D}} = \left\{z^{\{E\}}\in\mathbb{C}\,\middle|\,\left|R\left(z^{\{E\}},z^{\{I\}}\right)\right|\le 1,\ \forall z^{\{I\}}\in\mathbb{C}^-:\left|\arg\left(z^{\{I\}}\right)-\pi\right|\le\alpha\right\}. \tag{7.19}$$

In a sense, this quantifies the explicit stability of an IMEX method when $\lambda^{\{I\}}$ in (7.12) is chosen in a worst-case manner. Note that $\mathcal{S}^{\mathrm{lp}}_{\infty,\alpha}$ is always a subset of the stability region of the explicit base method and $\mathcal{S}^{\mathrm{lp}}_{\infty,\alpha_1} \subseteq \mathcal{S}^{\mathrm{lp}}_{\infty,\alpha_2}$ for $0 \le \alpha_1 \le \alpha_2 \le \pi$.

Again, we can consider the limit $z^{\{I\}} \to -\infty$. For $\mathbb{M}1$ methods, stiff accuracy in the implicit partition, i.e. (7.11b), implies $R\big(z^{\{E\}}, -\infty\big) = 0$. $\mathbb{M}2$ methods are stiffly accurate by design but require more to achieve perfect damping. One such way is given by the following theorem.

Theorem 7.1. *If for an $\mathbb{M}2$ method there exists $v \in \mathbb{R}^{s^{\{I\}}-1}$ such that*

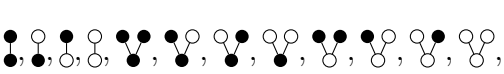
$$\mathbf{A}_1^{\{I,I\}} = \widehat{\mathbf{A}}^{\{I,I\}} v \quad \textit{and} \quad \mathbf{A}_1^{\{E,I\}} = \widehat{\mathbf{A}}^{\{E,I\}} v, \tag{7.20}$$

then $R\big(z^{\{E\}}, -\infty\big) = -e^T_{s^{\{I\}}-1} v$.

Proof. This follows by substituting (7.20) into (7.18) and examining the last component since the method is stiffly accurate. □

7.4 Classical Order Conditions

Like ARK and ASIRK methods, the order conditions and error analysis of GARK methods can be studied via N-trees [12]. For IMEX methods there are $N = 2$ partitions, and we use the set of 2-trees:

$2T = \{\emptyset,$  $\dots\},$

$2T_p = \{\mathfrak{t} \in 2T : \rho(\mathfrak{t}) = p\}.$ (7.21)

Black vertices denote the explicit partition, and white vertices denote the implicit partition. In (7.21), $\rho(\mathfrak{t})$ gives the order (number of vertices) of $\mathfrak{t}$. If $\rho(\mathfrak{t}) > 1$, then $\mathfrak{t}$ can be decomposed in the form $[\mathfrak{t}_1, \mathfrak{t}_2, \dots, \mathfrak{t}_m]^{\{\nu\}}$ where $\nu \in \{E, I\}$ is the color of the root and each $\mathfrak{t}_j$ is a nonempty subtree attached directly to the root. For example

 $= \Big[$  $\Big]^{\{E\}} = \Big[[\bullet]^{\{I\}}, [\bullet]^{\{I\}}, \circ\Big]^{\{E\}}.$ (7.22)

Also associated with $\mathfrak{t}$ is the density $\gamma(\mathfrak{t})$ and the number of symmetries $\sigma(\mathfrak{t})$. These are defined recursively as

$$\begin{aligned}
\gamma(\mathfrak{t}) &= \sigma(\mathfrak{t}) = 1, && \text{for } \rho(\mathfrak{t}) \le 1,\\
\gamma(\mathfrak{t}) &= \rho(\mathfrak{t})\gamma(\mathfrak{t}_1)\cdots\gamma(\mathfrak{t}_m), && \text{for } \mathfrak{t} = [\mathfrak{t}_1, \dots, \mathfrak{t}_m]^{\{\nu\}},\\
\sigma(\mathfrak{t}) &= (\mu_1!\mu_2!\cdots)\,\sigma(\mathfrak{t}_1)\cdots\sigma(\mathfrak{t}_m), && \text{for } \mathfrak{t} = [\mathfrak{t}_1, \dots, \mathfrak{t}_m]^{\{\nu\}},
\end{aligned}$$

where $\mu_1, \mu_2, \dots$ are the multiplicities of the subtrees $\mathfrak{t}_1, \dots, \mathfrak{t}_m$. The tree in (7.22) has $\mu_1 = 2$ and $\mu_2 = 1$, for example.

Theorem 7.2 (Order conditions [132, Section 2.4]). *The IMEX GARK method* (7.4) *has order p iff*

$$e(\mathfrak{t}) = \frac{1}{\sigma(\mathfrak{t})}\left(\Phi(\mathfrak{t}) - \frac{1}{\gamma(\mathfrak{t})}\right) = 0, \quad \forall \mathfrak{t} \in 2T, \quad 1 \le \rho(\mathfrak{t}) \le p, \tag{7.23}$$

where the elementary weights are

$$\Phi(\mathfrak{t}) = \begin{cases} \mathbf{b}^{\{E\}T}\mathbb{1}_{s^{\{E\}}}, & \mathfrak{t} = \bullet, \\ \mathbf{b}^{\{I\}T}\mathbb{1}_{s^{\{I\}}}, & \mathfrak{t} = \circ, \\ \mathbf{b}^{\{\nu\}T}\left(\Phi^{\{\nu\}}(\mathfrak{t}_1) \times \cdots \times \Phi^{\{\nu\}}(\mathfrak{t}_m)\right), & \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m]^{\{\nu\}}, \end{cases}$$

$$\Phi^{\{\nu\}}(\mathfrak{t}) = \begin{cases} \mathbf{c}^{\{\nu,E\}}, & \mathfrak{t} = \bullet, \\ \mathbf{c}^{\{\nu,I\}}, & \mathfrak{t} = \circ, \\ \mathbf{A}^{\{\nu,\mu\}}\left(\Phi^{\{\mu\}}(\mathfrak{t}_1) \times \cdots \times \Phi^{\{\mu\}}(\mathfrak{t}_m)\right), & \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m]^{\{\mu\}}. \end{cases}$$

IMEX GARK order conditions up to order four can be found in [131, Appendix C]. Note that these assume internal consistency while proposition 7.2 does not.

For the embedded method (7.4d), we can define $\widehat{e}(\mathfrak{t})$ similar to (7.23) but with the $\widehat{\mathbf{b}}^{\{E\}}$ and $\widehat{\mathbf{b}}^{\{I\}}$ coefficients. In this work, all embedded methods have order $\widehat{p} = p - 1$. To compare the accuracy of methods of the same order, we use the principal errors

$$A^{(p+1)} = \sqrt{\sum_{\mathfrak{t} \in 2T_{p+1}} e(\mathfrak{t})^2} \quad \text{and} \quad \widehat{A}^{(\widehat{p}+1)} = \sqrt{\sum_{\mathfrak{t} \in 2T_{\widehat{p}+1}} \widehat{e}(\mathfrak{t})^2}. \tag{7.24}$$

In [85, 88], the principal error is defined slightly differently so as to exclude trees whose order conditions are redundant due to simplifying assumptions. Instead, we consider all trees equally. The principal errors for the explicit and implicit base methods are denoted with $A^{\{E\}(p+1)}$ and $A^{\{I\}(p+1)}$, respectively.

The quality of an embedded method can be assessed using the quantities

$$B^{(\widehat{p}+2)} = \frac{\widehat{A}^{(p+2)}}{\widehat{A}^{(p+1)}}, \quad C^{(\widehat{p}+2)} = \frac{\sqrt{\sum_{\mathfrak{t} \in 2T_{\widehat{p}+2}} \left(\widehat{e}(\mathfrak{t}) - e(\mathfrak{t})\right)^2}}{\widehat{A}^{(\widehat{p}+1)}}, \quad E^{(\widehat{p}+2)} = \frac{A^{(\widehat{p}+2)}}{\widehat{A}^{(\widehat{p}+1)}}, \tag{7.25}$$

which extend those in [20, 85]. Ideally each quantity in (7.25) will be close to 1 so that the embedded method provides an accurate approximation of the local truncation error over a wide range of h.

Finally, the quantity

$$D = \max\left\{\left|a_{i,j}^{\{\nu,\mu\}}\right|, \left|b_i^{\{\nu\}}\right|, \left|\widehat{b}_i^{\{\nu\}}\right|, \left|c_i^{\{\nu\}}\right|\right\}$$

gives the largest element in the tableau (7.5) in magnitude. While there are differing suggestions on bounds for D, it should be kept small to reduce the effect of cancellation errors and floating-point inaccuracies.

7.4.1 Simplifying Assumptions

Simplifying assumptions commonly used for traditional Runge–Kutta methods have natural extensions to the GARK framework. From [153, Section 2.3],

$$
\begin{aligned}
B^{\{\nu\}}(p): &\quad \mathbf{b}^{\{\nu\}T}\mathbf{c}^{\{\nu\}\times(k-1)} = \frac{1}{k}, && k = 1,\dots,p,\\
C^{\{\nu,\mu\}}(\eta): &\quad \mathbf{A}^{\{\nu,\mu\}}\mathbf{c}^{\{\mu\}\times(k-1)} = \frac{\mathbf{c}^{\{\nu\}\times k}}{k}, && k = 1,\dots,\eta,\\
D^{\{\nu,\mu\}}(\zeta): &\quad \left(\mathbf{b}^{\{\nu\}} \times \mathbf{c}^{\{\nu\}\times(k-1)}\right)^T \mathbf{A}^{\{\nu,\mu\}} = \frac{\mathbf{b}^{\{\mu\}} \times \left(\mathbb{1}_{s^{\{\mu\}}} - \mathbf{c}^{\{\mu\}\times k}\right)}{k}, && k = 1,\dots,\zeta.
\end{aligned}
$$

Note that the B conditions are already required by the classical order conditions and $C^{\{\nu,\mu\}}(1)$ for $\nu,\mu \in \{E, I\}$ is the internal consistency conditions (7.8).

Higher order C simplifying assumptions do not appear feasible for $\mathbb{M}1$ methods as both base methods have stage order one. If one is willing to forego stiff accuracy, $D^{\{I,I\}}(1)$ can be enforced. Instead, the $D^{\{E,E\}}(1)$, $D^{\{E,I\}}(1)$, and $D^{\{I,E\}}(1)$ assumptions appear less restrictive and should be considered first.

For $\mathbb{M}2$ methods, both $C^{\{E,I\}}(2)$ and $C^{\{I,I\}}(2)$ are possible, but $C^{\{I,E\}}(2)$ is practically impossible because $\mathbf{A}^{\{I,E\}}$ is block strictly lower triangular. Now $D^{\{I,I\}}(1)$ is impossible, but the other $D^{\{\nu,\mu\}}(1)$ conditions remain viable.

7.5 IMEX GARK for Index-1 DAEs

Consider the singular perturbation problem

$$
\begin{aligned}
y' &= f(x,z),\\
\varepsilon z' &= g(y,z),
\end{aligned}
\tag{7.26}
$$

where $0 \le \varepsilon \ll 1$. Of interest to this section is the case when $\varepsilon = 0$ and (7.26) becomes a DAE:

$$
\begin{aligned}
y' &= f(y,z),\\
0 &= g(y,z).
\end{aligned}
\tag{7.27}
$$

We will assume $\frac{\partial g}{\partial z}(y,z)$ is invertible in a region about the exact solution so that the index is 1. By applying a GARK method to (7.26) and taking the limit $\varepsilon \to 0$, we arrive at the

following equation to solve an index-1 DAE:

$$\begin{aligned}
Y^{\{E\}} &= \mathbb{1}_{s^{\{E\}}} \otimes y_n + h\left(\mathbf{A}^{\{E,E\}} \otimes I\right) F\big(Y^{\{E\}}, Z^{\{E\}}\big),\\
Z^{\{E\}} &= R_{\text{int}}^{\{E\}}(0,\infty) \otimes z_n + \bar{\mathbf{A}}^{\{E,I\}} \otimes Z^{\{I\}},\\
Y^{\{I\}} &= \mathbb{1}_{s^{\{I\}}} \otimes y_n + h\left(\mathbf{A}^{\{I,E\}} \otimes I\right) F\big(Y^{\{E\}}, Z^{\{E\}}\big),\\
0 &= G\big(Y^{\{I\}}, Z^{\{I\}}\big),\\
y_{n+1} &= y_n + h\left(\mathbf{b}^{\{E\}T} \otimes I\right) F\big(Y^{\{E\}}, Z^{\{E\}}\big),\\
z_{n+1} &= R(0,\infty) z_n + \bar{\mathbf{b}}^{\{I\}T} \otimes Z^{\{I\}}.
\end{aligned} \tag{7.28}$$

The Kronecker product is denoted with $\otimes$, and we use the notation

$$F\big(Y^{\{E\}}, Z^{\{E\}}\big) = \begin{bmatrix} f\left(Y_1^{\{E\}}, Z_1^{\{E\}}\right)\\ \vdots \\ f\left(Y_{s^{\{E\}}}^{\{E\}}, Z_{s^{\{E\}}}^{\{E\}}\right)\end{bmatrix}, \qquad G\big(Y^{\{I\}}, Z^{\{I\}}\big) = \begin{bmatrix} g\left(Y_1^{\{I\}}, Z_1^{\{I\}}\right)\\ \vdots \\ g\left(Y_{s^{\{I\}}}^{\{I\}}, Z_{s^{\{I\}}}^{\{E\}}\right)\end{bmatrix}.$$

The unspecified coefficients in (7.28) are defined as

$$\begin{aligned}
\bar{\mathbf{b}}^{\{I\}T} &= \mathbf{b}^{\{I\}T}\Omega, \qquad \bar{\mathbf{A}}^{\{\nu,I\}} = \mathbf{A}^{\{\nu,I\}}\Omega, \text{ for } \nu \in \{E, I\}\\
\Omega &= \begin{cases} \mathbf{A}^{\{I,I\}\times -1}, & \text{for } \mathbb{M}1 \text{ methods},\\ \begin{bmatrix} 0 & 0_{s^{\{I\}}-1}^T \\ 0_{s^{\{I\}}-1} & \widehat{\mathbf{A}}^{\{I,I\}\times -1}\end{bmatrix}, & \text{for } \mathbb{M}2 \text{ methods}.\end{cases}
\end{aligned} \tag{7.29}$$

If $\mathbf{A}^{\{I,I\}}$ is invertible, e.g., an $\mathbb{M}1$ method, we can equivalently express an IMEX GARK method applied to (7.27) as

$$\begin{aligned}
K_i^{\{E\}} &= hf\left(y_n + \sum_{j=1}^{s^{\{E\}}} a_{i,j}^{\{E,E\}} K_j^{\{E\}}, z_n + \sum_{j=1}^{s^{\{I\}}} a_{i,j}^{\{E,I\}} K_j^{\{I\}}\right), \qquad i = 1,\dots,s^{\{E\}},\\
0 &= g\left(y_n + \sum_{j=1}^{s^{\{E\}}} a_{i,j}^{\{I,E\}} K_j^{\{E\}}, z_n + \sum_{j=1}^{s^{\{I\}}} a_{i,j}^{\{I,I\}} K_j^{\{I\}}\right), \qquad i = 1,\dots,s^{\{I\}},\\
y_{n+1} &= y_n + \sum_{i=1}^{s^{\{E\}}} b_i^{\{E\}} K_j^{\{E\}},\\
z_{n+1} &= z_n + \sum_{i=1}^{s^{\{I\}}} b_i^{\{I\}} K_j^{\{I\}}.
\end{aligned}$$

This form is useful from an implementation standpoint, but for the analysis, we will continue with (7.28).

An implicit assumption in (7.28) is that $R_{\text{int}}^{\{E\}}(0,\infty)$ is finite, but for general $\mathbb{M}2$ methods, this is not guaranteed. From section 7.3, we know $R_{\text{int}}^{\{E\}}(0,\infty)$ is finite if and only if (7.17a) holds. Further, note that before we take $\varepsilon \to 0$ in (7.28),

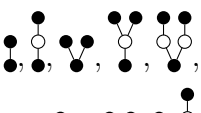
$$Z^{\{E\}} = \left(\mathbb{1} - \bar{\mathbf{A}}^{\{E,I\}}\mathbb{1}\right) \otimes z_n + \bar{\mathbf{A}}^{\{E,I\}} \otimes Z^{\{I\}} + \begin{bmatrix} 0 \\ \frac{h}{\varepsilon}\widehat{r} \otimes g\left(Y_1^{\{I\}}, Z_1^{\{I\}}\right) \end{bmatrix}.$$

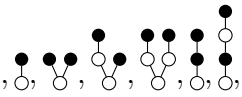
If $Y_1^{\{I\}}$ and $Z_1^{\{I\}}$ fail to satisfy the algebraic constraint exactly and $\widehat{r}$ (from (7.16)) is nonzero, the last term can diverge as $\varepsilon \to 0$. Again, this shows $\mathbb{M}2$ methods must satisfy the simplifying assumption (7.17a) to be well-posed for index-1 DAEs.

7.5.1 Order Conditions

Our error analysis for index-1 DAEs uses the set of DAE trees [72, 124]

$$DAT = DAT_y \cup DAT_z,$$

$DAT_y = \{\emptyset,$  $\}$,

$DAT_z = \{\emptyset,$  $\}$.

"Meager," black vertices correspond to the differential function f, and "fat," white vertices correspond to the algebraic function g. For $\mathfrak{t} \in DAT$, the order $\rho(\mathfrak{t})$ is the number of meager vertices in the tree. If $\mathfrak{t}_1, \ldots, \mathfrak{t}_m \in DAT_y$ and $\mathfrak{u}_1, \ldots, \mathfrak{u}_n \in DAT_z$, then $[\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_y$ is the tree formed by connecting the roots of all $m+n$ subtrees to a new, meager root. The tree $[\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_z$ is defined similarly but with a fat root. The density of a tree is defined by

$$\begin{aligned}
&\gamma(\emptyset) = \gamma(\bullet) = 1, \\
&\gamma(\mathfrak{t}) = \rho(\mathfrak{t})\gamma(\mathfrak{t}_1)\cdots\gamma(\mathfrak{t}_m)\gamma(\mathfrak{u}_1)\cdots\gamma(\mathfrak{u}_m), && \text{if } \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_y, \\
&\gamma(\mathfrak{u}) = \gamma(\mathfrak{t}_1)\cdots\gamma(\mathfrak{t}_m)\gamma(\mathfrak{u}_1)\cdots\gamma(\mathfrak{u}_m), && \text{if } \mathfrak{u} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_z.
\end{aligned}$$

Now we are ready to define the order conditions.

Theorem 7.3 (Order conditions for index-1 DAEs)**.** *The IMEX GARK method* (7.28) *has differential order p_y and algebraic order p_z, that is*

$$y(t_0 + h) - y_1 = \mathcal{O}\left(h^{p_y+1}\right), \qquad z(t_0 + h) - z_1 = \mathcal{O}\left(h^{p_z+1}\right), \tag{7.30}$$

if and only if

$$\Phi(\mathfrak{t}) = \frac{1}{\gamma(\mathfrak{t})}, \quad \forall \mathfrak{t} \in DAT_y \quad 1 \le \rho(\mathfrak{t}) \le p_y, \tag{7.31a}$$

$$\Phi(\mathfrak{u}) = \frac{1}{\gamma(\mathfrak{u})}, \quad \forall \mathfrak{u} \in DAT_z \quad 1 \le \rho(\mathfrak{u}) \le p_z, \tag{7.31b}$$

where the elementary weights are

$$\Phi(\mathfrak{t}) = \begin{cases} \mathbf{b}^{\{E\}T} \mathbb{1}_{s^{\{E\}}}, & \mathfrak{t} = \bullet, \\ \mathbf{b}^{\{E\}T} \left(\Phi^{\{E\}}(\mathfrak{t}_1) \times \cdots \times \Phi^{\{E\}}(\mathfrak{t}_m) \right. \\ \quad \left. \times \Phi^{\{E\}}(\mathfrak{u}_1) \times \cdots \times \Phi^{\{E\}}(\mathfrak{u}_m)\right), & \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_y, \\ \bar{\mathbf{b}}^{\{I\}T} \left(\Phi^{\{I\}}(\mathfrak{t}_1) \times \cdots \times \Phi^{\{I\}}(\mathfrak{t}_m) \right. \\ \quad \left. \times \Phi^{\{I\}}(\mathfrak{u}_1) \times \cdots \times \Phi^{\{I\}}(\mathfrak{u}_m)\right), & \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_z, \end{cases}$$

$$\Phi^{\{\nu\}}(\mathfrak{t}) = \begin{cases} \mathbf{c}^{\{\nu,E\}}, & \mathfrak{t} = \bullet, \\ \mathbf{A}^{\{\nu,E\}} \left(\Phi^{\{E\}}(\mathfrak{t}_1) \times \cdots \times \Phi^{\{E\}}(\mathfrak{t}_m) \right. \\ \quad \left. \times \Phi^{\{E\}}(\mathfrak{u}_1) \times \cdots \times \Phi^{\{E\}}(\mathfrak{u}_m)\right), & \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_y, \\ \bar{\mathbf{A}}^{\{\nu,I\}} \left(\Phi^{\{I\}}(\mathfrak{t}_1) \times \cdots \times \Phi^{\{I\}}(\mathfrak{t}_m) \right. \\ \quad \left. \times \Phi^{\{I\}}(\mathfrak{u}_1) \times \cdots \times \Phi^{\{I\}}(\mathfrak{u}_m)\right), & \mathfrak{t} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m, \mathfrak{u}_1, \ldots, \mathfrak{u}_n]_z. \end{cases}$$

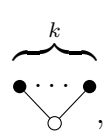
Remark 7.4. Due to the structure of the elementary weights, there are redundant order conditions produced by DAE trees in which a fat vertex has a fat child. In fact, by collapsing connected, fat vertices into a single fat vertex, the density and order conditions are unchanged [153, Corollary 3.7]. It suffices to only consider DAT_z trees of the form $[\mathfrak{t}_1, \ldots, \mathfrak{t}_m]_z$.

