Time Stepping Methods for Multiphysics Problems

Arash Sarshar

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
Young Cao
Traian Iliescu
Carol S. Woodward

August 12, 2021
Blacksburg, Virginia

Keywords: Time integration methods, Numerical methods for initial value problems

Time Stepping Methods for Multiphysics Problems

Arash Sarshar

(ABSTRACT)

Mathematical modeling of physical processes often leads to systems of differential and algebraic equations involving quantities of interest. A computer model created based on these equations can be numerically integrated to predict future states of the system and its evolution in time. This thesis investigates current methods in numerical time-stepping schemes, identifying a number of important features needed to speed up and increase the accuracy of the solutions. The focus is on developing new methods suitable for large-scale applications with multiple physical processes, potentially with significant differences in their time-scales. Various families of new methods are introduced with special attention to multirating, low computational cost implicitness, high order of convergence, and robustness. For each family, the order condition theory is discussed and a number of examples are derived. The accuracy and stability of the methods are investigated using standard analysis techniques and numerical experiments are performed to verify the abilities of the new methods.

Time Stepping Methods for Multiphysics Problems

Arash Sarshar

(GENERAL AUDIENCE ABSTRACT)

Mathematical descriptions of physical processes are often in the form of systems of differential equations describing the time-evolution of a phenomenon. Computer simulations are realizations of these equations using well-known discretization schemes. Numerical time-stepping methods allow us to advance the state of a computer model using a sequence of time-steps. This thesis investigates current methods in time-stepping schemes, identifying a number of additional features needed to improve the speed and accuracy of simulations, and devises new methods suitable for large-scale applications where multiple processes of different physical nature drive the equations, potentially with significant differences in their time-scales. Various families of new methods are introduced with proper mathematical formulations provided for creating new ones on demand. The accuracy and stability of the methods are investigated using standard analysis techniques. These methods are then used in numerical experiments to investigate their abilities.

Dedication

To my parents.

Acknowledgments

I would like to express my gratitude to my academic advisor, Dr. Adrian Sandu, for the opportuninty to work with him in the Computational Science Lab. This work would not be possible without his profound guidance and deep knowledge of the advanced time-stepping research. Also, the commitee members have provided me with valuable feedback about the research presented here, I am grateful for their input. Many thanks to Dr. Steven Roberts for his countless suggestions for improvements and careful critique of this thesis. Other members of the Computational Science Lab, both current and past, have helped me on numerous occations, I am very thankful of them.

I would also like to express my appreciation to my friends and colleagues who have supported me in many ways over the years I have spent in Blacksburg. Special thanks to Drs. A. Nelson, A. Azizi, A. Behzad, H. Parsian, A. Chenault, and S.E.A.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Simulation of physical systems is an essential component of many engineering and scientific fields. In many applications, numerical simulations of dynamical systems are used for design verification and as a means to test ideas in different scenarios without paying the high costs of prototyping. Simulations also generate predictions that inform domain scientists about the state of a complex natural phenomenon by finding solutions to a mathematical model constructed to represent the true dynamics. Forward integration is also an important step in solving parameter estimation, optimization, and inverse problems.

This work is focused on developing new and efficient numerical methods for the solution of Ordinary differential Equations (ODEs) arising from multi-physics problems. In many cases the ODE system is a result of discretization methods applied to the differential equations governing numerous physical processes. The resulting ODE system is large and integrating it in time to find the solution at future times is a demanding task: Ideally a numerical scheme offers high accuracy and low computational cost and robustness for a wide breadth of applications. Traditional methods don't provide all these features. While explicit methods are computationally low-cost, they are limited by their small time-steps due to stability issues. On the other hand, implicit methods are more stable but costly due to Newton iterations. Furthermore, solving Newton iterations requires information about the state-derivative of the system in the form of the Jacobian matrix, that may not be available. In other cases an inexact but cheap Jacobian may be at hand, but many traditional methods can not effectively use this approximate.

Classical Runge-Kutta methods, historically favored for solving initial value problems (IVPs) suffer from a number deficiencies when applied to large systems:

- Creating high order methods is very challenging due to increasing and complicated order conditions.
- The methods encounter order reduction when the ODE is stiff or when applied to PDEs with time dependent or in-homogeneous boundary conditions.
- For complicated systems where the exact Jacobian is not available, working with approximate Jacobians reduces the numerical order of convergence.

Instead of black-box treatment of time-dependent ODEs, we set out to design time-stepping methods that can provide better integration with multi-physics and multi-scale systems

by treating partitions of the system with appropriate methods. For example, linear and nonlinear, or fast and slow processes can be identified and integrated accordingly. We will also explore methods that can balance implicit treatment with efficiency while guaranteeing accuracy for stiff problems.