Proof. This follows from the DA-series of GARK methods derived by Tanner in [153, Theorem 3.6]. We note that the referenced theorem defines $\Omega = \mathbf{A}^{\{I,I\}\times -1}$ like we do for $\mathbb{M}1$ methods in (7.29). The DA-series remain the same even with our more general definition of Ω that supports $\mathbb{M}2$ methods. □

DAE order conditions up to $p_y = p_z = 3$ are listed in tables 7.1 to 7.3. We have used proposition 7.4 to eliminate redundant order conditions.

In (7.30), the power of h in the leading term of the differential error may be different than that of the algebraic error. Error controllers typically expect the local truncation error to have a consistent power of h, though. For adaptive methods, it will be helpful to impose $p_z = p_y$ for the main method and $\widehat{p}_z = \widehat{p}_y$ for the embedded method. Then, embedded methods and error controllers can be used just as they would when solving ODEs.

When a DAE tree has a branch of the form



,

Name	Tree	$\Phi(\mathfrak{t})$	$\gamma(\mathfrak{t})$
$\mathfrak{t}_1$		$\mathbf{b}^{\{E\}T}\mathbb{1}_{s^{\{E\}}}$	1
$\mathfrak{t}_{2,1}$		$\mathbf{b}^{\{E\}T}\mathbf{c}^{\{E,E\}}$	2
$\mathfrak{t}_{2,2}$		$\mathbf{b}^{\{E\}T}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}$	2
$\mathfrak{u}_1$		$\bar{\mathbf{b}}^{\{I\}}\mathbf{c}^{\{I,E\}}$	1
$\mathfrak{u}_{2,1}$		$\bar{\mathbf{b}}^{\{I\}}\mathbf{c}^{\{I,E\}\times 2}$	1
$\mathfrak{u}_{2,2}$		$\bar{\mathbf{b}}^{\{I\}}\mathbf{A}^{\{I,E\}}\mathbf{c}^{\{E,E\}}$	2
$\mathfrak{u}_{2,3}$		$\bar{\mathbf{b}}^{\{I\}}\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}$	2

Table 7.1: GARK order conditions and corresponding DAT trees up to order two. For internally consistent methods, $\mathfrak{t}_{2,2}$, $\mathfrak{u}_1$, and $\mathfrak{u}_{2,3}$ are redundant.

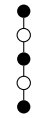
Name	Tree	$\Phi(\mathfrak{t})$	$\gamma(\mathfrak{t})$
$\mathfrak{t}_{3,1}$		$\mathbf{b}^{\{E\}T}\mathbf{c}^{\{E,E\}\times 2}$	3
$\mathfrak{t}_{3,2}$		$\mathbf{b}^{\{E\}T}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}\times 2}$	3
$\mathfrak{t}_{3,3}$		$\mathbf{b}^{\{E\}T}\left(\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}\times\mathbf{c}^{\{E,E\}}\right)$	3
$\mathfrak{t}_{3,4}$		$\mathbf{b}^{\{E\}T}\left(\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}\right)^2$	3
$\mathfrak{t}_{3,5}$		$\mathbf{b}^{\{E\}T}\mathbf{A}^{\{E,E\}}\mathbf{c}^{\{E,E\}}$	6
$\mathfrak{t}_{3,6}$		$\mathbf{b}^{\{E\}T}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{A}^{\{I,E\}}\mathbf{c}^{\{E,E\}}$	6
$\mathfrak{t}_{3,7}$		$\mathbf{b}^{\{E\}T}\mathbf{A}^{\{E,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}$	6
$\mathfrak{t}_{3,8}$		$\mathbf{b}^{\{E\}T}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}$	6

Table 7.2: GARK order conditions and corresponding DAT_y trees of order three. For internally consistent methods, $\mathfrak{t}_{3,3}$, $\mathfrak{t}_{3,4}$, $\mathfrak{t}_{3,7}$, and $\mathfrak{u}_{3,8}$ are redundant.

for $k \geq 1$, the corresponding order condition is of the form $(\dots)\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}\times k}$. Suppose the $C^{\{E,I\}}(k)$ and $C^{\{I,I\}}(k)$ simplifying assumptions hold. For $\mathbb{M}1$ methods, one can easily verify that $\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}\times k} = \mathbf{c}^{\{E\}\times k}$. In terms of the tree, this removes the fat vertex and attaches the k leaves to its parent. In particular, internal consistency implies trees with  as a branch have order conditions that coincide with trees without the fat vertex. This is also

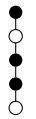
Name	Tree	$\Phi(\mathfrak{t})$	$\gamma(\mathfrak{t})$
$\mathfrak{u}_{3,1}$		$\bar{\mathbf{b}}^{\{I\}T}\left(\mathbf{A}^{\{I,E\}}\mathbf{c}^{\{E,E\}} \times \mathbf{c}^{\{I,E\}}\right)$	2
$\mathfrak{u}_{3,2}$		$\bar{\mathbf{b}}^{\{I\}T}\left(\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}} \times \mathbf{c}^{\{I,E\}}\right)$	2
$\mathfrak{u}_{3,3}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{c}^{\{I,E\}\times 3}$	1
$\mathfrak{u}_{3,4}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\mathbf{c}^{\{E,E\}\times 2}$	3
$\mathfrak{u}_{3,5}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}\times 2}$	3
$\mathfrak{u}_{3,6}$		$\bar{\mathbf{b}}^{\{I\}T}A^{\{I,E\}}\left(\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}} \times \mathbf{c}^{\{E,E\}}\right)$	3
$\mathfrak{u}_{3,7}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\left(\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}\right)^2$	3
$\mathfrak{u}_{3,8}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\mathbf{A}^{\{E,E\}}\mathbf{c}^{\{E,E\}}$	6
$\mathfrak{u}_{3,9}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\mathbf{A}^{\{E,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}$	6
$\mathfrak{u}_{3,10}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{A}^{\{I,E\}}\mathbf{c}^{\{E,E\}}$	6
$\mathfrak{u}_{3,11}$		$\bar{\mathbf{b}}^{\{I\}T}\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{A}^{\{I,E\}}\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}}$	6

Table 7.3: GARK order conditions and corresponding DAT_z trees of order three. For internally consistent methods, $\mathfrak{u}_{3,2}$, $\mathfrak{u}_{3,6}$, $\mathfrak{u}_{3,7}$, $\mathfrak{u}_{3,9}$, and $\mathfrak{u}_{3,11}$ are redundant.

true for $\mathbb{M}2$ methods, but some care is needed when $k = 1$:

$$\bar{\mathbf{A}}^{\{E,I\}}\mathbf{c}^{\{I,E\}} = \mathbf{c}^{\{E\}} + \begin{bmatrix} 0 \\ \widehat{r} \end{bmatrix}.$$

Note $\widehat{r} = 0_{s^{\{E\}}-1}$ because (7.17a) is already required for an $\mathbb{M}2$ method to be well-posed for DAEs.

As we will see in the following theorems, stiff accuracy also simplifies the index-1 DAE order conditions.

Theorem 7.5. *If the IMEX GARK method* (7.28) *is stiffly accurate in the algebraic partition* (7.11b) *and satisfies the differential order conditions* (7.31a) *up to order* p_y*, then the algebraic order conditions* (7.31b) *are satisfied up to* $p_z = p_y$*.*

Proof. To prove (7.31b), we can use proposition 7.4 and only consider DAT_z trees of the form $\mathfrak{u} = [\mathfrak{t}_1, \ldots, \mathfrak{t}_m]_z$ where $\rho(\mathfrak{u}) \leq p_y$. By the assumption (7.31a), the order condition for each differential tree $\mathfrak{t}_i$, for $i = 1, \ldots, m$, is satisfied. Lemma 3.11 from [153] implies the order condition for $\mathfrak{u}$ also holds. Thus, (7.31b) where $p_z = p_y$. □

Theorem 7.6. *Consider a stiffly accurate IMEX ARK method (7.2) where the implicit base method is diagonally implicit, possibly with an explicit first stage. If the IMEX ARK method is order p for ODEs, it is also order p for index-1 DAEs.*

Proof. From the implicit function theorem, there exists a function $\mathcal{G}$ such that $z = \mathcal{G}(y)$ and $y' = f(y, \mathcal{G}(y))$. If the explicit base method of the IMEX ARK method is applied to this ODE, we get the state space form method

$$\begin{aligned} Y &= \mathbb{1}_s \otimes y_n + h\left(\mathbf{A}^{\{E\}} \otimes I\right) F(Y, Z), \\ 0 &= G(Y, Z), \\ y_{n+1} &= Y_s, \\ z_{n+1} &= Z_s. \end{aligned} \tag{7.32}$$

This method is order p for the DAE (7.27). If the stiffly accurate IMEX ARK method is applied to the DAE (7.27), we also arrive at (7.32). This holds for $\mathbf{A}^{\{I,I\}}$ invertible or when it has an ESDIRK structure. Thus, our IMEX ARK method is order p for index-1 DAEs. See also [22, page 1612]. □

7.5.2 Global Error and Convergence

The convergence of IMEX GARK methods for index-1 DAEs can be proved using a wonderfully general theorem from [47]:

Theorem 7.7. *Assume the generic one-step method*

$$\begin{aligned} y_{n+1} &= y_n + h\phi(y_n, z_n, h), \\ z_{n+1} &= \psi(y_n, z_n, h), \end{aligned} \tag{7.33}$$

for the DAE (7.27) has the local truncation error

$$y_1 = y_0 + h\phi(y_0, z_0, h) + \mathcal{O}\big(h^{p+1}\big), \qquad z_1 = \psi(y_0, z_0, h) + \mathcal{O}(h^p),$$

and $\left\|\frac{\partial \psi}{\partial z}(y, z, 0)\right\| \leq 1$ *in a neighborhood about the solution. Then for* $t_f = t_0 + nh$ *fixed, the method is convergent of order p:*

$$y_n = y(t_n) + \mathcal{O}(h^p) \quad \textit{and} \quad z_n = z(t_n) + \mathcal{O}(h^p).$$

Indeed, IMEX GARK methods fit into the generic formulation (7.33). We need

$$\left\| \frac{\partial \psi}{\partial z}(y, z, 0) \right\| = |R(0, -\infty)| = \left| R^{\{I\}}(-\infty) \right| < 1, \tag{7.34}$$

where $R^{\{I\}}$ is the linear stability function of the implicit base method. It suffices to choose an implicit base method that is L-stable or strongly A-stable. We can conclude that an IMEX GARK method with differential order p_y, algebraic order p_z, and (7.34) will converge with the global error

$$y_n = y(t_n) + \mathcal{O}\big(h^{\min(p_y, p_z+1)}\big) \quad \text{and} \quad z_n = z(t_n) + \mathcal{O}\big(h^{\min(p_y, p_z+1)}\big).$$

7.6 New IMEX GARK Methods

Several new IMEX GARK methods are derived and presented in this section. Following the notation of [85], IMEX GARK methods will be named GARK$p(\widehat{p})s^{\{E\}}s^{\{I\}}S[q_{\text{so}}]X$, where p is the order, $\widehat{p}$ is the embedded method order, $s^{\{E\}}$ is the number of explicit stages, $s^{\{I\}}$ is the number of implicit stages, S describes the implicit stability, q_{so} is the implicit stage order ($C^{\{I,I\}}(q_{\text{so}})$), and X is for any other notable property.

7.6.1 A Second Order $\mathbb{M}1$ Method

To introduce and motivate the derivation of IMEX GARK methods, we will consider three types of IMEX methods based on the second order Runge–Kutta schemes

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c_2 & c_2 & 0 \\ \hline & 1-\frac{1}{2c_2} & \frac{1}{2c_2} \end{array} \quad \text{and} \quad \begin{array}{c|cc} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 1 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\ \hline & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{array}, \tag{7.35}$$

where $c_2 \neq 0$. The implicit method is L-stable and stiffly accurate: a property we would like to preserve in an IMEX method. In fact, stiff accuracy and internal consistency uniquely define the second order GARK method

$$\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ c_2 & 0 & c_2 & 0 \\ \hline 1-\frac{1}{\sqrt{2}} & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\ 1-\frac{1}{2c_2} & \frac{1}{2c_2} & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\ \hline 1-\frac{1}{2c_2} & \frac{1}{2c_2} & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{array}.$$

The principal error of this $\mathbb{M}1$ method is

$$A^{(3)} = \frac{\sqrt{6|2-3c_2|^2 - 324\sqrt{2} + 467}}{12},$$

which achieves a minimal value of approximately 0.247 when $c_2 = \frac{2}{3}$. In that case, the explicit base method becomes Ralston's optimal, two stage, second order Runge–Kutta method [114].

If we use (7.35) to construct an ASIRK method (7.3), we need $\mathbf{b}^{\{E\}} = \mathbf{b}^{\{I\}}$. Thus, $c_2 = 1+\frac{1}{\sqrt{2}}$, which links the two base methods together and no longer provides a parameterized family. Cast into the GARK framework, the ASIRK method is

$$\begin{array}{cc|cc}
0 & 0 & 0 & 0 \\
1+\frac{1}{\sqrt{2}} & 0 & 1+\frac{1}{\sqrt{2}} & 0 \\
\hline
1-\frac{1}{\sqrt{2}} & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\
\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}}
\end{array}.$$

Now, $A^{(3)} \approx 0.683$ which is over 2.75 times larger than that of the GARK method.

An ARK method (7.2) based on (7.35) requires padding of the methods with unused stages and $c_2 = 1-\frac{1}{\sqrt{2}}$ for internal consistency to hold. A method with these properties has already been proposed in [14, Section 2.6]. Cast into the GARK framework, it reads

$$\begin{array}{ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 \\
1-\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\
-\frac{1}{\sqrt{2}} & 1+\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
1-\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\
-\frac{1}{\sqrt{2}} & 1+\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
-\frac{1}{\sqrt{2}} & 1+\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}}
\end{array}.$$

With $A^{(3)} \approx 0.337$, the error is 36% larger than the GARK error.

Only the GARK framework allows c_2 to remain as a free parameter when internal consistency is imposed, and it offers the smallest error. Of course, stability also plays a role in the relative efficiency of these methods. Interestingly, all three share the same linear stability function:

$$R\left(z^{\{E\}}, z^{\{I\}}\right) = \frac{\left(2\sqrt{2}+3\right)\left(z^{\{E\}\times 2} + 2\left(\sqrt{2}-1\right)\left(z^{\{E\}}+1\right)z^{\{I\}} + 2z^{\{E\}} + 2\right)}{\left(-z^{\{I\}} + \sqrt{2} + 2\right)^2}.$$

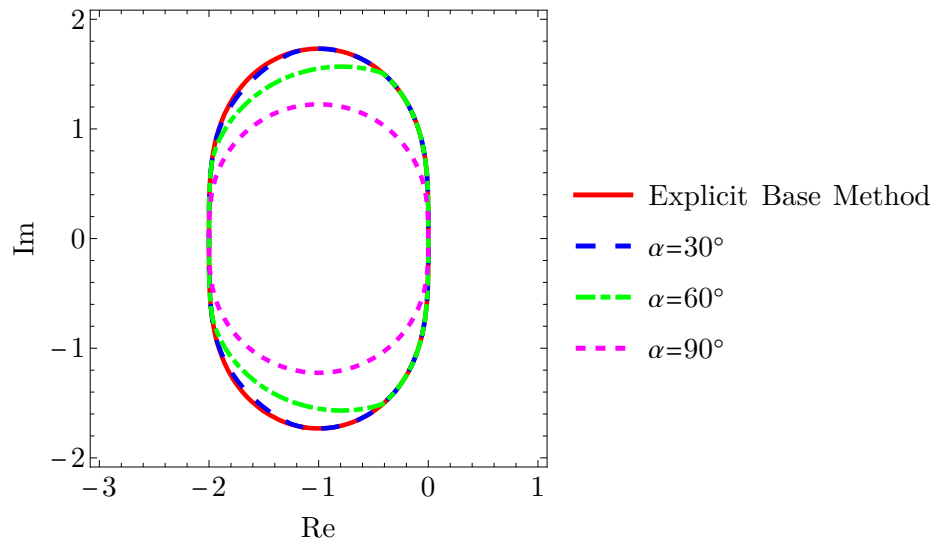


Figure 7.1: Stability regions for (7.36) and the other second order IMEX methods that share its linear stability function. This figure includes the stability region of the explicit base method and $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ for three values of α.

From the stiff accuracy condition (7.11b), $R\big(z^{\{E\}}, -\infty\big) = 0$. Stability plots for the stability function can be found in fig. 7.1.

To conclude, our new method GARK2(1)22L[1]SA uses the optimal value $c_2 = \frac{2}{3}$ and includes an embedded method of order one:

$$\begin{array}{cc|cc}
0 & 0 & 0 & 0 \\
\frac{2}{3} & 0 & \frac{2}{3} & 0 \\
\hline
1-\frac{1}{\sqrt{2}} & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\
\frac{1}{4} & \frac{3}{4} & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
\frac{1}{4} & \frac{3}{4} & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
0 & 1 & \frac{2\sqrt{2}-1}{3} & \frac{2(2-\sqrt{2})}{3}
\end{array} . \tag{7.36}$$

7.6.2 A Third Order $\mathbb{M}2$ Method

For the derivation of a third order IMEX GARK scheme, we start by independently selecting optimized base methods. In the survey of diagonally implicit Runge–Kutta by Kennedy and Carpenter [86], the method ESDIRK3(2)5L[2]SA was found to be one of the best at order

three. It is L-stable, uses the $C^{\{I,I\}}(2)$ simplifying assumption, and has the tableau

$$
\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{20} & \frac{9}{40} & \frac{9}{40} & 0 & 0 & 0 \\
\frac{9(\sqrt{2}+2)}{40} & \frac{9(\sqrt{2}+1)}{80} & \frac{9(\sqrt{2}+1)}{80} & \frac{9}{40} & 0 & 0 \\
\frac{3}{5} & \frac{7\sqrt{2}+8}{80} & \frac{(7\sqrt{2}+8)}{80} & \frac{-7(\sqrt{2}-1)}{40} & \frac{9}{40} & 0 \\
1 & \frac{1187\sqrt{2}-1181}{2835} & \frac{1187\sqrt{2}-1181}{2835} & -\frac{2374(\sqrt{2}-1)}{2835} & \frac{5827}{7560} & \frac{9}{40} \\
\hline
 & \frac{1187\sqrt{2}-1181}{2835} & \frac{1187\sqrt{2}-1181}{2835} & -\frac{2374(\sqrt{2}-1)}{2835} & \frac{5827}{7560} & \frac{9}{40} \\
\hline
 & \frac{5547709\sqrt{2}-4800247}{16519545} & \frac{5547709\sqrt{2}-4800247}{16519545} & -\frac{11095418(\sqrt{2}-1)}{16519545} & \frac{30698249}{44052120} & \frac{49563}{233080}
\end{array}
.
$$

We derive our explicit method from scratch because there are few existing methods with five stages. With the excess of coefficients, we impose $D^{\{E,E\}}(1)$, minimize the error $A^{\{E\}(p+1)}$, and ensure a large stability region. The resulting tableau is

$$
\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\
\frac{2}{3} & 0 & \frac{2}{3} & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
1 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 \\
\hline
 & -\frac{391709805}{8420574392} & \frac{5377304043}{8420574392} & \frac{98431707}{271631432} & \frac{5507}{46616} & -\frac{9}{124}
\end{array}
.
$$

For the overall $\mathbb{M}2$ scheme, classical order conditions up to order three are needed, and we impose internal consistency. With $s^{\{E\}} = s^{\{I\}} = 5$, there are enough degrees of freedom to satisfy (7.17a) and the DAE order conditions up to $p_y = p_z = 3$. Proposition 7.5 helps to reduce the number of these order conditions. Free parameters are used to control the internal stability and minimize the error. The new method GARK3(2)55L[2]DAE is given

Method	GARK3(2)55L[2]DAE	ARK3(2)4L[2]SA	BHR(5,5,3) $\gamma \approx \frac{424782}{974569}$	BHR(5,5,3) $\gamma \approx \frac{2051948}{3582211}$
Source	(7.37)	[85]	[23]	[23]
$(s^{\{E\}}, s^{\{I\}})$	$(5,5)$	$(4,4)$	$(5,5)$	$(5,5)$
FSAL	Yes	No	No	No
Diff. Order p_y	3	3	3	3
Alg. Order p_z	3	1	2	2
$A^{(4)}$	0.0508	0.1663	0.6309	0.1995
$A^{\{E\}(4)}$	0.0196	0.0224	0.1446	0.0373
$A^{\{I\}(4)}$	0.00078	0.0366	0.1491	0.0373
$B^{(4)}$	1.37	3.51	—	—
$C^{(4)}$	1.38	1.55	—	—
D	1	1.04	3.14	2.33
$E^{(4)}$	0.52	4.31	—	—
$R(z^{\{E\}}, -\infty)$	0	0	0	$0.852z^{\{E\}}$

Table 7.4: Properties of third order IMEX methods.

by

$$
\begin{array}{ccccc|ccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 \\
0 & \frac{2}{3} & 0 & 0 & 0 & \frac{8-3\sqrt{2}}{15} & \frac{8-3\sqrt{2}}{15} & \frac{2(\sqrt{2}-1)}{5} & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \frac{743-131\sqrt{2}}{1890} & \frac{743-131\sqrt{2}}{1890} & \frac{131(\sqrt{2}-1)}{945} & \frac{37}{105} & 0 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \frac{1187\sqrt{2}-1181}{2835} & \frac{1187\sqrt{2}-1181}{2835} & \frac{2374(1-\sqrt{2})}{2835} & \frac{5827}{7560} & \frac{9}{40} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{20} & 0 & 0 & 0 & 0 & \frac{9}{40} & \frac{9}{40} & 0 & 0 & 0 \\
\frac{9(52-271\sqrt{2})}{12920} & \frac{2673(\sqrt{2}+1)}{6460} & 0 & 0 & 0 & \frac{9(\sqrt{2}+1)}{80} & \frac{9(\sqrt{2}+1)}{80} & \frac{9}{40} & 0 & 0 \\
-\frac{881835}{7528484} & \frac{7282818}{9410605} & -\frac{1323}{23308} & 0 & 0 & \frac{7\sqrt{2}+8}{80} & \frac{7\sqrt{2}+8}{80} & \frac{7(1-\sqrt{2})}{40} & \frac{9}{40} & 0 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \frac{1187\sqrt{2}-1181}{2835} & \frac{1187\sqrt{2}-1181}{2835} & \frac{2374(1-\sqrt{2})}{2835} & \frac{5827}{7560} & \frac{9}{40} \\
\hline
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \frac{1187\sqrt{2}-1181}{2835} & \frac{1187\sqrt{2}-1181}{2835} & \frac{2374(1-\sqrt{2})}{2835} & \frac{5827}{7560} & \frac{9}{40} \\
\hline
-\frac{391709805}{8420574392} & \frac{5377304043}{8420574392} & \frac{98431707}{271631432} & \frac{5507}{46616} & -\frac{9}{124} & \frac{5547709\sqrt{2}-4800247}{16519545} & \frac{5547709\sqrt{2}-4800247}{16519545} & \frac{11095418(1-\sqrt{2})}{16519545} & \frac{30698249}{44052120} & \frac{49563}{233080}
\end{array}
\tag{7.37}
$$

Proposition 7.1 can be used by setting $v = e_1 \in \mathbb{R}^{s^{\{I\}}-1}$ in (7.20). Not only does this prove $R\big(z^{\{E\}}, -\infty\big) = 0$ but it explains why the first two columns of $\mathbf{A}^{\{E,I\}}$ and $\mathbf{A}^{\{I,I\}}$ are identical. While additional simplifying assumptions such as $C^{\{E,I\}}(2)$ are possible, it constrains too many coefficients to achieve a small principal error.

Figure 7.2 plots the stability for GARK3(2)55[2]DAE, and table 7.4 compares the method with existing IMEX ARK methods. Notably, GARK3(2)55[2]DAE has a principal error over three times smaller than the other methods.

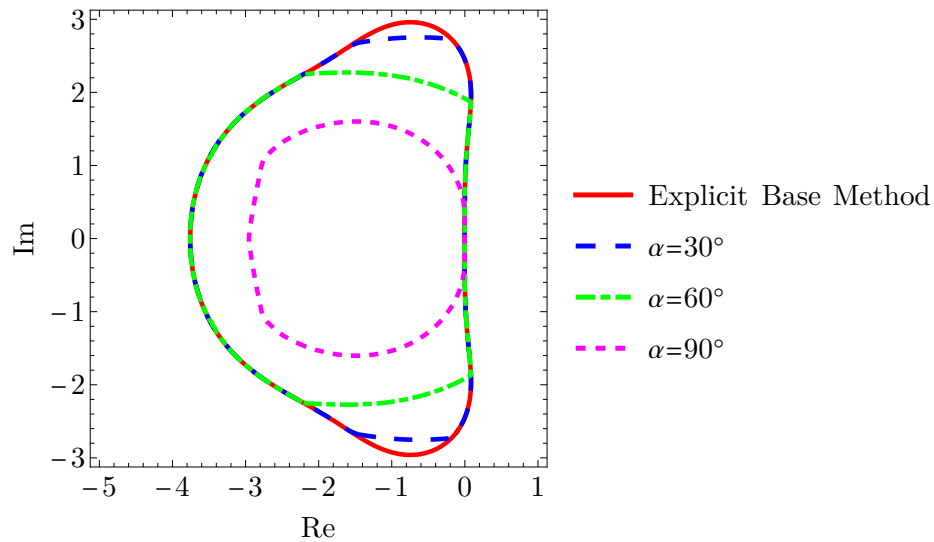


Figure 7.2: Stability regions for (7.37) including the explicit base method and $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ for three values of α.

7.6.3 Fourth order IMEX Methods

At order four, the number of order conditions grows rapidly. Without any simplifying assumptions, there are 72 ODE order conditions and 289 DAE order conditions. Internal consistency and stiff accuracy are critical to bring this to a manageable number.

To start, we will consider two IMEX GARK methods designed for ODEs only. The first, GARK4(3)55L[1]SA, is provided in appendix E.1. It has the $\mathbb{M}1$ structure, and for simplicity, it uses relatively few stages with $s^{\{E\}} = s^{\{I\}} = 5$. On the other hand, GARK4(3)77L[2]SA in appendix E.2 has the $\mathbb{M}2$ structure and uses additional stages to achieve a significantly smaller principal error. Stability is more challenging to control at high orders, and the $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ regions in figs. E.1 and E.2 cover a slightly smaller percentage of the explicit stability region compared to the second and third order methods. Nevertheless, GARK4(3)55L[1]SA appears to be the method best equipped to solve (7.1) when mild stiffness in $f^{\{E\}}$ limits the timestep.

For DAEs, we can bypass the extra order conditions by using stiffly accurate IMEX ARK methods as discussed in proposition 7.6. In a divergence from the derivation of most high-order IMEX ARK methods, we do not use the simplifying assumption $\mathbf{b}^{\{E\}} = \mathbf{b}^{\{I\}}$ because it is incompatible with stiff accuracy. We are still able to use internal consistency and an implicit method with stage order two. Our new method, ARK4(3)8L[2]DAE, and its stability can be found in appendix E.3.

Our fourth order methods and two existing IMEX ARK methods are compared in table 7.5. Among methods with seven or fewer stages, GARK4(3)77L[2]SA achieves the smallest $A^{(5)}$, but the overall smallest $A^{(5)}$ belongs to the eight stage ARK4(3)8L[2]DAE. Note that without

Method	GARK4(3)55L[1]SA	ARK4(3)6L[2]SA	GARK4(3)77L[2]SA	ARK4(3)8L[2]DAE	ARK4(3)7L[2]SA$_1$
Source	(E.1)	[85]	(E.2)	(E.3)	[88]
$(s^{\{E\}}, s^{\{I\}})$	$(5,5)$	$(6,6)$	$(7,7)$	$(8,8)$	$(7,7)$
FSAL	No	No	Yes	Yes	No
Diff. Order p_y	2	4	2	4	4
Alg. Order p_z	2	2	2	4	2
$A^{(5)}$	0.04293	0.03576	0.00792	0.00641	0.01112
$A^{\{E\}(5)}$	0.00471	0.0224	0.00091	0.00130	0.00195
$A^{\{I\}(5)}$	0.00201	0.0366	0.00026	0.00039	0.00163
$B^{(5)}$	1.92	5.83	2.01	1.44	13.01
$C^{(5)}$	1.87	1.88	1.98	1.35	1.84
D	3.03	1.06	2.65	5.67	7.36
$E^{(5)}$	0.59	4.21	0.32	0.42	14.36
$R(z^{\{E\}}, -\infty)$	0	0	0	0	0

Table 7.5: Properties of fourth order IMEX methods.

specifically enforcing DAE order conditions, our fourth order IMEX GARK methods only resolve the differential variables of a DAE to order two.

7.7 Numerical Tests

Next, we will evaluate the convergence and performance properties of IMEX GARK by applying them to two test problems.

7.7.1 The ZLA-Kinetics DAE

In order to test the convergence properties of methods designed for index-1 DAEs, we will apply them to the simple, nonlinear ZLA-kinetics problem [151]:

$$\begin{aligned} y_1' &= -2r_1 + r_2 - r_3 - r_4, & y_2' &= -\frac{1}{2}r_1 - r_4 - \frac{1}{2}r_5 + F_{\text{in}}, \\ y_3' &= r_1 - r_2 + r_3, & y_4' &= -r_2 + r_3 - 2r_4, \\ y_5' &= r_2 - r_3 + r_5, & 0 &= K_s y_1 y_4 - y_6. \end{aligned} \tag{7.38}$$

The model parameters are given in table 7.6, and auxiliary variables are defined as

$$r_1 = k_1\, y_1^4\, y_2^{1/2}, \quad r_2 = k_2\, y_3\, y_4, \qquad r_3 = (k_2/K)\, y_1\, y_5,$$
$$r_4 = k_3\, y_1\, y_4^2, \qquad r_5 = k_4\, y_6^2\, y_2^{1/2}, \quad F_{\text{in}} = klA\,(p(\text{CO}_2)/H - y_2)\,.$$

The initial condition

Parameter	k_1	k_2	k_3	k_4	K	klA	K_s	$p(\text{CO}_2)$	H
Value	18.7	0.58	0.09	0.42	34.4	3.3	115.83	0.9	737

Table 7.6: Values of parameters appearing in the ZLA-kinetics problem (7.38)

$$y(t_0 = 0) = \begin{bmatrix} 0.444 & 0.00123 & 0 & 0.007 & 0 & K_s\, y_{0,1}\, y_{0,4} \end{bmatrix}^T,$$

is consistent with the algebraic constraint, and we seek the solution at $t_f = 180$.

For IMEX methods, the five differential equations in (7.38) define f and the one algebraic constrain defines g. We will compare the convergence of GARK3(2)55L[2]DAE and ARK4(3)8L[2]DAE with other IMEX Runge–Kutta methods suitable for DAEs. From [23], we use the method BHR(5,5,3) which satisfies order conditions for both index-1 and index-2 DAEs. We also apply extrapolation using the harmonic sequence to IMEX Euler,

$$y_{n+1} = y_n + hf(y_n, z_n),$$
$$z_{n+1} = g(y_{n+1}, z_{n+1}),$$

to generate IMEX methods of order $p = 3, 4$. These can be viewed as ASIRK methods with $s = \frac{p(p+1)}{2}$ stages.

In the experiment, each method solves (7.38) using 20 different values of h. Using a high-accuracy reference solution, each error is computed using the 2-norm. The convergence results are plotted in fig. 7.3. Indeed, every method converges at the theoretically predicted order. For a fixed number of steps, GARK3(2)55L[2]DAE and ARK4(3)8L[2]DAE provide the smallest errors for their respective orders.

7.7.2 The BSVD Reaction-Diffusion PDE

The BSVD equation [76] is a bistable reaction-diffusion PDE governed by

$$\begin{aligned} \frac{\partial u}{\partial y} &= \nabla \cdot (D(x,y)\nabla u) + 10(1-u^2)(u+0.6), \\ u(0,x,y) &= 2\exp\big(-10((x-0.5)^2 + (y+0.1)^2)\big) - 1, \end{aligned} \tag{7.39}$$

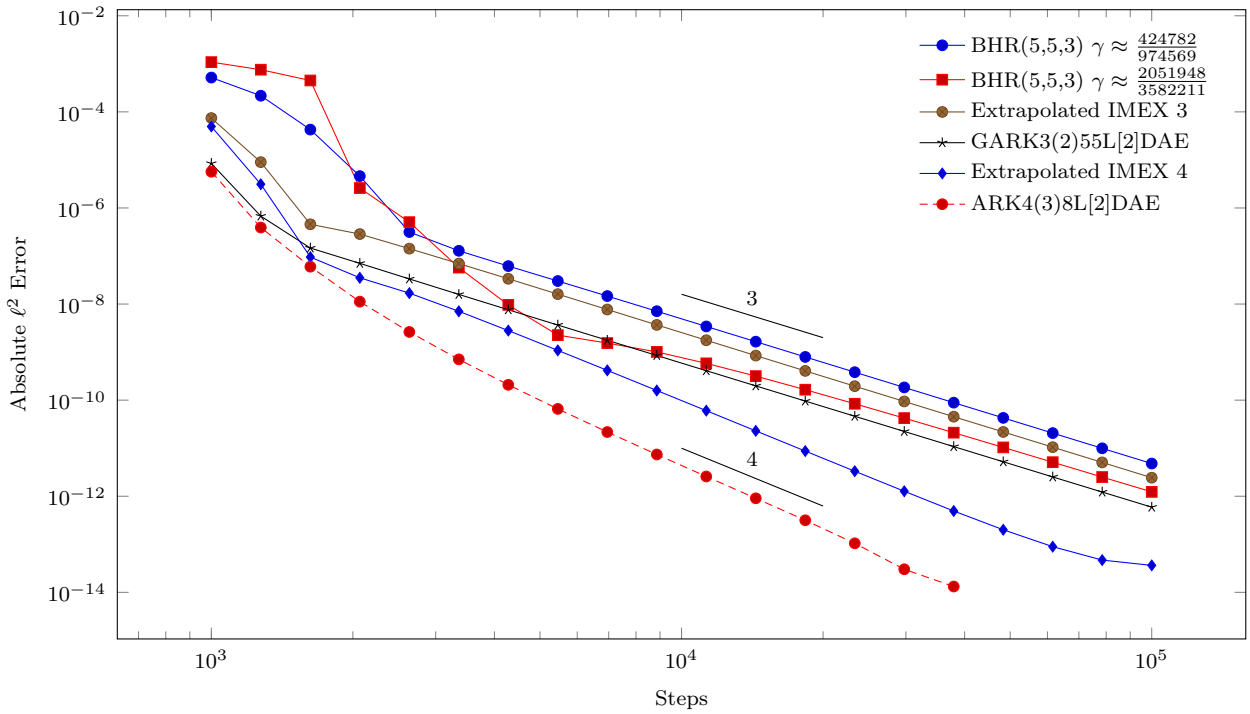


Figure 7.3: Convergence of IMEX methods on the ZLA-kinetics problem (7.38).

on the unit square domain $x, y \in (0, 1)$. The space-dependent diffusion is given by

$$D(x, y) = \frac{1}{10} \sum_{i=1}^{3} \exp\big(-100((x - 0.5)^2 + (y - y_i)^2)\big),$$

where $y_1 = 0.6$, $y_2 = 0.75$, and $y_3 = 0.9$. Over the timespan $[0, 7]$, zero Neumann boundary conditions are imposed on all four edges of the domain.

Using FEniCS [5], we apply a continuous finite element method to (7.39) with piecewise linear basis functions. The domain is discretized into a 100×100 grid with quadrilateral elements. This yields the ODE

$$My' = \underbrace{Ky}_{f^{\{I\}}(y)} + \underbrace{10(\mathbb{1} - y^2)(y + 0.6\mathbb{1})}_{f^{\{E\}}(y)} \in \mathbb{R}^{10201}, \tag{7.40}$$

where M and K are mass and stiffness matrices, respectively.

For this experiment, we will compare ARK3(2)4L[2]SA, GARK4(3)77L[2]SA, and ARK4(3)8L[2]DAE with several methods from the literature. This includes ASIRK-3A from [168], BHR(5,5,3) from [23], ARK3(2)4L[2]SA from [85], and ARK4(3)7L[2]SA$_1$ from [88]. First, fig. 7.4 provides the converge results over a range of eight different stepsizes. At order three, BHR(5,5,3) shows superconvergence for one value of γ, and all other methods converge at the theoretical order. GARK3(2)55L[2]DAE produces the smallest error for a fixed

number of steps. At order four, ARK4(3)7L[2]SA$_1$ and GARK4(3)77L[2]SA have similar results, but ARK4(3)8L[2]DAE proves to be the most accurate. Second, fig. 7.5 plots the timing results for the same experiment. Again, GARK3(2)55L[2]DAE leads the third order methods, and ARK4(3)8L[2]DAE leads the fourth order methods.

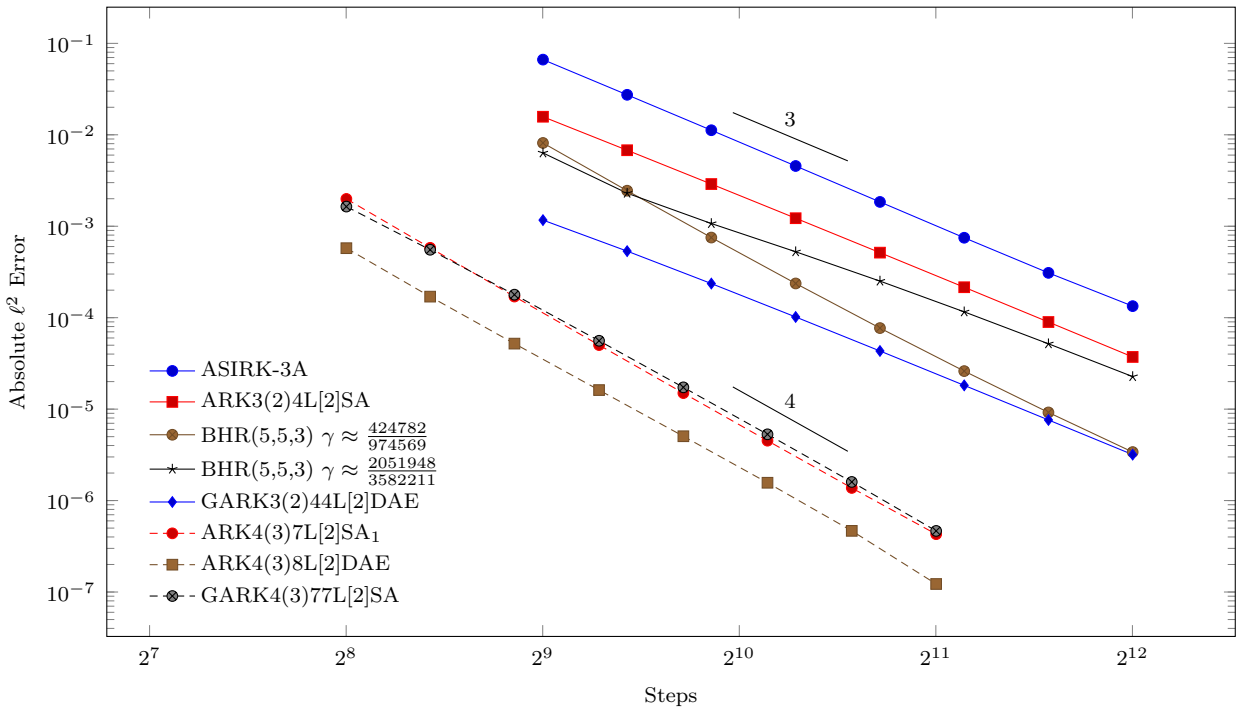


Figure 7.4: Convergence of IMEX methods on the BSVD problem (7.40).

7.8 Conclusion

This work has studied IMEX Runge–Kutta methods from the perspective of the GARK framework. Unlike the ARK and ASIRK frameworks that define an IMEX method solely by an explicit and implicit Runge–Kutta method, the GARK framework also includes coupling matrices $\mathbf{A}^{\{E,I\}}$ and $\mathbf{A}^{\{I,E\}}$. We have shown how these introduce new method structures, provide additional flexibility in the choice of base methods, and allow for further method optimizations.

The $\mathbb{M}1$ and $\mathbb{M}2$ classes have pros and cons associated with method derivation. The internal stability of $\mathbb{M}1$ methods is well-behaved, but C simplifying assumptions are not possible. The $\mathbb{M}2$ class requires additional conditions for internal stability and DAEs but has the FSAL property. At orders higher than four, $\mathbb{M}2$ may be the preferred class due to the feasibility of C simplifying assumption.

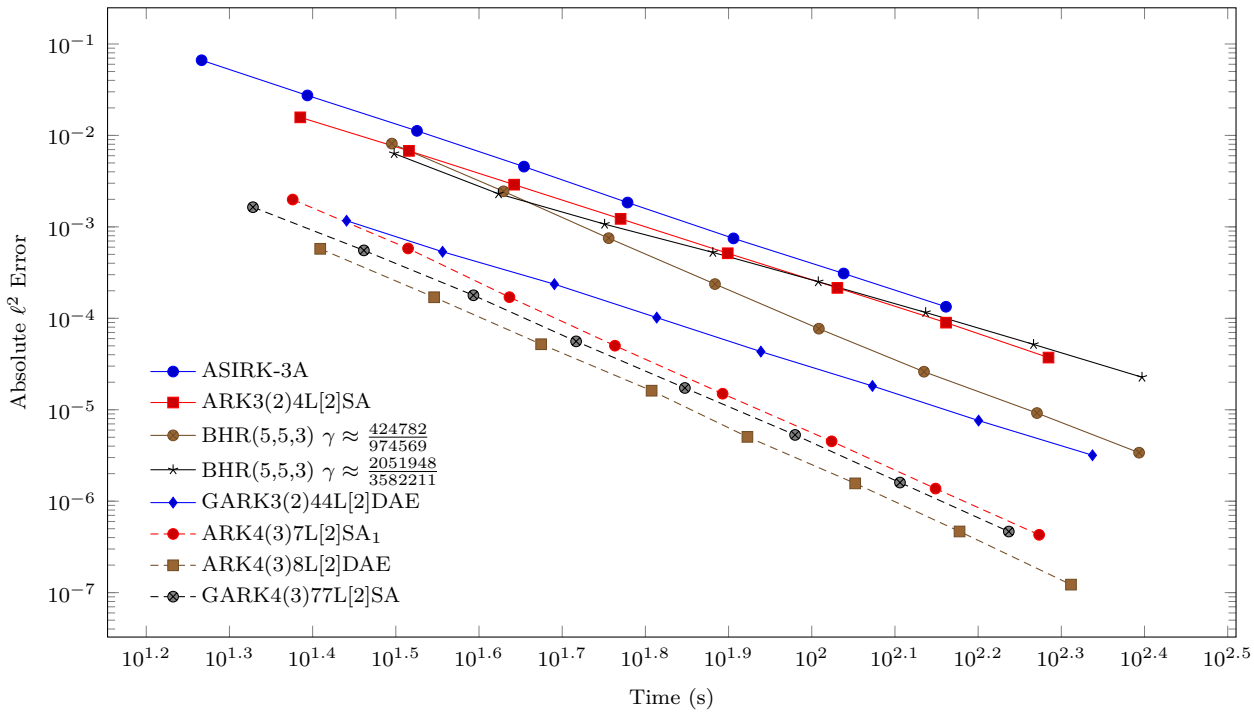


Figure 7.5: Performance of IMEX methods on the BSVD problem (7.40).

We have derived four new IMEX GARK methods, as well as one IMEX ARK method. One of the most important design criteria was minimizing the principal error to achieve better efficiency than existing IMEX methods of the same order. Indeed, the numerical experiments show improved efficiency for many of the methods. We do feel that additional improvements can be made to GARK4(3)77L[2]SA so that it is more competitive.

Future extensions of this work can include IMEX GARK methods for index-2 DAEs. This would provide further unification with half-explicit Runge–Kutta methods. Also of interest are IMEX GARK methods of order five and higher.

Chapter 8

Conclusion

A recurring theme found throughout this dissertation is that problematic terms in an ODE should be isolated and treated independently. The simplicity of traditional, monolithic time integration methods comes at the cost of efficiency. Time integration methods must be designed with large, multiscale, multiphysics problems in mind to best utilize modern computing resources. In this dissertation, we have discussed several new families of multimethods that are ideally-suited for these problems. Applications have included electric circuits, atmospheric and oceanographic flows, pattern-forming chemical reactions, and transient heat dynamics.