In chapter 2 we will survey traditional Runge-Kutta methods and their application on large systems with focus on computational efficiency of stiff problems. We will identify a number of deficiencies with these methods that need to be addressed. We consider matrix-free implementations, a popular approach for time-stepping methods applied to large CFD applications due to its adherence to scalable matrix-vector operations and a small memory footprint. We compare explicit methods with matrix-free implementations of implicit, linearly-implicit, as well as "lightly-implicit" Rosenbrock-Krylov methods. Based on the results of this chapter, we set out to create new families of time-stepping methods improving one or more of the issues we identified in traditional schemes.

Chapter 3 discusses a new approach to designing highly accurate multirate methods for multiphysics problems. Here, the assumption is that the system contains multitudes of natural time-scales differing significantly from each other, therefore, integrating partitions of the system at different rates is computationally favorable. In essence, multirate methods apply different step sizes to resolve different components of the system based on the local activity levels. While the multirate idea has been around for decades, construction of multirate schemes at high orders has not been studied profoundly due to difficult order conditions. The focus of this chapter is on the design of practical high-order multirate methods using the theoretical framework of generalized additive Runge–Kutta (MrGARK) methods. MrGARK schemes of up to order four will be presented that are explicit-explicit (both the fast and slow component are treated explicitly), implicit-explicit, and explicit-implicit. The variety of the available methods is important in broadening their application in different fields when the user has insight about the underlying physics involved.

In chapter 4 we will discuss the benefits of working with General Linear Methods (GLMs). Specifically, we will show how to use GLMs for Alternating Directions Implicit (ADI) integration on parabolic problems. ADI integration is an operator splitting approach to solve parabolic and elliptic partial differential equations in multiple dimensions. Instead of solving high-dimensional Newton iterations, ADI schemes solve a sequence of related low-dimensional equations. Classical ADI methods have order at most two, due to the splitting errors. Moreover, when the time discretization of stiff one-dimensional problems is based on Runge-Kutta schemes, additional order reduction may occur. In chapter 4 a new ADI scheme based on the partitioned General Linear Methods framework is proposed. This approach allows the construction of high order ADI methods. Moreover, due to their high stage order, the proposed methods can alleviate the order reduction phenomenon seen in other methods. Numerical experiments are shown to provide further insight into the accuracy, stability, and application of these new methods.

Chapter 5 proposes a new GLM framework for linearly implicit methods. Let us consider

the benefits of having such a framework: Linearly implicit methods are computationally more cost-effective and we can create methods that allow arbitrary Jacobian approximations in computing the stages. Unlike linearly implicit Runge-Kutta methods, we show that an elegant order conditions theory extending to high orders is available here. Multiple new families of methods are created in this framework and numerically tested on stiff test problems and ODE systems with sophisticated discretizations where calculating the exact Jacobian is not feasible. The behavior of the global error of these methods when applied to DAEs and singularly perturbed equations is also analyzed to show theoretical convergence rates in the stiff case.

Finally, chapter 6 provides a summary of the work presented in thesis. Main outcomes of each chapter are identified highlighting the achievements and results from numerical experiments.

Chapter 2

A survey of Runge–Kutta methods for matrix-free integration

2.1 Introduction

Explicit time integration methods have been used for time-accurate solutions of unsteady flow problems due to their low computational cost per step and moderate memory requirements. For example, in [66] a number of embedded high-order explicit Runge-Kutta methods with minimal memory storage have been developed for the compressible Navier-Stokes equations based on van der Houwen's technique [126] for stage memory storage reduction. However, stability constraints restrict the maximum time steps that explicit methods can employ. Sengupta et al. [111, 114] show that explicit Runge-Kutta methods restricted to small time-steps can better predict physical instabilities of a flow problem than multistep Adams-Bashforth methods. Bhaumik et al. [10] investigate numerical dispersion and phase-shift errors associated with Runge-Kutta methods and different spatial discretizations. These issues are addressed in [85, 110, 112, 113] by developing a class of explicit Runge–Kutta methods that, when coupled with accurate spatial discretization schemes, optimize wave properties in advection-dominated problems.

Due to their better stability, implicit time-stepping methods can use very large time steps, however, the computational cost per step is also greater. The overall computational efficiency of a method is determined by the trade off between the computational cost per step and the total number of steps required to carry out the simulation.