Several variants of multirate Runge–Kutta methods have been introduced including the implicit, discrete methods of chapter 2 and the implicit, infinitesimal methods of chapter 3. Between the two, multirate infinitesimal methods seem to offer the greatest potential for speedup by allowing any time integration method to propagate the fast dynamics. Both were analyzed in the GARK framework which consolidates and simplifies much of the order condition theory. We have seen several new results on linear stability and have shown that common assumptions like internal consistency or a decoupled structure can negatively impact stability. These multirate methods provided the foundation for the surrogate-accelerated time integration methods. In a principled way, Runge–Kutta methods can leverage developments in the booming areas of machine learning and model order reduction. Perhaps most interestingly, it demonstrates that multimethods are not limited to the traditional setting of multiphysics problems.

The second core topic we have covered in this dissertation is IMEX methods. Parallel IMEX GLMs have a coefficient structure that leads to a particularly simple form for the order conditions and linear stability regardless of the number of stages. This bypasses brute-force optimization approaches often used to derive (IMEX) GLMs. Parallel IMEX GLMs are an excellent choice for solving stiff problems when a high order of accuracy is required. For more modest orders, IMEX GARK methods are a promising alternative. We have seen examples of how to derive high-quality methods and some of the important design criteria to consider. Experiments demonstrate IMEX GARK methods can outperform existing IMEX methods based on more restrictive frameworks like ARK and ASIRK. To naturally and fully realize the coupling between an explicit and implicit Runge–Kutta method, the GARK formalism is crucial.

In addition, we have shown how the order reduction phenomenon can be addressed using

multimethods. Nonhomogeneity can be treated with a fully implicit Runge–Kutta, and the stiff linearity is left to be solved with any method a practitioner desires. In most practical cases, computational overheads with the GARK-based methods are negligible.

Extensions of the surrogate-acceleration idea to new families of multirate methods and new classes of problems will be a primary focus of future research. The completion of ODE Test Problems [119] will aid in the evaluation and testing of these methods. The Mathematica package used to derive the methods throughout this dissertation will continue to be developed and will be released publicly upon completion.

Bibliography

[1] Saul Abarbanel, David Gottlieb, and Mark H. Carpenter. On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 17(3):777–782, 1996. doi:10.1137/S1064827595282520.

[2] Assyr Abdulle, E Weinan, Björn Engquist, and Eric Vanden-Eijnden. The heterogeneous multiscale method. *Acta Numerica*, 21:1–87, 2012. doi:10.1017/S0962492912000025.

[3] Assyr Abdulle, Marcus J Grote, and Giacomo Rosilho de Souza. Explicit stabilized multirate method for stiff differential equations. *arXiv preprint arXiv:2006.00744*, 2020.

[4] Roger Alexander. Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977. doi:10.1137/0714068.

[5] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015. doi:10.11588/ans.2015.100.20553.

[6] Isaías Alonso-Mallo. Runge–Kutta methods without order reduction for linear initial boundary value problems. *Numerische Mathematik*, 91(4):577–603, 2002. doi:10.1007/s002110100332.

[7] Isaías Alonso-Mallo and Begoña Cano. Avoiding order reduction of Runge–Kutta discretizations for linear time-dependent parabolic problems. *BIT Numerical Mathematics*, 44(1):1–20, 2004. doi:10.1023/B:BITN.0000025087.83146.33.

[8] Robert Anderson, Julian Andrej, Andrew Barker, Jamie Bramwell, Jean-Sylvain Camier, Jakub Cerveny, Veselin Dobrev, Yohann Dudouit, Aaron Fisher, Tzanio Kolev, Will Pazner, Mark Stowell, Vladimir Tomov, Ido Akkerman, Johann Dahm, David Medina, and Stefano Zampini. MFEM: A modular finite element methods library. *Computers & Mathematics with Applications*, 81:42–74, 2021. doi:10.1016/j.camwa.2020.06.009. Development and Application of Open-source Software for Problems with Numerical PDEs.

[9] Jan Frederick Andrus. Numerical solution of systems of ordinary differential equations separated into subsystems. *SIAM Journal on Numerical Analysis*, 16(4):605–611, 1979. doi:10.1137/0716045.

[10] Jan Frederick Andrus. Stability of a multi-rate method for numerical integration of ODE's. *Computers & Mathematics with Applications*, 25(2):3–14, 1993. doi:10.1016/0898-1221(93)90218-K.

[11] Akio Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *Journal of Computational Physics*, 1(1):119–143, 1966. doi:10.1016/0021-9991(66)90015-5.

[12] Adérito Araújo, Ander Murua, and Jesús María Sanz-Serna. Symplectic methods based on decompositions. *SIAM Journal on Numerical Analysis*, 34(5):1926–1947, 1997. doi:10.1137/S0036142995292128.

[13] Uri M. Ascher, Steven J. Ruuth, and Brian T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995. doi:10.1137/0732037.

[14] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2):151 – 167, 1997. doi:10.1016/S0168-9274(97)00056-1. Special Issue on Time Integration.

[15] Randolph E Bank, William M Coughran, Wolfgang Fichtner, Eric H Grosse, Donald J Rose, and R Kent Smith. Transient simulation of silicon devices and circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 4(4): 436–451, 1985. doi:10.1109/TCAD.1985.1270142.

[16] Andreas Bartel and Michael Günther. A multirate W-method for electrical networks in state–space formulation. *Journal of Computational and Applied Mathematics*, 147 (2):411–425, 2002. doi:10.1016/S0377-0427(02)00476-4.

[17] Tobias Peter Bauer and Oswald Knoth. Extended multirate infinitesimal step methods: Derivation of order conditions. *Journal of Computational and Applied Mathematics*, 387:112541, 2021. doi:10.1016/j.cam.2019.112541. Numerical Solution of Differential and Differential-Algebraic Equations. Selected Papers from NUMDIFF-15.

[18] Richard M. Beam and Robert F. Warming. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, 22 (1):87–110, 1976. doi:10.1016/0021-9991(76)90110-8.

[19] Hisham bin Zubair. *Efficient multigrid methods based on improved coarse grid correction techniques*. PhD thesis, Delft University of Technology, Netherlands, 2009.

[20] Przemyslaw Bogacki and Lawrence F. Shampine. A 3(2) pair of Runge - Kutta formulas. *Applied Mathematics Letters*, 2(4):321–325, 1989. doi:10.1016/0893-9659(89)90079-7.

[21] Luca Bonaventura, Francesco Casella, L Delpopolo Carciopolo, and Akshay Ranade. A self adjusting multirate algorithm for robust time discretization of partial differential equations. *Computers & Mathematics with Applications*, 79(7):2086–2098, 2020. doi:10.1016/j.camwa.2019.11.023. Advanced Computational methods for PDEs.

[22] Sebastiano Boscarino. Error analysis of IMEX Runge–Kutta methods derived from differential-algebraic systems. *SIAM Journal on Numerical Analysis*, 45(4):1600–1621, 2007. doi:10.1137/060656929.

[23] Sebastiano Boscarino. On an accurate third order implicit-explicit Runge–Kutta method for stiff problems. *Applied Numerical Mathematics*, 59(7):1515–1528, 2009. ISSN 0168-9274. doi:10.1016/j.apnum.2008.10.003.

[24] Sebastiano Boscarino and Giovanni Russo. On a class of uniformly accurate IMEX Runge–Kutta schemes and applications to hyperbolic systems with relaxation. *SIAM Journal on Scientific Computing*, 31(3):1926–1945, 2009. doi:10.1137/080713562.

[25] Michał Braś, Giuseppe Izzo, and Zdzisław Jackiewicz. Accurate implicit–explicit general linear methods with inherent Runge–Kutta stability. *Journal of Scientific Computing*, 70(3):1105–1143, 2017. doi:10.1007/s10915-016-0273-y.

[26] Michał Braś, Angelamaria Cardone, Zdzisław Jackiewicz, and Paweł Pierzchała. Error propagation for implicit–explicit general linear methods. *Applied Numerical Mathematics*, 131:207–231, 2018. doi:10.1016/j.apnum.2018.05.004.

[27] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. doi:10.1017/9781108380690.

[28] Kevin Burrage, Willem Hundsdorfer, and Jan G. Verwer. A study of B-convergence of Runge–Kutta methods. *Computing*, 36(1):17–34, 1986. doi:10.1007/BF02238189.

[29] John C. Butcher. Diagonally-implicit multi-stage integration methods. *Applied Numerical Mathematics*, 11(5):347–363, 1993. doi:10.1016/0168-9274(93)90059-Z.

[30] John C. Butcher. General linear methods for the parallel solution of ordinary differential equations. In *Contributions in Numerical Mathematics*, pages 99–111. World Scientific, 1993. doi:10.1142/9789812798886_0008.

[31] John C. Butcher. Order and stability of parallel methods for stiff problems. *Advances in Computational Mathematics*, 7(1):79–96, 1997. doi:10.1023/A:1018934516771.

[32] John C. Butcher and Philippe Chartier. Parallel general linear methods for stiff ordinary differential and differential algebraic equations. *Applied Numerical Mathematics*, 17(3):213 – 222, 1995. doi:10.1016/0168-9274(95)00029-T. Special Issue on Numerical Methods for Ordinary Differential Equations.

[33] Giovanna Califano, Giuseppe Izzo, and Zdzisław Jackiewicz. Starting procedures for general linear methods. *Applied Numerical Mathematics*, 120:165–175, 2017. doi:10.1016/j.apnum.2017.05.009.

[34] Angelamaria Cardone, Zdzisław Jackiewicz, Adrian Sandu, and Hong Zhang. Extrapolated IMEX Runge–Kutta methods. *Mathematical Modelling and Analysis*, 19(2): 18–43, 2014. doi:10.3846/13926292.2014.892903.

[35] Angelamaria Cardone, Zdzisław Jackiewicz, Adrian Sandu, and Hong Zhang. Extrapolation-based implicit-explicit general linear methods. *Numerical Algorithms*, 65(3):377–399, 2014. doi:10.1007/s11075-013-9759-y.

[36] Angelamaria Cardone, Zdzisław Jackiewicz, Adrian Sandu, and Hong Zhang. Construction of highly stable implicit-explicit general linear methods. In *AIMS proceedings*, volume Dynamical Systems, Differential Equations, and Applications, Madrid, Spain, 2015. doi:10.3934/proc.2015.0185.

[37] Luis Chacón, Guangye Chen, Dana A Knoll, C Newman, H Park, William Taitano, Jeff A Willert, and Geoffrey Womeldorff. Multiscale high-order/low-order (HOLO) algorithms and applications. *Journal of Computational Physics*, 330:21–45, 2017. doi:10.1016/j.jcp.2016.10.069.

[38] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010. doi:10.1137/090766498.

[39] Rujeko Chinomona and Daniel R Reynolds. Implicit-explicit multirate infinitesimal GARK methods. *arXiv preprint arXiv:2007.09776*, 2020.

[40] Jeffrey M. Connors and Attou Miloua. Partitioned time discretization for parallel solution of coupled ODE systems. *BIT Numerical Mathematics*, 51(2):253–273, 2011. doi:10.1007/s10543-010-0295-z.

[41] Emil M. Constantinescu and Adrian Sandu. Multirate timestepping methods for hyperbolic conservation laws. *Journal of Scientific Computing*, 33(3):239–278, 2007. doi:10.1007/s10915-007-9151-y.

[42] Emil M. Constantinescu and Adrian Sandu. Extrapolated implicit-explicit time stepping. *SIAM Journal on Scientific Computing*, 31(6):4452–4477, 2010. doi:10.1137/080732833.

[43] Emil M. Constantinescu and Adrian Sandu. Extrapolated multirate methods for differential equations with multiple time scales. *Journal of Scientific Computing*, 56(1): 28–44, 2013. doi:10.1007/s10915-012-9662-z.

[44] G. J. Cooper and Ali Sayfy. Additive methods for the numerical solution of ordinary differential equations. *Mathematics of Computation*, 35(152):1159–1172, 1980. doi:10.2307/2006380.

[45] G. J. Cooper and Ali Sayfy. Additive Runge–Kutta methods for stiff ordinary differential equations. *Mathematics of Computation*, 40(161):207–207, 1983. doi:10.1090/s0025-5718-1983-0679441-1.

[46] Ludovica Delpopolo Carciopolo, Luca Bonaventura, Anna Scotti, and Luca Formaggia. A conservative implicit multirate method for hyperbolic problems. *Computational Geosciences*, 23(4):647–664, 2019. doi:10.1007/s10596-018-9764-2.

[47] Peter Deuflhard, Ernst Hairer, and J Zugck. One-step and extrapolation methods for differential-algebraic systems. *Numerische Mathematik*, 51(5):501–516, 1987. doi:10.1007/BF01400352.

[48] Adi Ditkowski, Sigal Gottlieb, and Zachary J Grant. IMEX error inhibiting schemes with post-processing. *arXiv preprint arXiv:1912.10027*, 2019.

[49] Jim Douglas, Jr. On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):42–65, 1955. doi:10.1137/0103004.

[50] Bjorn Engquist and Yen-Hsi Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Mathematics of Computation*, 74(252):1707–1742, 2005. doi:10.1090/S0025-5718-05-01745-X.

[51] Christian Engstler and Christian Lubich. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58(2):173–185, 1997. doi:10.1007/BF02684438.

[52] James Ferguson. A numerical solution for the barotropic vorticity equation forced by an equatorially trapped wave. Master's thesis, University of Victoria, 2008.

[53] Alexander I.J. Forrester and Andy J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009. doi:10.1016/j.paerosci.2008.11.001.

[54] Erich L. Foster, Traian Iliescu, and Zhu Wang. A finite element discretization of the streamfunction formulation of the stationary quasi-geostrophic equations of the ocean. *Computer Methods in Applied Mechanics and Engineering*, 261-262:105–117, 2013. doi:10.1016/j.cma.2013.04.008.

[55] Erich L. Foster, Traian Iliescu, and David R. Wells. A conforming finite element discretization of the streamfunction form of the unsteady quasi-geostrophic equations. *International Journal of Numerical Analysis & Modeling*, 13(6), 2016.

[56] Jason Frank, Willem Hundsdorfer, and Jan G. Verwer. On the stability of implicit-explicit linear multistep methods. *Applied Numerical Mathematics*, 25(2):193–205, 1997. doi:10.1016/S0168-9274(97)00059-7. Special Issue on Time Integration.

[57] Reinhard Frank, Josef Schneid, and Christoph W. Ueberhuber. The concept of B-convergence. *SIAM Journal on Numerical Analysis*, 18(5):753–780, 1981. doi:10.1137/0718051.

[58] David John Gagne, Hannah M. Christensen, Aneesh C. Subramanian, and Adam H. Monahan. Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz '96 model. *Journal of Advances in Modeling Earth Systems*, 12(3):e2019MS001896, 2020. doi:10.1029/2019MS001896.

[59] David J. Gardner, Jorge E. Guerra, François P. Hamon, Daniel R. Reynolds, Paul A. Ullrich, and Carol S. Woodward. Implicit–explicit (IMEX) Runge–Kutta methods for non-hydrostatic atmospheric models. *Geoscientific Model Development*, 11(4): 1497–1515, 2018. doi:10.5194/gmd-11-1497-2018.

[60] Charles W. Gear. Multirate methods for ordinary differential equations. Technical report, Illinois Univ., Urbana (USA). Dept. of Computer Science, 1974.

[61] Charles W. Gear and Daniel R. Wells. Multirate linear multistep methods. *BIT Numerical Mathematics*, 24(4):484–502, 1984. doi:10.1007/BF01934907.

[62] Severiano González-Pinto, Domingo Hernández-Abreu, Maria S Pérez-Rodríguez, Arash Sarshar, Steven Roberts, and Adrian Sandu. A unified formulation of splitting-based implicit time integration schemes. *arXiv preprint arXiv:2103.00757*, 2021.

[63] Eric Grimme. *Krylov projection methods for model reduction*. Theses, University of Illinois at Urbana Champaign, 1997.

[64] Michael Günther and Peter Rentrop. Multirate ROW methods and latency of electric circuits. *Applied Numerical Mathematics*, 13(1):83 – 102, 1993. doi:10.1016/0168-9274(93)90133-C.

[65] Michael Günther and Peter Rentrop. Partitioning and multirate strategies in latent electric circuits. In R. E. Bank, H. Gajewski, R. Bulirsch, and K. Merten, editors, *Mathematical Modelling and Simulation of Electrical Circuits and Semiconductor Devices*, pages 33–60. Birkhäuser Basel, 1994. doi:10.1007/978-3-0348-8528-7_3.

[66] Michael Günther and Adrian Sandu. Multirate generalized additive Runge–Kutta methods. *Numerische Mathematik*, 133(3):497–524, 2016. doi:10.1007/s00211-015-0756-z.

[67] Michael Günther, Anne Kværnø, and Peter Rentrop. Multirate partitioned Runge–Kutta methods. *BIT Numerical Mathematics*, 41(3):504–514, 2001. doi:10.1023/A:1021967112503.

[68] Michael Günther and Markus Hoschek. ROW methods adapted to electric circuit simulation packages. *Journal of Computational and Applied Mathematics*, 82(1):159–170, 1997. doi:10.1016/S0377-0427(97)00043-5. 7th ICCAM 96 Congress.

[69] Christoph Hachtel, Johanna Kerler-Back, Andreas Bartel, Michael Günther, and Tatjana Stykel. Multirate DAE/ODE-simulation and model order reduction for coupled field-circuit systems. In Ulrich Langer, Wolfgang Amrhein, and Walter Zulehner, editors, *Scientific Computing in Electrical Engineering*, pages 91–100, Cham, 2018. Springer International Publishing. ISBN 978-3-319-75538-0. doi:10.1007/978-3-319-75538-0_9.

[70] Christoph Hachtel, Andreas Bartel, Michael Günther, and Adrian Sandu. Multirate implicit Euler schemes for a class of differential-algebraic equations of index-1. *Journal of Computational and Applied Mathematics*, 387:112499, 2021. doi:10.1016/j.cam.2019.112499.

[71] Doan Duy Hai and Atsushi Yagi. Rosenbrock strong stability-preserving methods for convection–diffusion–reaction equations. *Japan Journal of Industrial and Applied Mathematics*, 31(2):401–417, 2014. doi:10.1007/s13160-014-0143-7.

[72] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Number 14 in Springer Series in Computational Mathematics. Springer-Verlag Berlin Heidelberg, 2 edition, 1996. doi:10.1007/978-3-642-05221-7.

[73] Ernst Hairer, Georg Bader, and Christian Lubich. On the stability of semi-implicit methods for ordinary differential equations. *BIT Numerical Mathematics*, 22(2): 211–232, 1982. doi:10.1007/BF01944478.

[74] Ernst Hairer, Gerhard Wanner, and Syvert P. Nørsett. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Number 8 in Springer Series in Computational Mathematics. Springer-Verlag Berlin Heidelberg, 2nd edition, 1993. doi:10.1007/978-3-540-78862-1.

[75] Ernst Hairer, Gerhard Wanner, and Christian Lubich. *Geometric Numerical Integration*. Springer-Verlag, 2006. doi:10.1007/3-540-30666-8.

[76] Wolfram Heineken and Gerald Warnecke. Partitioning methods for reaction–diffusion problems. *Applied Numerical Mathematics*, 56(7):981–1000, 2006. ISSN 0168-9274. doi:10.1016/j.apnum.2005.09.001.

[77] Inmaculada Higueras. Strong stability for additive Runge–Kutta methods. *SIAM Journal on Numerical Analysis*, 44(4):1735–1758, 2006. doi:10.1137/040612968.

[78] Inmaculada Higueras and Teo Roldán. Construction of additive semi-implicit Runge–Kutta methods with low-storage requirements. *Journal of Scientific Computing*, 67 (3):1019–1042, 2016. doi:10.1007/s10915-015-0116-2.

[79] Willem Hundsdorfer and Steven J. Ruuth. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *Journal of Computational Physics*, 225(2):2016–2042, 2007. doi:10.1016/j.jcp.2007.03.003.

[80] Willem Hundsdorfer and Valeriu Savcenco. Analysis of a multirate theta-method for stiff ODEs. *Applied Numerical Mathematics*, 59(3):693–706, 2009. doi:10.1016/j.apnum.2008.03.022. Selected Papers from NUMDIFF-11.

[81] Giuseppe Izzo and Zdzisław Jackiewicz. Transformed implicit-explicit DIMSIMs with strong stability preserving explicit part. *Numerical Algorithms*, 81(4):1343–1359, 2019. doi:10.1007/s11075-018-0647-3.

[82] Zdzisław Jackiewicz. *General linear methods for ordinary differential equations*. John Wiley and Sons, 2009. doi:10.1002/9780470522165.

[83] Zdzisław Jackiewicz and Hans Mittelmann. Construction of IMEX DIMSIMs of high order and stage order. *Applied Numerical Mathematics*, 121:234–248, 2017. doi:10.1016/j.apnum.2017.07.004.

[84] Toshiji Kato and Takeshi Kataoka. Circuit analysis by a new multirate method. *Electrical Engineering in Japan*, 126(4):55–62, 1999. doi:10.1002/(SICI)1520-6416(199903)126:4<55::AID-EEJ7>3.0.CO;2-G.

[85] Christopher A. Kennedy and Mark H. Carpenter. Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Applied Numerical Mathematics*, 44(1):139 – 181, 2003. doi:10.1016/S0168-9274(02)00138-1.

[86] Christopher A. Kennedy and Mark H. Carpenter. Diagonally implicit Runge–Kutta methods for ordinary differential equations. A review. Technical Report NASA/TM-2016-219173, NASA, 2016.

[87] Christopher A. Kennedy and Mark H. Carpenter. Diagonally implicit Runge–Kutta methods for stiff ODEs. *Applied Numerical Mathematics*, 146:221–244, 2019. doi:https://doi.org/10.1016/j.apnum.2019.07.008.

[88] Christopher A. Kennedy and Mark H. Carpenter. Higher-order additive Runge–Kutta schemes for ordinary differential equations. *Applied Numerical Mathematics*, 136: 183–205, 2019. doi:10.1016/j.apnum.2018.10.007.

[89] David I. Ketcheson, Benjamin Seibold, David Shirokoff, and Dong Zhou. DIRK schemes with high weak stage order. In Spencer J. Sherwin, David Moxey, Joaquim Peiró, Peter E. Vincent, and Christoph Schwab, editors, *Spectral and High Order*

Methods for Partial Differential Equations ICOSAHOM 2018, pages 453–463. Springer International Publishing, 2020. doi:10.1007/978-3-030-39647-3_36.

[90] Ioannis G. Kevrekidis, C. William Gear, and Gerhard Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal*, 50(7): 1346–1355, 2004. doi:10.1002/aic.10106.

[91] Oswald Knoth and Jörg Wensch. Generalized split-explicit Runge-–Kutta methods for the compressible Euler equations. *Monthly Weather Review*, 142(5):2067 – 2081, 2014. doi:10.1175/MWR-D-13-00068.1.

[92] Oswald Knoth and Ralf Wolke. Implicit-explicit Runge–Kutta methods for computing atmospheric reactive flows. *Applied Numerical Mathematics*, 28(2):327–341, 1998. doi:10.1016/S0168-9274(98)00051-8.

[93] Anne Kværnø. Stability of multirate Runge–Kutta schemes. *International Journal of Differential Equations and Applications*, 1(1):97–105, 2000.

[94] Anne Kværnø and Peter Rentrop. Low order multirate Runge–Kutta methods in electric circuit simulation, 1999.

[95] M. Paul Laiu, Eirik Endeve, Ran Chu, J. Austin Harris, and O. E. Bronson Messer. A DG-IMEX method for two-moment neutrino transport: Nonlinear solvers for neutrino–matter coupling. *The Astrophysical Journal Supplement Series*, 253(2):52, 2021. doi:10.3847/1538-4365/abe2a8.

[96] Jens Lang and Willem Hundsdorfer. Extrapolation-based implicit–explicit Peer methods with optimised stability regions. *Journal of Computational Physics*, 337:203–215, 2017. doi:10.1016/j.jcp.2017.02.034.

[97] Anders Logg. Multi-adaptive Galerkin methods for ODEs I. *SIAM Journal on Scientific Computing*, 24(6):1879–1902, 2003. doi:10.1137/S1064827501389722.

[98] Edward N. Lorenz. Predictability: A problem partly solved. In *Seminar on Predictability, 4-8 September 1995*, volume 1, pages 1–18, Shinfield Park, Reading, 1995. ECMWF, ECMWF.

[99] Changhong Mou, Honghu Liu, David R. Wells, and Traian Iliescu. Data-driven correction reduced order models for the quasi-geostrophic equations: A numerical investigation. *International Journal of Computational Fluid Dynamics*, 34(2):147–159, 2020. doi:10.1080/10618562.2020.1723556.

[100] Mahesh Narayanamurthi, Paul Tranquilli, Adrian Sandu, and Mayya Tokman. EPIRK-W and EPIRK-K time discretization methods. *Journal of Scientific Computing*, 78 (1):167–201, 2019. doi:10.1007/s10915-018-0761-3.

[101] Syvert P Nørsett. Semi-explicit Runge–Kutta methods. Technical Report 6/74, Department of Mathematics, University of Trondheim, 1974.

[102] Yew S. Ong, Prasanth B. Nair, and Andrew J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4): 687–696, 2003. doi:10.2514/2.1999.

[103] Alexander Ostermann and Michel Roche. Runge–Kutta methods for partial differential equations and fractional orders of convergence. *Mathematics of Computation*, 59(200): 403–403, 1992. doi:10.1090/s0025-5718-1992-1142285-6.

[104] Alexander Ostermann and Michel Roche. Rosenbrock methods for partial differential equations and fractional orders of convergence. *SIAM Journal on Numerical Analysis*, 30(4):1084–1098, 1993. doi:10.1137/0730056.

[105] Lorenzo Pareschi and Giovanni Russo. Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific Computing*, 25(1):129–155, 2005. doi:10.1007/s10915-004-4636-4.

[106] D. Pathria. The correct formulation of intermediate boundary conditions for Runge–Kutta time integration of initial boundary value problems. *SIAM Journal on Scientific Computing*, 18(5):1255–1266, 1997. doi:10.1137/S1064827594273948.

[107] Donald W. Peaceman and Henry H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):28–41, 1955. doi:10.2307/2098834.

[108] John E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993. doi:10.1126/science.261.5118.189.

[109] Janet S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):777–786, 1989. doi:10.1137/0910047.

[110] Andrey A. Popov, Changhong Mou, Adrian Sandu, and Traian Iliescu. A multifidelity ensemble Kalman filter with reduced order control variates. *SIAM Journal on Scientific Computing*, 43(2):A1134–A1162, 2021. doi:10.1137/20M1349965.

[111] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Mathematics of Computation*, 28(125):145–162, 1974. doi:10.1090/S0025-5718-1974-0331793-2.

[112] Nestor V. Queipo, Raphael T. Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P. Kevin Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005. doi:10.1016/j.paerosci.2005.02.001.

[113] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378: 686–707, 2019. doi:10.1016/j.jcp.2018.10.045.

[114] Anthony Ralston. Runge–Kutta methods with minimum error bounds. *Mathematics of Computation*, 16(80):431–437, 1962. doi:10.2307/2003133.

[115] Joachim Rang. An analysis of the Prothero–Robinson example for constructing new DIRK and ROW methods. *Journal of Computational and Applied Mathematics*, 262: 105–114, 2014. doi:10.1016/j.cam.2013.09.062. Selected Papers from NUMDIFF-13.

[116] Joachim Rang. The Prothero and Robinson example: Convergence studies for Runge–Kutta and Rosenbrock–Wanner methods. *Applied Numerical Mathematics*, 108:37–56, 2016. ISSN 0168-9274. doi:10.1016/j.apnum.2016.04.012.

[117] Stephan Rasp. Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: General algorithms and Lorenz 96 case study (v1.0). *Geoscientific Model Development*, 13(5):2185–2196, 2020. doi:10.5194/gmd-13-2185-2020.

[118] John R. Rice. Split Runge–Kutta methods for simultaneous equations. *Journal of Research of the National Institute of Standards and Technology*, 60(B), 1960. doi:10.6028/jres.064b.018.

[119] Steven Roberts, Andrey A Popov, and Adrian Sandu. ODE test problems: a MATLAB suite of initial value problems. *arXiv preprint arXiv:1901.04098*, 2019.

[120] Steven Roberts, Andrey A Popov, Arash Sarshar, and Adrian Sandu. A fast time-stepping strategy for dynamical systems equipped with a surrogate model. *arXiv preprint arXiv:2011.03688*, 2020.

[121] Steven Roberts, Arash Sarshar, and Adrian Sandu. Coupled multirate infinitesimal GARK schemes for stiff systems with multiple time scales. *SIAM Journal on Scientific Computing*, 42(3):A1609–A1638, 2020. doi:10.1137/19M1266952.

[122] Steven Roberts, Arash Sarshar, and Adrian Sandu. Parallel implicit-explicit general linear methods. *Communications on Applied Mathematics and Computation*, 2020. doi:10.1007/s42967-020-00083-5.

[123] Steven Roberts, John Loffeld, Arash Sarshar, Carol S. Woodward, and Adrian Sandu. Implicit multirate GARK methods. *Journal of Scientific Computing*, 87(1):4, 2021. doi:10.1007/s10915-020-01400-z.

[124] Michel Roche. Implicit Runge–Kutta methods for differential algebraic equations. *SIAM Journal on Numerical Analysis*, 26(4):963–975, 1989. doi:10.1137/0726053.

[125] Gustavo Rodríguez-Gómez, Pedro González-Casanova, and Jorge Martínez-Carballido. Computing general companion matrices and stability regions of multirate methods. *International Journal for Numerical Methods in Engineering*, 61(2):255–273, 2004. doi:10.1002/nme.1065.

[126] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4), 2017. doi:10.1126/sciadv.1602614.

[127] Omer San and Traian Iliescu. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Advances in Computational Mathematics*, 41(5):1289–1319, 2015. doi:10.1007/s10444-015-9417-0.

[128] Jørgen Sand and Stig Skelboe. Stability of backward Euler multirate methods and convergence of waveform relaxation. *BIT Numerical Mathematics*, 32(2):350–366, 1992. doi:10.1007/BF01994887.

[129] Adrian Sandu. A class of multirate infinitesimal GARK methods. *SIAM Journal on Numerical Analysis*, 57(5):2300–2327, 2019. doi:10.1137/18M1205492.

[130] Adrian Sandu and Emil M. Constantinescu. Multirate explicit Adams methods for time integration of conservation laws. *Journal of Scientific Computing*, 38(2):229–249, 2009. doi:10.1007/s10915-008-9235-3.

[131] Adrian Sandu and Michael Günther. A class of generalized additive Runge–Kutta methods. *arXiv preprint arXiv:1310.5573*, 2013.

[132] Adrian Sandu and Michael Günther. A generalized-structure approach to additive Runge–Kutta methods. *SIAM Journal on Numerical Analysis*, 53(1):17–42, 2015. doi:10.1137/130943224.

[133] Adrian Sandu, Michael Günther, and Steven Roberts. Linearly implicit GARK schemes. *Applied Numerical Mathematics*, 161:286–310, 2021. doi:10.1016/j.apnum.2020.11.014.

[134] Jesús María Sanz-Serna, Jan G. Verwer, and Willem Hundsdorfer. Convergence and order reduction of Runge–Kutta schemes applied to evolutionary problems in partial differential equations. *Numerische Mathematik*, 50(4):405–418, 1986. doi:10.1007/BF01396661.

[135] Arash Sarshar, Steven Roberts, and Adrian Sandu. Design of high-order decoupled multirate GARK schemes. *SIAM Journal on Scientific Computing*, 41(2):A816–A847, 2019. doi:10.1137/18M1182875.

[136] Arash Sarshar, Steven Roberts, and Adrian Sandu. Alternating directions implicit integration in a general linear method framework. *Journal of Computational and Applied Mathematics*, 387:112619, 2021. doi:10.1016/j.cam.2019.112619. Numerical Solution of Differential and Differential-Algebraic Equations. Selected Papers from NUMDIFF-15.

[137] Valeriu Savcenco. Comparison of the asymptotic stability properties for two multirate strategies. *Journal of Computational and Applied Mathematics*, 220(1):508–524, 2008. doi:10.1016/j.cam.2007.09.005.

[138] Valeriu Savcenco. Construction of a multirate RODAS method for stiff ODEs. *Journal of Computational and Applied Mathematics*, 225(2):323–337, 2009. doi:10.1016/j.cam.2008.07.041.

[139] Valeriu Savcenco, Willem Hundsdorfer, and Jan G. Verwer. A multirate time stepping strategy for stiff ordinary differential equations. *BIT Numerical Mathematics*, 47(1): 137–155, 2007. doi:10.1007/s10543-006-0095-7.

[140] Volker Scheidemann. *Introduction to complex analysis in several variables*. Birkhäuser Basel, 2005. doi:10.1007/3-7643-7491-8.

[141] Martin Schlegel, Oswald Knoth, Martin Arnold, and Ralf Wolke. Multirate Runge–Kutta schemes for advection equations. *Journal of Computational and Applied Mathematics*, 226(2):345–357, 2009. doi:10.1016/j.cam.2008.08.009. Special Issue: Large scale scientific computations.

[142] Martin Schlegel, Oswald Knoth, Martin Arnold, and Ralf Wolke. Multirate implicit-explicit time integration schemes in atmospheric modelling. In *AIP Conference Proceedings*, volume 1281, pages 1831–1834. International Conference of Numerical Analysis and Applied Mathematics, 2010. doi:10.1063/1.3498252.

[143] Martin Schlegel, Oswald Knoth, Martin Arnold, and Ralf Wolke. Implementation of splitting methods for air pollution modeling. *Geoscientific Model Development Discussions*, 4(4):2937–2972, 2011. doi:10.5194/gmd-5-1395-2012.

[144] Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi:10.1017/S0022112010001217.

[145] Moritz Schneider, Jens Lang, and Willem Hundsdorfer. Extrapolation-based superconvergent implicit-explicit Peer methods with A-stable implicit part. *Journal of Computational Physics*, 367:121–133, 2018. doi:10.1016/j.jcp.2018.04.006.

[146] Jean M. Sexton and Daniel R. Reynolds. Relaxed multirate infinitesimal step methods for initial-value problems. *arXiv preprint arXiv:1808.03718*, 2019.

[147] Lawrence Sirovich. Turbulence and the dynamics of coherent structures part I: Coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[148] Stig Skelboe and Per Ulfkjaer Andersen. Stability properties of backward Euler multirate formulas. *SIAM Journal on Scientific and Statistical Computing*, 10(5):1000–1009, 1989. doi:10.1137/0910059.

[149] Behnam Soleimani and Rüdiger Weiner. Superconvergent IMEX peer methods. *Applied Numerical Mathematics*, 130:70–85, 2018. doi:10.1016/j.apnum.2018.03.014.

[150] Raymond J. Spiteri and Ryan C. Dean. On the performance of an implicit–explicit Runge–Kutta method in models of cardiac electrical activity. *IEEE Transactions on Biomedical Engineering*, 55(5):1488–1495, 2008. doi:10.1109/TBME.2007.914677.

[151] Walter Johannes Henricus Stortelder. *Parameter estimation in nonlinear dynamical systems*. PhD thesis, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, 1998.

[152] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. doi:10.1137/0705041.

[153] Gregory Mark Tanner. *Generalized additive Runge–Kutta methods for stiff odes*. PhD thesis, The University of Iowa, 2018. URL `https://ir.uiowa.edu/etd/6507`.

[154] Gilles Tissot, Laurent Cordier, Nicolas Benard, and Bernd R. Noack. Model reduction using dynamic mode decomposition. *Comptes Rendus Mécanique*, 342(6):410–416, 2014. doi:10.1016/j.crme.2013.12.011.

[155] Mayya Tokman. A new class of exponential propagation iterative methods of Runge–Kutta type (EPIRK). *Journal of Computational Physics*, 230(24):8762–8778, 2011. doi:10.1016/j.jcp.2011.08.023.

[156] Eric Vanden-Eijnden. On HMM-like integrators and projective integration methods for systems with multiple time scales. *Communications in Mathematical Sciences*, 5 (2):495 – 505, 2007. doi:10.4310/CMS.2007.v5.n2.a14.

[157] Arie Verhoeven, Theo G. J. Beelen, Ahmed El Guennouni, E. Jan W. Ter Maten, and Robert M.M. Mattheij. Error analysis of BDF compound-fast multirate method for differential-algebraic equations. Technical Report CASA-report 06-10, Technische Universiteit Eindhoven, 2006.

[158] Arie Verhoeven, Ahmed El Guennouni, E. Jan W. Ter Maten, and Robert M.M. Mattheij. A general compound multirate method for circuit simulation problems. In Angelo Marcello Anile, Giuseppe Alì, and Giovanni Mascali, editors, *Scientific Computing in Electrical Engineering*, pages 143–149. Springer Berlin Heidelberg, 2006.

[159] Arie Verhoeven, E. Jan W. Ter Maten, Robert M.M. Mattheij, and Bratislav Tasić. Stability analysis of the BDF slowest-first multirate methods. *International Journal of Computer Mathematics*, 84(6):895–923, 2007. doi:10.1080/00207160701458641.

[160] Jan G. Verwer. Convergence and order reduction of diagonally implicit Runge–Kutta schemes in the method of lines. *Numerical Analysis*, 140:220–237, 1986.

[161] Jörg Wensch, Oswald Knoth, and Alexander Galant. Multirate infinitesimal step methods for atmospheric flow simulation. *BIT Numerical Mathematics*, 49(2):449–473, 2009. doi:10.1007/s10543-009-0222-3.

[162] Antonella Zanna. Discrete variational methods and symplectic generalized additive Runge–Kutta methods. *arXiv preprint arXiv:2001.07185*, 2020.

[163] Hong Zhang and Adrian Sandu. A second-order diagonally-implicit-explicit multi-stage integration method. In *Proceedings of the International Conference on Computational Science ICCS 2012*, volume 9, pages 1039–1046, 2012. doi:10.1016/j.procs.2012.04.112.

[164] Hong Zhang, Adrian Sandu, and Sébastien Blaise. Partitioned and implicit–explicit general linear methods for ordinary differential equations. *Journal of Scientific Computing*, 61(1):119–144, 2014. doi:10.1007/s10915-014-9819-z.

[165] Hong Zhang, Adrian Sandu, and Sébastien Blaise. High order implicit-explicit general linear methods with optimized stability regions. *SIAM Journal on Scientific Computing*, 38(3):A1430–A1453, 2016. doi:10.1137/15M1018897.

[166] Danyong Zhao, Yijing Li, and Jernej Barbič. Asynchronous implicit backward Euler integration. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, pages 1–9, Goslar Germany, Germany, 2016. Eurographics Association. doi:10.2312/sca.20161217.

[167] Evgeniy Zharovsky, Adrian Sandu, and Hong Zhang. A class of implicit-explicit two-step Runge–Kutta methods. *SIAM Journal on Numerical Analysis*, 53(1):321–341, 2015. doi:10.1137/130937883.

[168] Xiaolin Zhong. Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows. *Journal of Computational Physics*, 128(1):19–31, 1996. doi:10.1006/jcph.1996.0193.

Appendices

Appendix A

New Compound-Fast MrGARKs

A.1 Third Order Compound-Fast MrGARK

A third order compound-fast method, which we will refer to as compound-fast MrGARK SDIRK3, is built on the following base method of Alexander [4]:

$$
\begin{array}{c|ccc}
\gamma & \gamma & 0 & 0 \\
\frac{\gamma}{2}+\frac{1}{2} & \frac{1}{2}-\frac{\gamma}{2} & \gamma & 0 \\
1 & -\frac{3\gamma^2}{2}+4\gamma-\frac{1}{4} & \frac{3\gamma^2}{2}-5\gamma+\frac{5}{4} & \gamma \\
\hline
 & -\frac{3\gamma^2}{2}+4\gamma-\frac{1}{4} & \frac{3\gamma^2}{2}-5\gamma+\frac{5}{4} & \gamma \\
\hline
 & -\frac{3\gamma^2}{2}+3\gamma-\frac{1}{4} & \frac{3\gamma^2}{2}-3\gamma+\frac{5}{4} & 0
\end{array}
\tag{A.1}
$$

Here, $\gamma \approx 0.44$ is the middle root of $6\gamma^3 - 18\gamma^2 + 9\gamma - 1 = 0$. The coupling coefficients are

$$
\begin{aligned}
a_{1,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{(6\gamma^2 - 24\gamma + 5)\,(2\gamma^2 + 2\gamma(\lambda - 1) + (\lambda - 1)^2) - 8\gamma^3 M^2}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2} \\
&\quad - \frac{(6\gamma^3 - 30\gamma^2 - 15\gamma + 5)\,M(\gamma + \lambda - 1)}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{1,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-5(\lambda - 1)^2 + 12\gamma^4(M - 1) + 4\gamma^3(M - 1)(3\lambda + 4M - 17)}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2} \\
&\quad + \frac{\gamma^2\,(-6\lambda^2 + 68\lambda + (82 - 72\lambda)M - 72) + 2\gamma(\lambda - 1)(14\lambda + 5M - 19)}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{1,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{4\gamma\,((\lambda - 1)^2 + 2\gamma^2(M - 1)^2 - 2\gamma(\lambda - 1)(2M - 1))}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{2,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{-2\,(6\gamma^2 - 24\gamma + 5)\,(\gamma(\lambda + 1) + (\lambda - 1)\lambda) + 16\gamma^3 M^2}{2(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2} \\
&\quad - \frac{(6\gamma^3 - 30\gamma^2 - 15\gamma + 5)\,M(\gamma + 2\lambda - 1)}{2(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{2,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{\gamma\,(28\lambda^2 - 33\lambda + 5(2\lambda - 1)M - 5) + 2\gamma^3\,(8M^2 - 3(\lambda + 1) + 3(2\lambda - 7)M)}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2} \\
&\quad + \frac{6\gamma^4 M - 5(\lambda - 1)\lambda + \gamma^2\,(-6\lambda^2 + 34\lambda + (41 - 72\lambda)M + 28)}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{2,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{4\gamma\,((\lambda - 1)\lambda + 2\gamma^2(M - 1)M + \gamma(\lambda + (2 - 4\lambda)M + 1))}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{3,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-36\gamma^5 + 252\gamma^4 + 20\lambda^2 - 4\gamma^3\,(8M^2 + 6\lambda M + 129)}{4(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2} \\
&\quad + \frac{24\gamma^2\,(\lambda^2 + 5\lambda M + 13) + \gamma\,(-96\lambda^2 + 60\lambda M - 69) - 20\lambda M + 5}{4(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{3,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{36\gamma^5 - 276\gamma^4 - 5\,(4\lambda^2 + 1) + \gamma^3\,(64M^2 + 48\lambda M + 588)}{4(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2} \\
&\quad + \frac{-12\gamma^2\,(2\lambda^2 + 24\lambda M + 29) + \gamma\,(112\lambda^2 + 40\lambda M + 73)}{4(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}, \\
a_{3,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{\gamma\,(6\gamma^3 - 4\lambda^2 - 2\gamma^2\,(4M^2 + 9) + \gamma(16\lambda M + 9) - 1)}{(\gamma - 1)\,(6\gamma^2 - 20\gamma + 5)\,M^2}.
\end{aligned}
\tag{A.2}
$$

A.2 Fourth Order Compound-Fast MrGARK

For the compound-fast method of order four, we start with a new base method, solving the coupling and base conditions together. This allows more flexibility to keep the coupling

coefficients bounded functions of λ and M. The following L-stable base method was derived:

$$
\begin{array}{c|ccccc}
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\
1 & \frac{3}{4} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{2}{5} & \frac{69}{400} & -\frac{9}{400} & \frac{1}{4} & 0 & 0 \\
\frac{7}{11} & \frac{103241}{143748} & -\frac{1751}{71874} & -\frac{11050}{35937} & \frac{1}{4} & 0 \\
1 & \frac{400}{459} & -\frac{35}{216} & -\frac{250}{351} & \frac{1331}{1768} & \frac{1}{4} \\
\hline
 & \frac{400}{459} & -\frac{35}{216} & -\frac{250}{351} & \frac{1331}{1768} & \frac{1}{4} \\
\hline
 & \frac{10388}{10557} & -\frac{1399}{4968} & -\frac{30425}{32292} & \frac{73205}{81328} & \frac{125}{368}
\end{array} . \tag{A.3}
$$