The main cost of implicit methods is associated with solving a large system of nonlinear equations at each step. Newton type methods for the solution of nonlinear systems are commonly used in the literature in conjunction with preconditioned Krylov-based solvers for the inherent linear systems. The popular Jacobian-free Newton Krylov (JFNK) methods employ finite difference approximations of the Jacobian-vector products required by Krylov solvers [29]. Studies of JFNK methods applied to Navier-Stokes equations [83] have shown that error tolerances of Krylov space solvers need to be carefully optimized for performance and accuracy. Interested readers may also refer to [8] for detailed experiments quantifying the effects of Krylov solver tolerance on the convergence and efficiency of high order Rosenbrock methods in CFD applications. Furthermore, Development of matrix-free space-time implicit methods for DG discretizations are reported in [67, 120, 121].

There is considerable interest in developing numerical schemes that provide a suitable level of implicitness for time integration of stiff problems, such as to allow relatively large time steps while keeping the cost per time step comparable to that of explicit methods.

This chapter studies the efficiency of matrix-free time stepping schemes. The remaining part of this chapter is structured as follows. Section 2.2 reviews numerical methods accessible for the time integration of large systems of ordinary differential equations. The numerical methods investigated here and their implementation are presented in Section 2.3. Section 2.4 applies these methods to a number of test problems and studies their effectiveness in terms of their numerical accuracy, stability, and computational efficiency in case of high dimensional problems. Conclusions and future work directions are discussed in Section 2.5.

2.2 Runge-Kutta time integration methods

Consider the autonomous initial value problem:

$$\frac{dy}{dt} = f(y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq t_F, \quad y(t) \in \mathbb{R}^N, \quad f : \mathbb{R}^N \to \mathbb{R}^N. \tag{2.1}$$

In this chapter equation (2.1) represents the system of ODEs resulting from the spatial semi-discretization of the governing equations in the method-of-lines framework. The system is considered autonomous without loss of generality: any system can be written in autonomous form by appending the time variable to the solution vector. With only time derivatives remaining in equation (2.1), it is the choice of time-stepping method that determines the stability, accuracy, and efficiency of the numerical solution as the solution is propagated in time.

We next review several important classes of numerical time integration algorithms.

2.2.1 Runge-Kutta methods

The historically well-known time integration schemes attributed to Runge and Kutta are well-studied [17, 37] and extensively utilized in flow applications [62, 65]. Let $y_n \approx y(t_n)$ be a numerical approximation of the solution of the system (2.1). An s-stage Runge-Kutta method(advances the numerical solution to the next time step $t_{n+1} = t_n + h$ as follows:

$$k_i = f\left(y_n + h \sum_{j=1}^{s} a_{i,j}\, k_j\right), \quad i = 1, \ldots, s; \tag{2.2a}$$

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j\, k_j. \tag{2.2b}$$

The method coefficients

$$a = [a_{i,j}]_{1\le i,j\le s} \quad b = [b_i]_{1\le i\le s} \quad c = [c_i]_{1\le i\le s},$$

are determined such that the method (2.2) has the desired accuracy and stability properties [52, II.1]. If the right hand side function f has a time argument such that $y' = f(t, y)$ the the stages are evaluated as:

$$k_i = f\left(t_{n-1} + c_i h,\, y_n + h\sum_{j=1}^{s} a_{i,j}\, k_j\right), \quad i = 1, \dots, s \tag{2.3}$$

Explicit Runge-Kutta (ERK) methods are characterized by coefficients $a_{i,j} = 0$ for any $j \le i$. This means that each stage value k_i (2.2a) depends only on previously stage vectors $k_1, \dots, k_{i-1}$. This leads to the convenient result that explicit Runge-Kutta methods need only one ODE right-hand-side function evaluation per stage, and no linear or nonlinear systems of equations are solved in the process. The stability requirements due to CFL conditions limit the step size h, and therefore impact the efficiency of the method.

Singly Diagonally Implicit Runge-Kutta methods (SDIRK) [51, IV.6] are characterized by coefficients $a_{i,j} = 0$ for any $j < i$, and $a_{i,i} = \gamma > 0$ for all stages $i = 1, \dots, s$. Solving for the stage vector k_i requires the solution of a nonlinear system of equations at each stage

$$F_i(k_i) = k_i - f\left(\xi_i + h\gamma k_i\right) = 0 \quad \text{for} \quad i = 1, \dots, s, \tag{2.4}$$

which makes the computational cost per step significantly larger than for ERK. However, this also leads to improved stability properties and the ability to use much larger time steps. The nonlinear equation (2.4) is solved using Newton-type iterations:

$$\Delta k_i^{\{\ell\}} = -\left(\frac{\partial F_i}{\partial k_i}\right)^{-1} F_i\left(k_i^{\{\ell\}}\right), \quad k_i^{\{\ell+1\}} = k_i^{\{\ell\}} + \Delta k_i^{\{\ell\}}, \quad \ell = 0, 1, \dots \tag{2.5}$$

where

$$\frac{\partial F_i}{\partial k_i} = \mathbf{I}_N - h\,\gamma\,\mathbf{J}_n, \tag{2.6}$$

and $\mathbf{J}_n$ is the Jacobian of the ODE right-hand-side function:

$$\mathbf{J}_n = \left.\frac{\partial f(y)}{\partial y}\right|_{y=y_n}. \tag{2.7}$$