When paired with the following coupling coefficients, we have the compound-fast MrGARK SDIRK4 scheme:

$$
\begin{aligned}
a_{1,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-165\left(64\lambda^4-192\lambda^3+240\lambda^2-148\lambda+37\right)+2688(4\lambda-3)M^3-3408\left(8\lambda^2-12\lambda+5\right)M^2+448\left(64\lambda^3-144\lambda^2+120\lambda-37\right)M}{1836M^4},\\
a_{1,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{1110\left(64\lambda^4-192\lambda^3+240\lambda^2-148\lambda+37\right)-1296M^4-2292(4\lambda-3)M^3+8781\left(8\lambda^2-12\lambda+5\right)M^2-2101\left(64\lambda^3-144\lambda^2+120\lambda-37\right)M}{22464M^4},\\
a_{1,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{125\left(-33\left(64\lambda^4-192\lambda^3+240\lambda^2-148\lambda+37\right)+336(4\lambda-3)M^3-552\left(8\lambda^2-12\lambda+5\right)M^2+83\left(64\lambda^3-144\lambda^2+120\lambda-37\right)M\right)}{22464M^4},\\
a_{1,4}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{1331\left(-5\left(64\lambda^4-192\lambda^3+240\lambda^2-148\lambda+37\right)+32(4\lambda-3)M^3-60\left(8\lambda^2-12\lambda+5\right)M^2+11\left(64\lambda^3-144\lambda^2+120\lambda-37\right)M\right)}{56576M^4},\\
a_{1,5}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-85\left(64\lambda^4-192\lambda^3+240\lambda^2-148\lambda+37\right)+192M^4+16(4\lambda-3)M^3-648\left(8\lambda^2-12\lambda+5\right)M^2+175\left(64\lambda^3-144\lambda^2+120\lambda-37\right)M}{3328M^4},\\
a_{2,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-165\left(64\lambda^4-48\lambda^2+68\lambda-17\right)+10752\lambda M^3-3408\left(8\lambda^2-1\right)M^2+448\left(64\lambda^3-24\lambda+17\right)M}{1836M^4},\\
a_{2,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{1110\left(64\lambda^4-48\lambda^2+68\lambda-17\right)-1296M^4-9168\lambda M^3+8781\left(8\lambda^2-1\right)M^2-2101\left(64\lambda^3-24\lambda+17\right)M}{22464M^4},\\
a_{2,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{125\left(-33\left(64\lambda^4-48\lambda^2+68\lambda-17\right)+1344\lambda M^3-552\left(8\lambda^2-1\right)M^2+83\left(64\lambda^3-24\lambda+17\right)M\right)}{22464M^4},\\
a_{2,4}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{1331\left(5\left(-64\lambda^4+48\lambda^2-68\lambda+17\right)+128\lambda M^3-60\left(8\lambda^2-1\right)M^2+11\left(64\lambda^3-24\lambda+17\right)M\right)}{56576M^4},\\
a_{2,5}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-85\left(64\lambda^4-48\lambda^2+68\lambda-17\right)+192M^4+64\lambda M^3-648\left(8\lambda^2-1\right)M^2+175\left(64\lambda^3-24\lambda+17\right)M}{3328M^4},\\
a_{3,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-33\left(32000\lambda^4-76800\lambda^3+84720\lambda^2-56180\lambda+15773\right)+215040(5\lambda-3)M^3}{183600M^4}\\
&\quad+\frac{-3408\left(800\lambda^2-960\lambda+353\right)M^2+448\left(6400\lambda^3-11520\lambda^2+8472\lambda-2809\right)M}{183600M^4},\\
a_{3,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{222\left(32000\lambda^4-76800\lambda^3+84720\lambda^2-56180\lambda+15773\right)-129600M^4-183360(5\lambda-3)M^3}{2246400M^4}\\
&\quad+\frac{8781\left(800\lambda^2-960\lambda+353\right)M^2-2101\left(6400\lambda^3-11520\lambda^2+8472\lambda-2809\right)M}{2246400M^4},\\
a_{3,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{33\left(32000\lambda^4-76800\lambda^3+84720\lambda^2-56180\lambda+15773\right)-134400(5\lambda-3)M^3}{89856M^4}\\
&\quad+\frac{2760\left(800\lambda^2-960\lambda+353\right)M^2-415\left(6400\lambda^3-11520\lambda^2+8472\lambda-2809\right)M}{89856M^4},\\
a_{3,4}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{1331\left(-32000\lambda^4+76800\lambda^3-84720\lambda^2+56180\lambda+2560(5\lambda-3)M^3\right)}{5657600M^4}\\
&\quad+\frac{1331\left(-60\left(800\lambda^2-960\lambda+353\right)M^2+11\left(6400\lambda^3-11520\lambda^2+8472\lambda-2809\right)M-15773\right)}{5657600M^4},\\
a_{3,5}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-17\left(32000\lambda^4-76800\lambda^3+84720\lambda^2-56180\lambda+15773\right)+19200M^4+1280(5\lambda-3)M^3}{332800M^4}\\
&\quad+\frac{-648\left(800\lambda^2-960\lambda+353\right)M^2+175\left(6400\lambda^3-11520\lambda^2+8472\lambda-2809\right)M}{332800M^4},\\
a_{4,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-33\left(425920\lambda^4-619520\lambda^3+280720\lambda^2-35660\lambda+2387\right)+1300992(11\lambda-4)M^3}{2443716M^4}\\
&\quad+\frac{-137456\left(264\lambda^2-192\lambda+29\right)M^2+448\left(85184\lambda^3-92928\lambda^2+28072\lambda-1783\right)M}{2443716M^4},\\
a_{4,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{222\left(425920\lambda^4-619520\lambda^3+280720\lambda^2-35660\lambda+2387\right)-1724976M^4-1109328(11\lambda-4)M^3}{29899584M^4}\\
&\quad+\frac{354167\left(264\lambda^2-192\lambda+29\right)M^2-2101\left(85184\lambda^3-92928\lambda^2+28072\lambda-1783\right)M}{29899584M^4},\\
a_{4,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{25\left(-33\left(425920\lambda^4-619520\lambda^3+280720\lambda^2-35660\lambda+2387\right)+813120(11\lambda-4)M^3\right)}{29899584M^4}\\
&\quad-\frac{25\left(-111320\left(264\lambda^2-192\lambda+29\right)M^2+415\left(85184\lambda^3-92928\lambda^2+28072\lambda-1783\right)M\right)}{29899584M^4},\\
a_{4,4}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-425920\lambda^4+619520\lambda^3-280720\lambda^2+35660\lambda+15488(11\lambda-4)M^3}{56576M^4}\\
&\quad+\frac{-2420\left(264\lambda^2-192\lambda+29\right)M^2+11\left(85184\lambda^3-92928\lambda^2+28072\lambda-1783\right)M-2387}{56576M^4},\\
a_{4,5}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-17\left(425920\lambda^4-619520\lambda^3+280720\lambda^2-35660\lambda+2387\right)+255552M^4+7744(11\lambda-4)M^3}{4429568M^4}\\
&\quad+\frac{-26136\left(264\lambda^2-192\lambda+29\right)M^2+175\left(85184\lambda^3-92928\lambda^2+28072\lambda-1783\right)M}{4429568M^4},\\
a_{5,1}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{16\lambda\left(-165\lambda^3+168M^3-426\lambda M^2+448\lambda^2M\right)}{459M^4},\\
a_{5,2}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= -\frac{-8880\lambda^4+162M^4+1146\lambda M^3-8781\lambda^2M^2+16808\lambda^3M}{2808M^4},\\
a_{5,3}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{125\lambda\left(33\lambda^3-21M^3+69\lambda M^2-83\lambda^2M\right)}{351M^4},\\
a_{5,4}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{1331\lambda\left(-10\lambda^3+4M^3-15\lambda M^2+22\lambda^2M\right)}{1768M^4},\\
a_{5,5}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} &= \frac{-85\lambda^4+3M^4+\lambda M^3-81\lambda^2M^2+175\lambda^3M}{52M^4}.
\end{aligned}
\tag{A.4}
$$

Appendix B

Conservation of Linear Invariants for GARK methods

For some applications, it is desirable that a numerical integrator uphold invariant properties of the system, such as conservation of mass and energy. These invariants are generally expressed as *first integrals* of the system. It is well known that non-partitioned explicit and implicit Runge–Kutta methods conserve linear first integrals but must meet certain restrictions to conserve quadratic ones, e.g. as with symplectic methods. Partitioned Runge–Kutta methods must obey additional restrictions to conserve even linear invariants. A detailed discussion about preservation of first integrals by Runge–Kutta methods can be found in [75].

GARK schemes are partitioned methods, and here we briefly describe conditions for them to uphold linear first integrals. Consider an ODE (2.1) satisfying the following linear invariant:

$$\zeta^T \left(f^{\{\mathfrak{f}\}}(y) + f^{\{\mathfrak{s}\}}(y) \right) = 0 \quad \Rightarrow \quad \zeta^T y(t) = \text{const} \tag{B.1}$$

When a GARK method applied to this system, the step update (2.3) satisfies

$$\zeta^T y_{n+1} = \zeta^T y_n + H \sum_{j=1}^{s^{\{\mathfrak{f}\}}} b_j^{\{\mathfrak{f}\}} \, \zeta^T \, f^{\{\mathfrak{f}\}}\left(Y_j^{\{\mathfrak{f}\}}\right) + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} b_j^{\{\mathfrak{s}\}} \, \zeta^T \, f^{\{\mathfrak{s}\}}\left(Y_j^{\{\mathfrak{s}\}}\right) \tag{B.2}$$

In order to apply (B.1), the arguments of $f^{\{\mathfrak{f}\}}$ and $f^{\{\mathfrak{s}\}}$ must be identical. In general, the arguments in (B.2) are different: $Y_j^{\{\mathfrak{f}\}} \neq Y_j^{\{\mathfrak{s}\}}$. Moreover, the number of slow stages can be different than the number of fast stages, which prevents the pairing of terms as in (B.1). Therefore, a general GARK method cannot be expected to preserve linear invariants.

There are special cases, however, where it is possible. If the subsystems individually satisfy

$$\zeta^T f^{\{\mathfrak{f}\}}(y) = 0, \quad \text{and} \quad \zeta^T f^{\{\mathfrak{s}\}}(y) = 0,$$

one can see (B.2) simplifies to $\zeta^T y_{n+1} = \zeta^T y_n$. Also, when

$$\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} = \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}, \quad \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} = \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}, \quad \mathbf{b}^{\{\mathfrak{f}\}} = \mathbf{b}^{\{\mathfrak{s}\}}, \tag{B.3}$$

we can achieve $s^{\{\mathfrak{f}\}} = s^{\{\mathfrak{s}\}}$ and $Y_j^{\{\mathfrak{f}\}} = Y_j^{\{\mathfrak{s}\}}$. In fact, this GARK method degenerates into a special class of partitioned Runge–Kutta schemes known to preserve linear invariants [75]. The multirate methods in [41], for example, satisfy (B.3).

Appendix C

New Coupled MRI-GARK methods

Here, we present the newly derived SPC-MRI-GARK and IPC-MRI-GARK methods. In some cases, the exact representation of the method coefficients is too long to fit on a page, so the first 16 digits are provided. The stability regions are computed according to [129]:

$$\begin{aligned}
\mathcal{S}^{1\text{D}}_{\rho,\alpha} &= \left\{ z^{\{\mathfrak{s}\}} \in \mathbb{C} \,\middle|\, \left| R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) \right| \leq 1, \forall z^{\{\mathfrak{f}\}} \in \mathbb{C}^- : \left| z^{\{\mathfrak{f}\}} \right| \leq \rho, \left| \arg\left(z^{\{\mathfrak{f}\}}\right) - \pi \right| \leq \alpha \right\}, \\
\mathcal{S}^{2\text{D}}_{\rho,\alpha} &= \left\{ z^{\{\mathfrak{s}\}} \in \mathbb{C} \,\middle|\, \max \left| \operatorname{eig} \mathbf{M}\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{f}\}}\right) \right| \leq 1, \right. \\
&\qquad \left. \forall z^{\{\mathfrak{f}\}} \in \mathbb{C}^- : \left| z^{\{\mathfrak{f}\}} \right| \leq \rho, \left| \arg\left(z^{\{\mathfrak{f}\}}\right) - \pi \right| \leq \alpha \right\}.
\end{aligned}$$

C.1 SPC-MRI-GARK Methods

We use the following tableau representation for SPC-MRI-GARK methods:

$$\begin{array}{c|ccc}
c_1^{\{\mathfrak{s}\}} & a_{1,1}^{\{\mathfrak{s}\}} & \dots & a_{1,s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} \\
\vdots & \vdots & \ddots & \vdots \\
c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} & a_{s^{\{\mathfrak{s}\}},1}^{\{\mathfrak{s}\}} & \dots & a_{s^{\{\mathfrak{s}\}},s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} \\
\hline
 & \gamma_1(t) & \dots & \gamma_{s^{\{\mathfrak{s}\}}}(t) \\
\hline
 & \widehat{\gamma}_1(t) & \dots & \widehat{\gamma}_{s^{\{\mathfrak{s}\}}}(t)
\end{array}.$$

C.1.1 SDIRK2(1)2

This method is based on the two stage, second order method in [4].

$$\begin{array}{c|cc}
1-\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & 0 \\
1 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
 & \left(12-9\sqrt{2}\right)t+5\sqrt{2}-6 & \left(9\sqrt{2}-12\right)t-5\sqrt{2}+7 \\
\hline
 & \left(\frac{78}{5}-12\sqrt{2}\right)t+6\sqrt{2}-\frac{36}{5} & \left(12\sqrt{2}-\frac{78}{5}\right)t-6\sqrt{2}+\frac{41}{5}
\end{array}$$

C.1.2 ESDIRK2(1)3

This method is based on TR-BDF2 in [15].

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
2-\sqrt{2} & 1-\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & 0 \\
1 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
 & \left(6-\frac{9}{\sqrt{2}}\right)t+\frac{5}{\sqrt{2}}-3 & \left(6-\frac{9}{\sqrt{2}}\right)t+\frac{5}{\sqrt{2}}-3 & \left(9\sqrt{2}-12\right)t-5\sqrt{2}+7 \\
\hline
 & \left(\frac{39}{5}-6\sqrt{2}\right)t+3\sqrt{2}-\frac{18}{5} & \left(\frac{39}{5}-6\sqrt{2}\right)t+3\sqrt{2}-\frac{18}{5} & \left(12\sqrt{2}-\frac{78}{5}\right)t-6\sqrt{2}+\frac{41}{5}
\end{array}
$$

C.1.3 SDIRK3(2)4

This method is based on SDIRK3M in [86].

$$
\begin{array}{c|cccc}
\frac{9}{40} & \frac{9}{40} & 0 & 0 & 0 \\
\frac{7}{13} & \frac{163}{520} & \frac{9}{40} & 0 & 0 \\
\frac{11}{15} & -\frac{6481433}{8838675} & \frac{87795409}{70709400} & \frac{9}{40} & 0 \\
1 & \frac{4032}{9943} & \frac{6929}{15485} & -\frac{723}{9272} & \frac{9}{40} \\
\hline
 & -\frac{21765t}{9943}+\frac{3}{2} & \frac{18740344238109t}{12407262101200}-\frac{46850957023}{152236344800} & -\frac{2318739807t}{928641703280}-\frac{2336165553}{30447268960} & \frac{341049771t}{500777450}-\frac{231399837}{2003109800} \\
\hline
 & -\frac{458t}{153}+\frac{17}{9} & \frac{1143703567597t}{484654507050}-\frac{5}{7} & \frac{12128361703356241349t}{41321158297274157120}-\frac{3214490524810792571}{14788625074813908864} & \frac{6985915649614123877t}{20539757048352651200}+\frac{70261070970241507}{1643180563868212096}
\end{array}
$$

C.1.4 ESDIRK3(2)4

This method is based on the optimal four stage, third order ESDIRK method described in [86].

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
0.8717330430169180 & 0.4358665215084590 & 0.4358665215084590 & 0 & 0 \\
0.6089666303771147 & 0.2648804871412033 & -0.09178037827254760 & 0.4358665215084590 & 0 \\
1.000000000000000 & 0.1921013555637903 & -0.6181218831132021 & 0.9901540060409528 & 0.4358665215084590 \\
\hline
 & 0.2335954530133717t+0.07530362905710443 & 3.847836453450424t-2.542040109838414 & -4.416875540651942t+3.198591776366924 & 0.2435443634188146 6t+0.2681447044143857 \\
\hline
 & -0.5331294033713856t+0.24812962236875004 & 0.1096316239241135t-1.000000000000000 & -0.7855025327869668t+1.688048335476923 & 1.209000312234239t-0.06934455916442332
\end{array}
$$

C.1.5 SDIRK4(3)5

This method is based on SDIRK4M in [86].

$$
\begin{array}{c|ccccc}
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\
\frac{9}{10} & \frac{13}{20} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{2}{3} & \frac{580}{1287} & -\frac{175}{5148} & \frac{1}{4} & 0 & 0 \\
\frac{3}{5} & \frac{12698}{37375} & -\frac{201}{2990} & \frac{891}{11500} & \frac{1}{4} & 0 \\
1 & \frac{944}{1365} & -\frac{400}{819} & \frac{99}{35} & -\frac{575}{252} & \frac{1}{4} \\
\hline
 & \frac{487}{273} - \frac{142t}{65} & -\frac{125t}{182} - \frac{475}{3276} & \frac{297t}{140} + \frac{99}{56} & -\frac{575}{252} & \frac{3t}{4} - \frac{1}{8} \\
\hline
 & \begin{array}{c}\frac{357179t}{270270} \\ +\frac{1}{27}\end{array} & \begin{array}{c}\frac{222331t}{72072} \\ -\frac{17}{8}\end{array} & \begin{array}{c}\frac{1135934341t}{442769040} \\ +\frac{110483689}{63252720}\end{array} & \begin{array}{c}-\frac{11524110095t}{1461137832} \\ +\frac{28581755}{18975816}\end{array} & \begin{array}{c}\frac{636740663t}{695779920} \\ -\frac{10434149}{63252720}\end{array}
\end{array}
$$

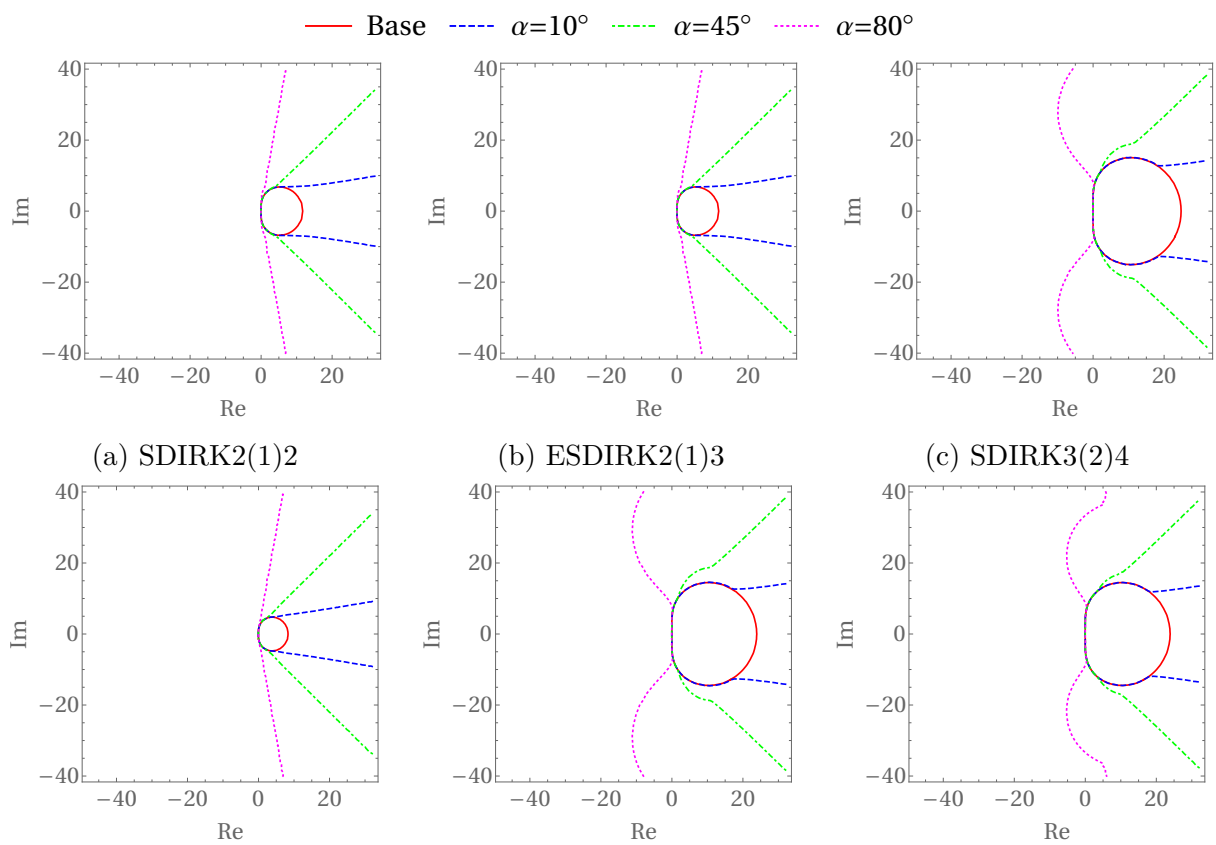
C.1.6 ESDIRK4(3)6

This method is based on ESDIRK4(3)6L[2]SA in [86] and has the property that the first and second column of coefficients are identical.

$$
\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & a_{2,1}^{\{s,s\}} & 0.2500000000000000 & 0 & 0 & 0 & 0 \\
\frac{2-\sqrt{2}}{4} & a_{3,1}^{\{s,s\}} & -0.05177669529663688 & 0.2500000000000000 & 0 & 0 & 0 \\
\frac{5}{8} & a_{4,1}^{\{s,s\}} & -0.07655460838455727 & 0.5281092167691145 & 0.2500000000000000 & 0 & 0 \\
\frac{26}{25} & a_{5,1}^{\{s,s\}} & -0.7274063478261298 & 1.584995061740679 & 0.6598176339115803 & 0.2500000000000000 & 0 \\
1 & a_{6,1}^{\{s,s\}} & -0.01558763503571650 & 0.24876576709132033 & 0.5017726195721632 & -0.1082550204139335 & 0.2500000000000000 \\
\hline
 & \gamma_1(t) & \begin{array}{c}-6.163979155637189t \\ +3.066401942782878\end{array} & \begin{array}{c}8.775315341826407t \\ -4.000000000000000\end{array} & \begin{array}{c}2.197069503808978t \\ -0.5967621323323260\end{array} & \begin{array}{c}1.703312350342134t \\ -0.9599111955850004\end{array} & \begin{array}{c}-0.24477388847031400t \\ +0.4238694423515700\end{array} \\
\hline
 & \widehat{\gamma}_1(t) & \begin{array}{c}-4.935764673620373t \\ +2.375000000000000\end{array} & \begin{array}{c}7.151127236629060t \\ -3.058823529411765\end{array} & \begin{array}{c}1.151758875793870t \\ -0.05607965938087753\end{array} & \begin{array}{c}3.303286684519598t \\ -1.734976675593132\end{array} & \begin{array}{c}-1.734643449701781t \\ +1.099879864385774\end{array}
\end{array}
$$

C.1.7 Stability Plots

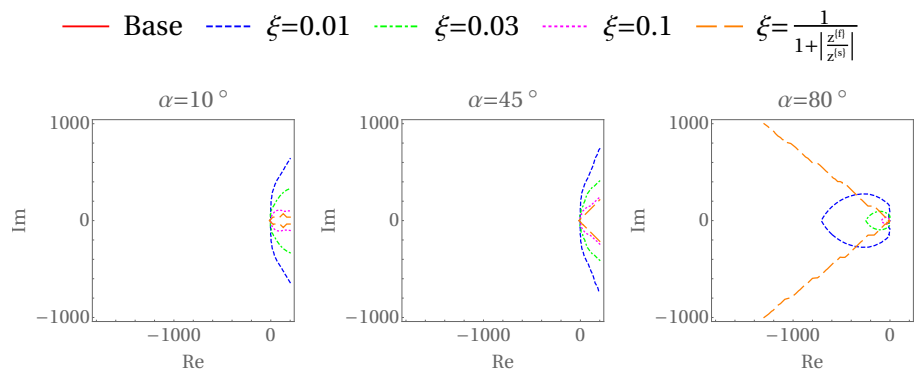
Figures C.1 and C.2 show the scalar and matrix stability regions, respectively.



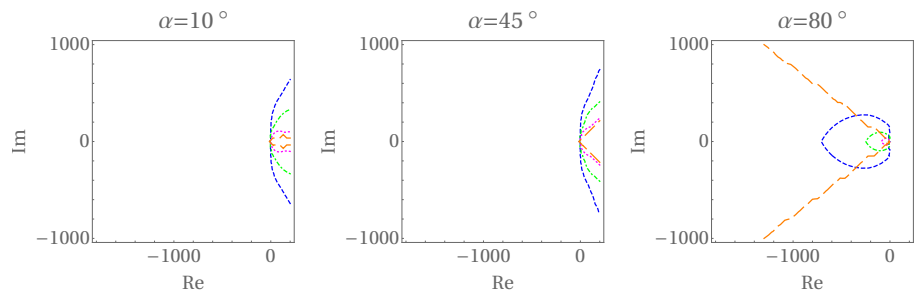
(a) SDIRK2(1)2 (b) ESDIRK2(1)3 (c) SDIRK3(2)4

(d) ESDIRK3(2)4 (e) SDIRK4(3)5 (f) ESDIRK4(3)6

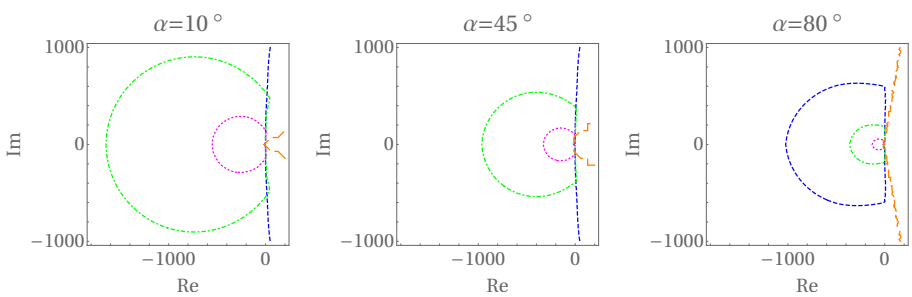
Figure C.1: Scalar stability regions $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ for SPC-MRI-GARK methods.



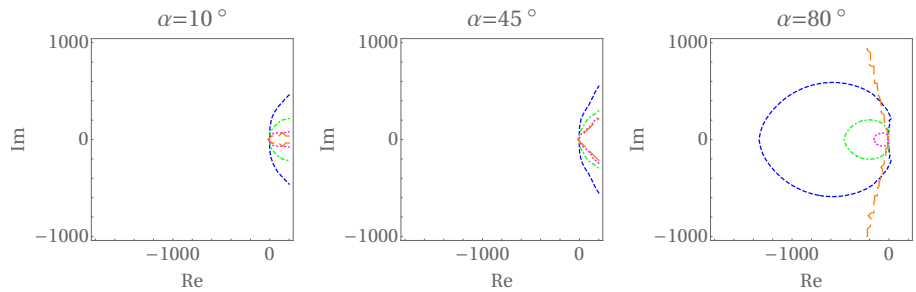
(a) SDIRK2(1)2



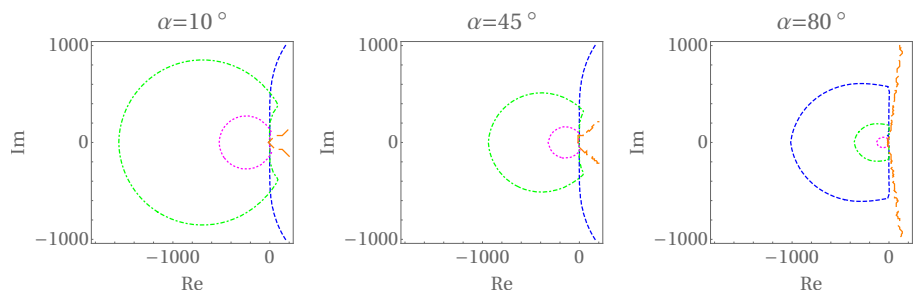
(b) ESDIRK2(1)3



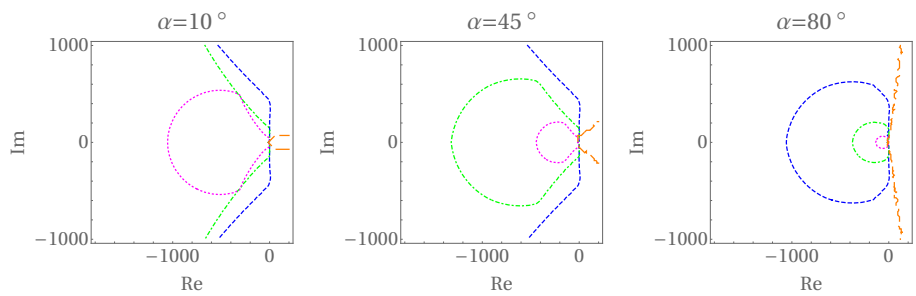
(c) SDIRK3(2)4



(d) ESDIRK3(2)4



(e) SDIRK4(3)5



(f) ESDIRK4(3)6

Figure C.2: Matrix stability regions $\mathcal{S}^{2\text{D}}_{\infty,\alpha}$ for SPC-MRI-GARK methods.

C.2 IPC-MRI-GARK Methods

We will using the following tableau representation for IPC-MRI-GARK methods:

$$
\begin{array}{c|cccc|cccc}
c_1^{\{\mathfrak{s}\}} & & & & & \psi_{1,1}(t) & & & \\
c_2^{\{\mathfrak{s}\}} & \gamma(t)_{2,1} & & & & \psi_{2,1}(t) & \psi_{2,2}(t) & & \\
\vdots & \vdots & \ddots & & & \vdots & \vdots & \ddots & \\
c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}} & \gamma(t)_{s^{\{\mathfrak{s}\}},1} & \dots & \gamma(t)_{s^{\{\mathfrak{s}\}},s^{\{\mathfrak{s}\}}-1} & & \psi_{s^{\{\mathfrak{s}\}},1}(t) & \psi_{s^{\{\mathfrak{s}\}},2}(t) & \dots & \psi_{s^{\{\mathfrak{s}\}},s^{\{\mathfrak{s}\}}}(t) \\
\hline
 & \widehat{\gamma}_1(t) & \dots & \widehat{\gamma}_{s^{\{\mathfrak{s}\}}-1}(t) & 0 & \widehat{\psi}_1(t) & \widehat{\psi}_2(t) & \dots & \widehat{\psi}_{s^{\{\mathfrak{s}\}}}(t)
\end{array}
.
$$

C.2.1 SDIRK2(1)2

This method is based on the two stage, second order method in [4].

$$
\begin{array}{c|cc|cc}
1-\frac{1}{\sqrt{2}} & 0 & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\
1 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}}-1 & 1-\frac{1}{\sqrt{2}} \\
\hline
 & \frac{3}{5} & 0 & \frac{1}{\sqrt{2}}-1 & \frac{2}{5}
\end{array}
$$

C.2.2 ESDIRK2(1)3

This method is based on TR-BDF2 in [15].

$$
\begin{array}{c|ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2-\sqrt{2} & 1-\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 1-\frac{1}{\sqrt{2}} & 0 \\
1 & \frac{3}{2\sqrt{2}}-1 & \frac{1}{2\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}}-1 & 1-\frac{1}{\sqrt{2}} \\
\hline
 & \frac{1}{\sqrt{2}}-\frac{7}{10} & \frac{3}{10} & 0 & 0 & \frac{1}{\sqrt{2}}-1 & \frac{2}{5}
\end{array}
$$

C.2.3 SDIRK3(2)5

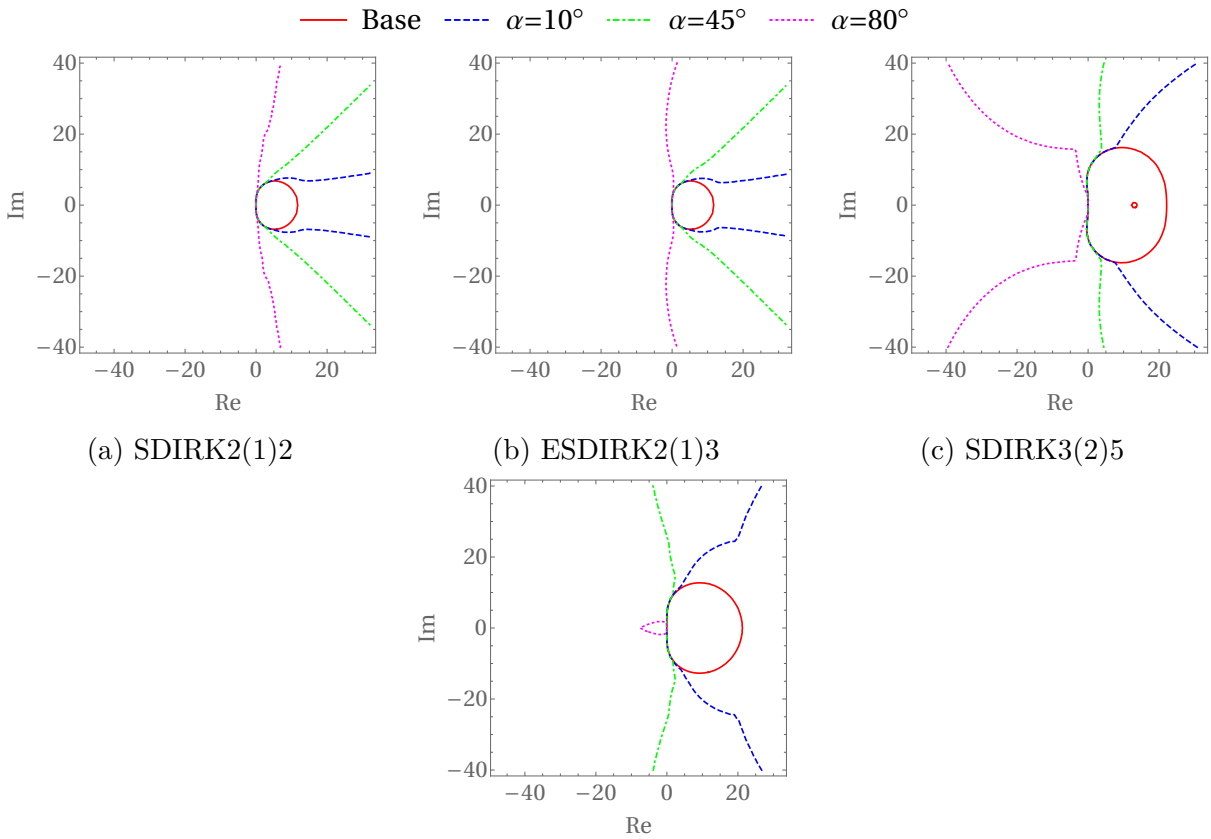
$$
\begin{array}{c|ccccc|cccccc}
\frac{7}{40} & 0 & 0 & 0 & 0 & 0 & \frac{7}{40} & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{19}{120} & 0 & 0 & 0 & 0 & -\frac{7}{40} & \frac{7}{40} & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{10} & -\frac{1}{10} & 0 & 0 & 0 & 0 & -\frac{7}{40} & \frac{7}{40} & 0 & 0 \\
1 & \frac{17341}{182400} & -\frac{73}{70} & \frac{687111}{425600} & 0 & 0 & 0 & 0 & -\frac{7}{40} & \frac{7}{40} & 0 \\
1 & -\frac{21487}{60800} & \frac{1618427}{1702400} & -\frac{1144471}{1702400} & \frac{3}{40} & 0 & 0 & 0 & 0 & -\frac{7}{40} & \frac{7}{40} \\
\hline
 & \frac{2833}{60800} & -\frac{9}{35} & \frac{17257}{425600} & \frac{1}{6} & 0 & 0 & 0 & 0 & -\frac{7}{40} & \frac{107}{600}
\end{array}
$$

C.2.4 SDIRK4(3)6

$$
\begin{array}{c|cccccc|cccccc}
\frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & -\frac{73t}{70} + \frac{4}{7} & 0 & 0 & 0 & 0 & 0 & \frac{73t}{14} - \frac{393}{140} & -\frac{146t}{35} + \frac{16}{7} & 0 & 0 & 0 & 0 \\
\frac{1}{2} & -\frac{2592641t}{425250} + \frac{2253133}{425250} & \frac{32t}{7} - \frac{30}{7} & 0 & 0 & 0 & 0 & \frac{454241t}{85050} - \frac{454241}{170100} & -\frac{714082t}{212625} + \frac{314516}{212625} & -\frac{16t}{35} + \frac{3}{7} & 0 & 0 & 0 \\
\frac{1}{2} & -\frac{79813t}{26425} - \frac{5}{14} & \frac{417821t}{79275} - \frac{5}{6} & -\frac{180296t}{237825} + \frac{4}{9} & 0 & 0 & 0 & -\frac{23293t}{1134} + \frac{23293}{2268} & \frac{20473t}{945} - \frac{20473}{1890} & -\frac{2891t}{9720} - \frac{997}{19440} & -\frac{22537t}{9720} + \frac{5285}{3888} & 0 & 0 \\
\frac{3}{4} & -\frac{1709523149t}{68615910} + \frac{6626912}{467775} & \frac{8462196t}{449225} - \frac{81}{7} & \frac{2352991367t}{1035014400} - \frac{8}{9} & \frac{180121t}{143616} - \frac{2}{11} & 0 & 0 & \frac{7713555547t}{310789710} - \frac{7713555547}{621579420} & -\frac{1703745478t}{70634025} + \frac{851872739}{70634025} & -\frac{3353446993t}{1130144400} + \frac{3353446993}{2260288800} & \frac{137392139t}{32289840} - \frac{30061615}{12915936} & \frac{360242t}{639485} - \frac{52224}{639485} & 0 \\
1 & \frac{1646963990099t}{204132332250} - \frac{796870764337}{204132332250} & -\frac{78294288t}{7636825} + \frac{113260367}{22910475} & \frac{49839881579t}{17595244800} - \frac{28671224497}{17595244800} & -\frac{5075915t}{2441472} + \frac{4299217}{2441472} & \frac{152t}{165} - \frac{2}{3} & 0 & \frac{9215792648141t}{449091130950} - \frac{9215792648141}{898182261900} & -\frac{349735368626t}{14580880875} + \frac{174867684313}{14580880875} & \frac{115392839939t}{653223463200} - \frac{9832286}{10871245} & \frac{61269407807t}{18663527520} - \frac{115392839939}{1306446926400} & \frac{15316074t}{10871245} - \frac{61269407807}{37327055040} & -\frac{76t}{85} + \frac{11}{17} \\
\hline
 & \frac{694507614551t}{96062274000} - \frac{66946175035 1}{192124548000} & -\frac{9882343t}{1078140} + \frac{9560707}{2156280} & \frac{13007509307t}{4968069120} - \frac{14640287027}{9936138240} & -\frac{1639519t}{940032} + \frac{2715895}{1880064} & \frac{49t}{99} - \frac{35}{99} & 0 & \frac{13988077t}{680400} - \frac{13988077}{1360800} & -\frac{10360601t}{425250} + \frac{10360601}{850500} & \frac{2t}{15} - \frac{1}{15} & \frac{11t}{3} - \frac{11}{6} & \frac{27t}{20} - \frac{7}{8} & -\frac{7t}{9} + \frac{5}{9}
\end{array}
$$

C.2.5 Stability Plots

Figures C.3 and C.4 show the scalar and matrix stability regions, respectively.



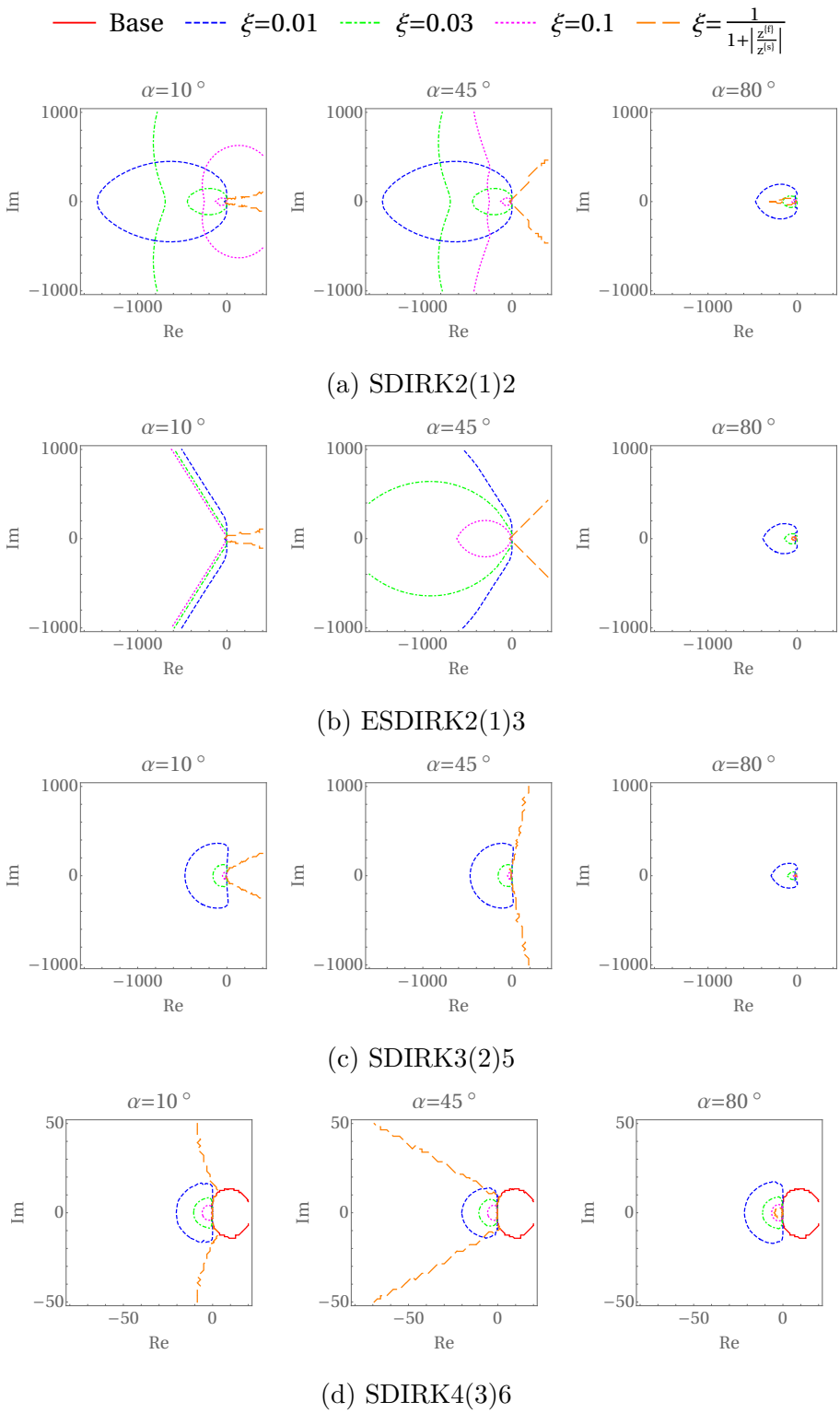
(a) SDIRK2(1)2

(b) ESDIRK2(1)3

(c) SDIRK3(2)5

(d) SDIRK4(3)6

Figure C.3: Scalar stability regions $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ for IPC-MRI-GARK methods.



(a) SDIRK2(1)2

(b) ESDIRK2(1)3

(c) SDIRK3(2)5

(d) SDIRK4(3)6

Figure C.4: Matrix stability regions $\mathcal{S}^{2\text{D}}_{\infty,\alpha}$ for IPC-MRI-GARK methods.