The fact that $a_{i,i} = \gamma$ for all stages allows re-using the LU decomposition of (2.6) in the solution of linear systems appearing in equation (2.5) for all stage vectors $i = 1, \dots, s$.

2.2.2 Rosenbrock methods

Linearly implicit methods avoid the nonlinear systems (2.5) and solve only linear systems at each stage. One step of a Rosenbrock (ROS) method [51, IV.7] reads:

$$Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j}\, k_j, \quad k_i = f\Big(Y_i\Big) + h\, \mathbf{J}_n \sum_{j=1}^{i} \gamma_{i,j}\, k_j, \quad i = 1, \ldots, s; \tag{2.8a}$$

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_j k_j. \tag{2.8b}$$

Therefore, the stage vectors k_i are found in succession by solving linear systems of the form:

$$\left(\mathbf{I}_N - h\, \gamma_{i,i}\, \mathbf{J}_n\right)\, k_i = f\left(Y_i\right) + h\, \mathbf{J}_n \sum_{j=1}^{i-1} \gamma_{i,j} k_j, \qquad i = 1, \ldots, s. \tag{2.9}$$

As with SDIRK methods, choosing $\gamma_{i,i} = \gamma > 0$ for $i = 1, \ldots, s$ helps reduce the computational costs by allowing to reuse the same LU factorization (2.6) for all stages.

Rosenbrock-Wanner methods

Rosenbrock methods simplify the computational effort necessary to solve the stage vector equations by limiting the implicitness to linear terms containing the exact Jacobian right-hand-side products [75]. As a consequence, the accuracy of the method depends on the availability of the exact Jacobian. In many practical cases an exact Jacobian is difficult to compute, however some approximation of the Jacobian may be available at reasonable computational cost. The Rosenbrock-Wanner (ROW) methods are Rosenbrock schemes that retain the order of accuracy for any matrix $\mathbf{A}_n$ used in place of the exact Jacobian $\mathbf{J}_n$ in (2.8). The preservation of accuracy is possible by imposing additional order conditions on the method coefficients [87]. Better approximations of the Jacobian $\mathbf{A}_n \approx \mathbf{J}_n$ will ensure better numerical stability. We note that while the formal definition for a ROW method is the same as in equation (2.8), the method coefficients are different due to the additional order conditions.

Rosenbrock-Krylov methods

The stage vectors in Rosenbrock-type methods are computed by solving the linear system of dimension N in equation (2.9). For large problems the solutions of these linear systems is best obtained via a Krylov-space iterative linear algebra solver such as `GMRES` [92].

Instead of using a Krylov-based iterative solver such as `GMRES`, Rosenbrock-Krylov (ROK) methods developed in [122] reformulate the method (2.8) using implicitness only in the

Krylov subspace of dimension M constructed using modified Arnoldi iteration [92]

$$\mathcal{K}_M\big(\mathbf{J}_n, f(y_n)\big) = \text{range}\{\mathbf{V_n}\}, \quad \mathbf{V_n} \in \mathbb{R}^{N\times M}, \quad \mathbf{V_n}^T\,\mathbf{V_n} = \mathbf{I}_M, \quad \mathbf{V_n}^T\,\mathbf{J}_n\,\mathbf{V_n} = \mathbf{H_n}.$$

Here $\mathbf{H}$ and $\mathbf{V}$ are the upper Hessenberg and the orthogonal basis of the Krylov space, respectively, and are results of Arnoldi process.