Appendix D

Explicit MRI-GARK and SPC-MRI-GARK Methods of Orders Two and Three

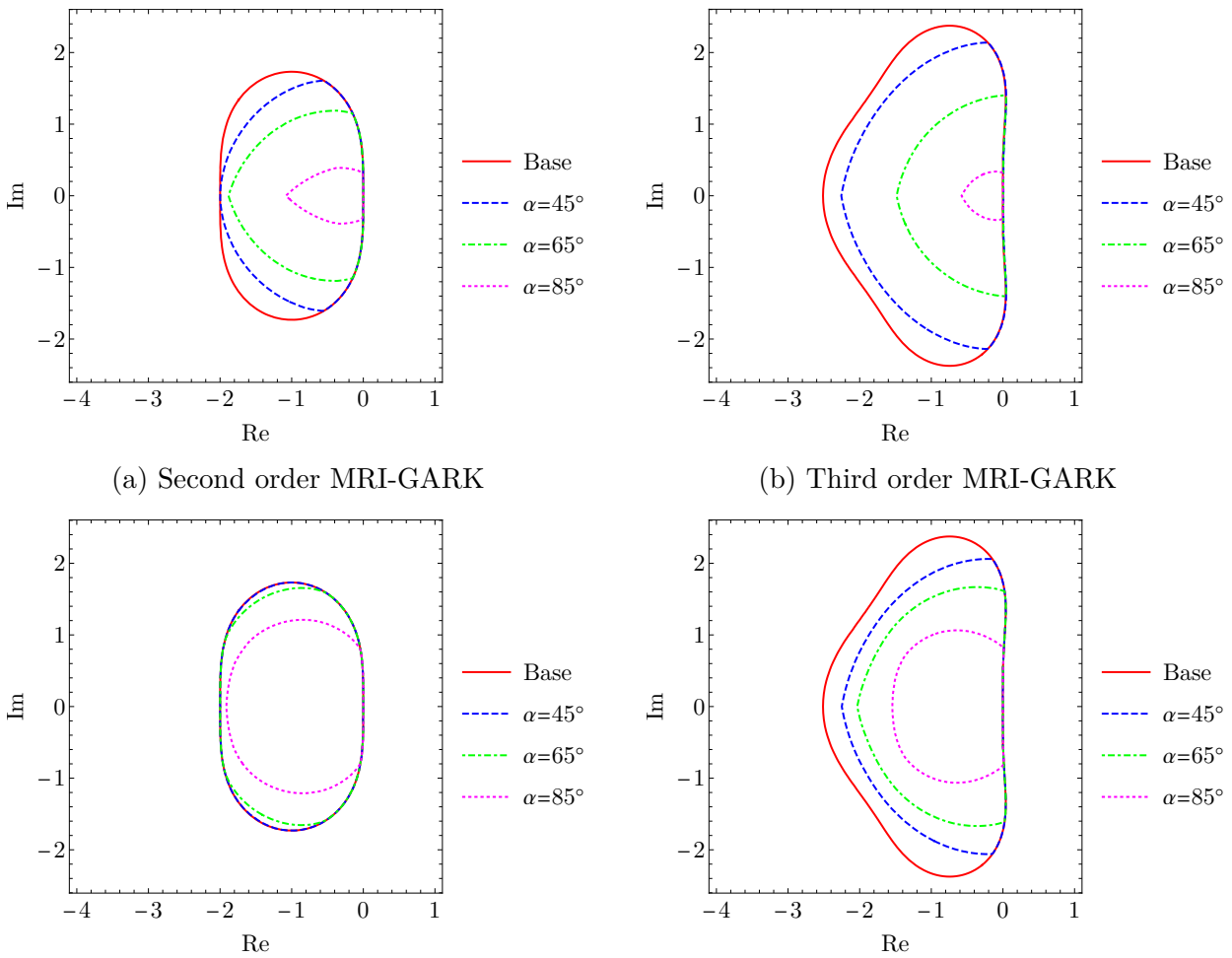
Table D.1 provides coefficients for new MRI-GARK and SPC-MRI-GARK schemes based on Ralston's optimal second and third order Runge–Kutta methods [114]. Figure 2.1 plots their scalar slow stability regions [129, Definition 4.1]

$$\mathcal{S}^{\mathrm{1D}}_{\infty,\alpha} = \left\{ z^{\{\mathfrak{s}\}} \in \mathbb{C} \,\middle|\, \left| R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) \right| \leq 1,\ \forall z^{\{\mathfrak{f}\}} \in \mathbb{C}^- : \left| \arg\left(z^{\{\mathfrak{f}\}}\right) - \pi \right| \leq \alpha \right\},$$

where $R(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}})$ is the scalar linear stability function. All methods in table D.1 satisfy $\lim_{z^{\{\mathfrak{f}\}} \to -\infty} R(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}) = 0$, so they are suitable for problems where the fast dynamics are stiff, but the slow dynamics are nonstiff.

Order	Base Method	MRI-GARK $\Gamma(t), \widehat{\gamma}(t)$	SPC-MRI-GARK $\gamma(t), \widehat{\gamma}(t)$
2	$\begin{array}{c\|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$	$\begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{5}{12} & \frac{3}{4} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} \\ 0 \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{2} + \frac{3t}{2} \\ \frac{3}{2} - \frac{3t}{2} \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
3	$\begin{array}{c\|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{3}{4} & 0 & \frac{3}{4} & 0 \\ \hline & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{array}$	$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{11}{4} + \frac{9t}{2} & 3 - \frac{9t}{2} & 0 \\ \frac{47}{36} - \frac{13t}{6} & -\frac{1}{6} - \frac{t}{2} & -\frac{8}{9} + \frac{8t}{3} \end{bmatrix}, \begin{bmatrix} \frac{1}{40} \\ \frac{7}{40} \\ \frac{1}{20} \end{bmatrix}$	$\begin{bmatrix} 1 - \frac{2t}{3} - \frac{4t^2}{3} \\ -2t + 4t^2 \\ \frac{8t}{3} - \frac{8t^2}{3} \end{bmatrix}, \begin{bmatrix} -\frac{7}{8} + \frac{9t}{5} \\ \frac{71}{40} - \frac{17t}{10} \\ \frac{1}{10} - \frac{t}{10} \end{bmatrix}$

Table D.1: Second and third order MRI-GARK and SPC-MRI-GARK coefficients.



(a) Second order MRI-GARK

(b) Third order MRI-GARK

(c) Second order SPC-MRI-GARK

(d) Third order SPC-MRI-GARK

Figure D.1: Stability regions for new methods in table D.1 including the base Runge–Kutta stability region and $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ for $\alpha = 45°, 65°, 85°$.

Appendix E

Coefficients for Fourth Order IMEX Methods

E.1 GARK4(3)55L[1]SA

The coefficients for GARK4(3)55L[1]SA are listed in (E.1), and the stability is plotted in fig. E.1.

$$
\begin{array}{ccccc|ccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{12}{29} & 0 & 0 & 0 & 0 & \frac{12}{29} & 0 & 0 & 0 & 0 \\
\frac{873}{4232} & \frac{783}{4232} & 0 & 0 & 0 & \frac{2916}{6877} & -\frac{225}{6877} & 0 & 0 & 0 \\
-\frac{12214}{46875} & -\frac{58406}{78125} & \frac{461288}{234375} & 0 & 0 & \frac{110138}{180375} & \frac{58147}{180375} & \frac{1}{37} & 0 & 0 \\
-\frac{8045}{18792} & -\frac{215}{216} & \frac{636916}{256041} & -\frac{21875}{341388} & 0 & \frac{541865807}{1231557210} & \frac{283237585}{492622884} & -\frac{112254521}{63156780} & \frac{460}{261} & 0 \\
\hline
\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\
-\frac{581}{4800} & \frac{4901}{4800} & 0 & 0 & 0 & \frac{13}{20} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{46303}{218592} & \frac{172289}{218592} & -\frac{1}{3} & 0 & 0 & \frac{580}{1287} & -\frac{175}{5148} & \frac{1}{4} & 0 & 0 \\
\frac{13559}{42320} & \frac{1203761}{1587000} & -\frac{462536}{940125} & \frac{525}{40112} & 0 & \frac{12698}{37375} & -\frac{201}{2990} & \frac{891}{11500} & \frac{1}{4} & 0 \\
\frac{317}{2592} & 0 & \frac{279841}{494424} & \frac{78125}{94176} & -\frac{29}{56} & \frac{944}{1365} & -\frac{400}{819} & \frac{99}{35} & -\frac{575}{252} & \frac{1}{4} \\
\hline
\frac{317}{2592} & 0 & \frac{279841}{494424} & \frac{78125}{94176} & -\frac{29}{56} & \frac{944}{1365} & -\frac{400}{819} & \frac{99}{35} & -\frac{575}{252} & \frac{1}{4} \\
\hline
\frac{29}{432} & 0 & \frac{2116}{2943} & -\frac{625}{5232} & \frac{1}{3} & \frac{41911}{60060} & -\frac{83975}{144144} & \frac{3393}{1120} & -\frac{27025}{11088} & \frac{103}{352}
\end{array}
\tag{E.1}
$$

E.2 GARK4(3)77L[2]SA

For GARK4(3)77L[2]SA, rational approximations of the coefficients accurate to 16 digits are listed in (E.2). The stability is plotted in fig. E.2.

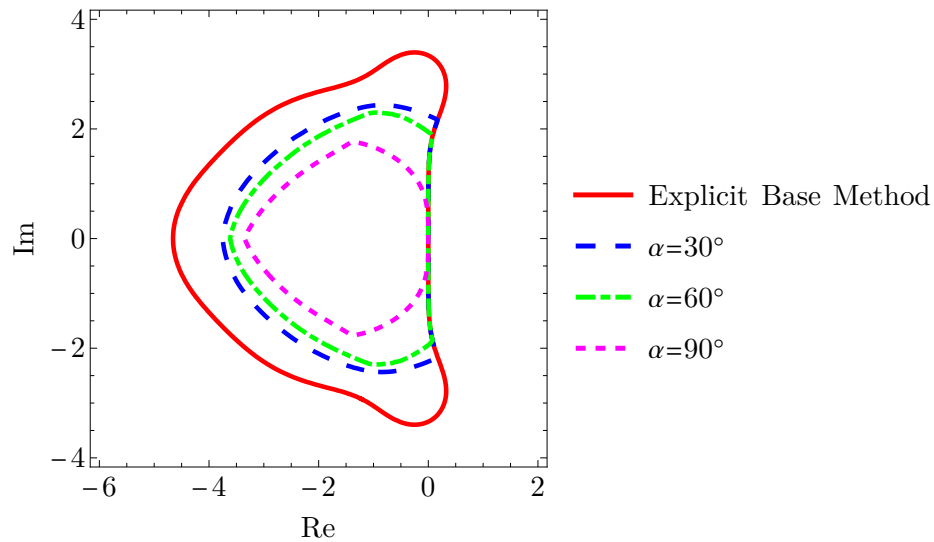


Figure E.1: Stability regions for (E.1) including the explicit base method and $\mathcal{S}^{1\mathrm{p}}_{\infty,\alpha}$ for three values of α.

$$
\begin{array}{ccccccc|ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{9} & \frac{2}{9} & 0 & 0 & 0 & 0 & 0 & \frac{16425639}{54608393} & \frac{16425639}{54608393} & -\frac{16425639}{61233502} & 0 & 0 & 0 & 0 \\
\frac{1419}{19375} & -\frac{1026}{19375} & \frac{11232}{19375} & 0 & 0 & 0 & 0 & -\frac{77612501}{130721827} & -\frac{77612501}{130721827} & \frac{151}{114} & \frac{34759432}{75093219} & 0 & 0 & 0 \\
\frac{21003}{128960} & -\frac{31311}{64480} & \frac{932337}{1031680} & \frac{1125}{6656} & 0 & 0 & 0 & -\frac{144402891}{102268870} & -\frac{144402891}{102268870} & \frac{419}{158} & \frac{90561680}{88030521} & -\frac{8}{75} & 0 & 0 \\
-\frac{91453}{272025} & \frac{53218}{30225} & -\frac{1137}{10075} & -\frac{775}{351} & \frac{256}{135} & 0 & 0 & \frac{167805857}{109921914} & \frac{167805857}{109921914} & -\frac{174817899}{83527346} & -\frac{213601621}{169356737} & \frac{135572762}{78568309} & -\frac{45}{106} & 0 \\
\frac{11}{108} & 0 & \frac{81}{160} & -\frac{125}{864} & \frac{64}{135} & \frac{1}{16} & 0 & -\frac{60852872}{151808973} & -\frac{60852872}{151808973} & \frac{138068585}{147014034} & \frac{47729168}{92044891} & \frac{68033599}{87727555} & -\frac{18138990}{32594683} & \frac{1}{8} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 \\
\frac{4341401}{47990009} & -\frac{1}{58} & 0 & 0 & 0 & 0 & 0 & -\frac{2655929}{102591677} & -\frac{2655929}{102591677} & \frac{1}{8} & 0 & 0 & 0 & 0 \\
\frac{32077659}{171989725} & -\frac{942861}{13337978} & \frac{68}{177} & 0 & 0 & 0 & 0 & \frac{29303500}{86597249} & \frac{29303500}{86597249} & -\frac{21306286}{70602821} & \frac{1}{8} & 0 & 0 & 0 \\
\frac{6422149}{55910568} & \frac{2221181}{82071183} & \frac{47}{130} & \frac{17}{88} & 0 & 0 & 0 & -\frac{67256452}{187215923} & -\frac{67256452}{187215923} & \frac{106344487}{113554292} & \frac{16506257}{46676381} & \frac{1}{8} & 0 & 0 \\
\frac{17269313}{120249265} & -\frac{7489877}{130828222} & \frac{6557339}{16703351} & \frac{64609429}{243455733} & -\frac{3}{79} & 0 & 0 & \frac{20074360}{85930857} & \frac{20074360}{85930857} & -\frac{4152823}{95874112} & \frac{1490617}{78318549} & \frac{11281353}{81506416} & \frac{1}{8} & 0 \\
\frac{11}{108} & 0 & \frac{81}{160} & -\frac{125}{864} & \frac{64}{135} & \frac{1}{16} & 0 & -\frac{60852872}{151808973} & -\frac{60852872}{151808973} & \frac{138068585}{147014034} & \frac{47729168}{92044891} & \frac{68033599}{87727555} & -\frac{18138990}{32594683} & \frac{1}{8} \\
\hline
\frac{11}{108} & 0 & \frac{81}{160} & -\frac{125}{864} & \frac{64}{135} & \frac{1}{16} & 0 & -\frac{60852872}{151808973} & -\frac{60852872}{151808973} & \frac{138068585}{147014034} & \frac{47729168}{92044891} & \frac{68033599}{87727555} & -\frac{18138990}{32594683} & \frac{1}{8} \\
\hline
\frac{3534129}{117061301} & 0 & \frac{81172009}{106874141} & -\frac{38440414}{91218725} & \frac{28234841}{53461673} & \frac{14200781}{196024533} & \frac{2448683}{78654186} & -\frac{6}{13} & -\frac{6}{13} & \frac{157844815}{152821426} & \frac{43658869}{72278142} & \frac{61557453}{85975976} & -\frac{60110287}{104593124} & \frac{28607143}{197443431}
\end{array}
\tag{E.2}
$$

E.3 ARK4(3)8L[2]DAE

For ARK4(3)8L[2]DAE, the explicit method is given in (E.3a), and the implicit method is given in (E.3b). The coefficients are rational approximation accurate to 16 digits. The stability is plotted in fig. E.3.

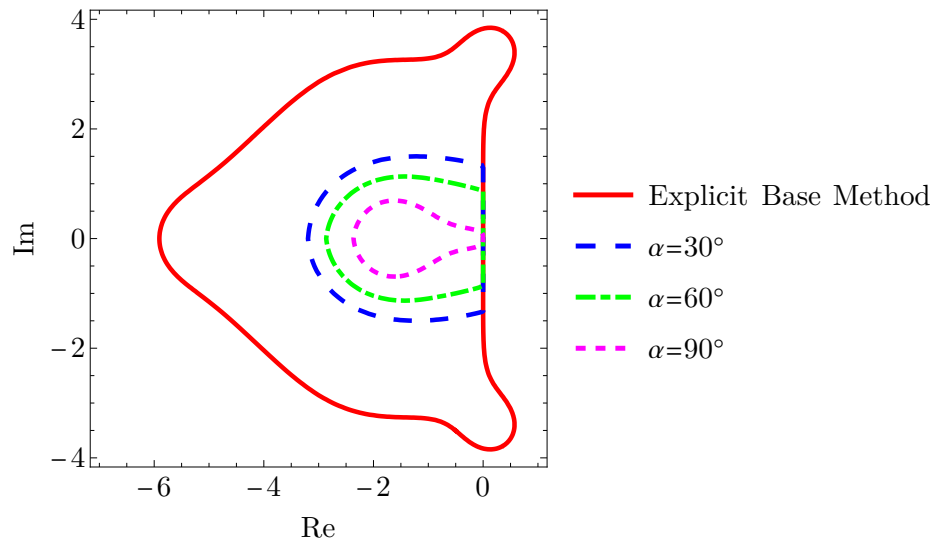


Figure E.2: Stability regions for (E.2) including the explicit base method and $\mathcal{S}^{1\mathrm{p}}_{\infty,\alpha}$ for three values of α.

$$
\begin{array}{c|ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{12}{59} & \frac{12}{59} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{41595977}{119801266} & \frac{3}{59} & \frac{44577989}{150418017} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{2078383}{19819871} & \frac{15151893}{178695740} & \frac{9}{29} & 0 & 0 & 0 & 0 & 0 \\
\frac{2}{3} & \frac{21521849}{106885896} & -\frac{18088527}{97911278} & \frac{45453627}{106466378} & \frac{20651489}{92554520} & 0 & 0 & 0 & 0 \\
\frac{5}{6} & \frac{21016693}{70682152} & -\frac{20134127}{73715441} & \frac{8047741}{69753204} & \frac{34683458}{57930339} & \frac{23}{242} & 0 & 0 & 0 \\
1 & \frac{14892883}{195274309} & \frac{1527884}{55401433} & \frac{88519811}{46789874} & -\frac{145571032}{39359105} & \frac{519904919}{153209666} & -\frac{42178121}{61075558} & 0 & 0 \\
1 & -\frac{729945}{105392882} & 0 & \frac{175213651}{68041802} & -\frac{319310885}{60712359} & \frac{336786603}{59367020} & -\frac{374157429}{143238715} & \frac{94554463}{149988897} & 0 \\
\hline
 & -\frac{729945}{105392882} & 0 & \frac{175213651}{68041802} & -\frac{319310885}{60712359} & \frac{336786603}{59367020} & -\frac{374157429}{143238715} & \frac{94554463}{149988897} & 0 \\
\hline
 & -\frac{598579}{90302121} & 0 & \frac{98460233}{81131138} & -\frac{92740625}{164372423} & -\frac{68651397}{107251003} & \frac{34}{27} & -\frac{29}{68} & \frac{13}{79}
\end{array}
\tag{E.3a}
$$

$$
\begin{array}{c|ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{12}{59} & \frac{6}{59} & \frac{6}{59} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{41595977}{119801266} & \frac{36987413}{301306866} & \frac{36987413}{301306866} & \frac{6}{59} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{21034655}{161026087} & \frac{21034655}{161026087} & \frac{21167767}{154455860} & \frac{6}{59} & 0 & 0 & 0 & 0 \\
\frac{2}{3} & \frac{14849759}{118829980} & \frac{14849759}{118829980} & \frac{15764882}{84485571} & \frac{14}{109} & \frac{6}{59} & 0 & 0 & 0 \\
\frac{5}{6} & \frac{17759028}{121612409} & \frac{17759028}{121612409} & -\frac{4469931}{76119220} & \frac{69643667}{146859490} & \frac{1495617}{62105222} & \frac{6}{59} & 0 & 0 \\
1 & \frac{8417779}{122161672} & \frac{8417779}{122161672} & \frac{102307162}{68015499} & -\frac{231602825}{82262861} & \frac{969445082}{353784535} & -\frac{92041417}{137684195} & \frac{6}{59} & 0 \\
1 & 0 & 0 & \frac{112004121}{45699490} & -\frac{649350941}{131324444} & \frac{468685465}{87863957} & -\frac{230333404}{94630015} & \frac{43250704}{87929917} & \frac{6}{59} \\
\hline
 & 0 & 0 & \frac{112004121}{45699490} & -\frac{649350941}{131324444} & \frac{468685465}{87863957} & -\frac{230333404}{94630015} & \frac{43250704}{87929917} & \frac{6}{59} \\
\hline
 & 0 & 0 & \frac{276942538}{113184465} & -\frac{494936623}{98687460} & \frac{481777685}{86509293} & -\frac{233204891}{87140409} & \frac{163872041}{295010343} & \frac{3}{25}
\end{array}
\tag{E.3b}
$$

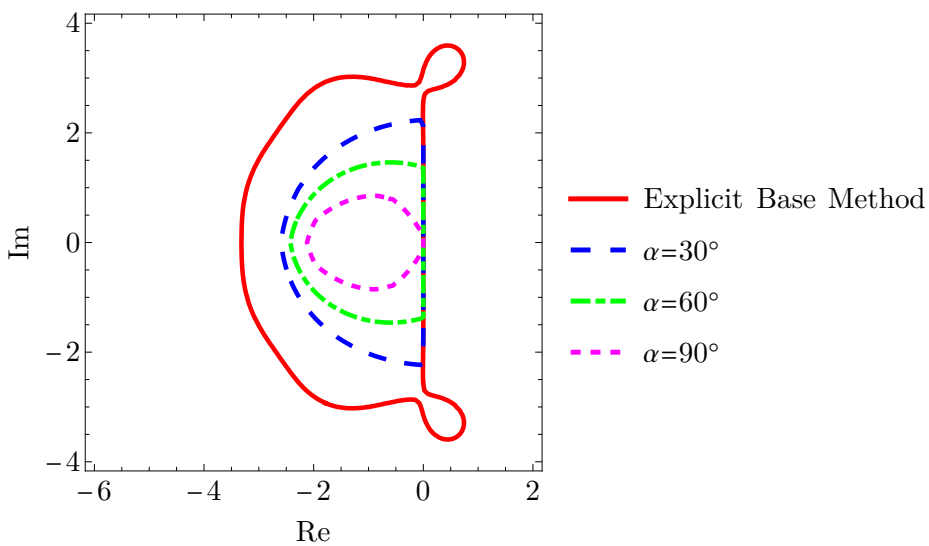


Figure E.3: Stability regions for (E.3) including the explicit base method and $\mathcal{S}^{1\text{D}}_{\infty,\alpha}$ for three values of α.