A single time step of a ROK method is constructed as follows [122]:

$$F_i = f\left(y_n + \sum_{j=1}^{i-1} \alpha_{i,j} k_j\right), \tag{2.10a}$$

$$\phi_i = \mathbf{V_n}^T\,F_i, \tag{2.10b}$$

$$\lambda_i = (\mathbf{I}_M - h\,\gamma\,\mathbf{H_n})^{-1}\left(h\,\phi_i + h\,\mathbf{H_n}\sum_{j=1}^{i-1}\gamma_{i,j}\,\lambda_j\right), \tag{2.10c}$$

$$k_i = \mathbf{V_n}\,\lambda_i + h\,(F_i - \mathbf{V_n}\,\phi_i), \tag{2.10d}$$

$$y_{n+1} = y_n + \sum_{i=1}^{s} b_i\,k_i. \tag{2.10e}$$

For $M \ll N$ the linear system (2.10c) is easily solvable using direct methods, and the stage vectors in full space can be recovered projecting the reduced space stage values back to full space[122]. The minimum dimension of the Krylov space is determined by the desired order of the numerical scheme. Readers interested in the derivation of the order conditions for this method may refer to [122]. In the extreme case $M = 0$ the method (2.10) reduces to an explicit Runge-Kutta method. In practice, however, we need the Krylov space to be large enough to capture some of the dominant eigenvalues of the full Jacobian, corresponding to fast-changing modes of (2.1), such as to alleviate the restrictions on step size of explicit methods imposed by the stiffness of the problem. An important question is whether the additional computational cost required by Arnoldi iteration can be compensated by the increases in step size due to the implicit nature of the method. The numerical results in Section 2.3 provide answers to this question.

2.2.3 Matrix-free implementations

As discussed in Sections 2.2.1 and 2.2.2, implicit time-stepping methods use the Jacobian matrix (2.7) to solve linear or nonlinear systems of equations at each step. The dimension of the state vector in (2.1) may become significantly large when a highly refined spatial discretization is required, whether to capture fine details of flow in CFD applications or in large data-driven models such as climate research. Computation and storage of Jacobian matrices, even in sparse form, is not favorable in such scenarios.

In some cases the complexity of the spatial discretization scheme impedes construction of analytic Jacobian matrices. In CFD applications the use of complex upwind flux schemes

including Roe's and Van Leer's flux scheme in addition to MUSCL reconstruction with flux limiters make the generation of an analytic Jacobian challenging. The framework of matrix-free methods allows us to exploit the benefits of advanced time-stepping methods without forming the Jacobian matrix directly.

Krylov space iterative methods for solving the stage equations (2.5) or (2.9) rely on computing Jacobian-vector products. Instead of computing the Jacobian matrix and then multiplying, it is possible to approximate directly Jacobian-vector products using the finite-difference approximation of a directional derivative:

$$\mathbf{J}_n \cdot v = \left.\frac{\partial f(y)}{\partial y}\right|_{y_n} \cdot v \approx \frac{f(y_n + \varepsilon v) - f(y_n)}{\varepsilon}. \tag{2.11}$$

The optimum ε is chosen considering the trade-off between truncation and round-off errors [68]. Higher-order approximations are not favorable here as they require more right-hand-side function evaluations, which, considering the large dimensions of the problem, are costly to compute.

A more in-depth analysis of different strategies to compute Jacobian-vector products, and their effects on convergence and efficiency of implicit time integration methods, can be found in [125]. Of special interest is using exact Jacobian-vector products instead of finite difference approximations. Numerical experiments applied to discretizations of PDEs in [125] indicate that exact Jacobian-vector products provide more robustness in observed convergence orders, and increased runtime efficiency over methods using approximate products (2.11).

2.3 The software infrastructure for numerical investigations

Testing the various time-stepping schemes introduced in chapter 1 in matrix-free mode will provide interesting results about the performance of each method as well some insight on the features of successful time-stepping methods for large systems. In the following section the software used for numerical experiments is described.

2.3.1 Time-stepping schemes and their implementation

The time integration software used in the numerical experiments is MATLODE [101], a Matlab library for integration of ODE systems including implicit and explicit Runge-Kutta methods, Rosenbrock methods and Krylov based methods. The package also supports forward, adjoint, and tangent linear models, enabling sensitivity analysis applications. Aside from the methods available in MATLODE package, Matlab's explicit time-stepping scheme based on Dormand and

Prince [31] is also included in tests for comparison. Table 2.1 summarizes the methods used in numerical experiments and their properties.

Table 2.1: Overview of time stepping methods used in numerical experiments.

Method	Family	Stages	Order	Stability
ERK	Explicit Runge-Kutta	5	4	Conditionally stable
DOPRI5	Explicit Runge-Kutta	7	5	Conditionally stable
DOPRI853	Explicit Runge-Kutta	12	8	Conditionally stable
SDIRK	Implicit Runge-Kutta	5	4	L-stable
ROS4	Rosenbrock	4	4	L-stable
ROW	Rosenbrock-W	4	3	L-stable
ROK	Rosenbrock-Krylov	5	4	Conditionally stable
ODE45	Explicit Runge-Kutta	5	4	Conditionally stable

2.3.2 The fluid flow simulation code

SENSE-Lite, the CFD code employed in these experiments, can solve both the Euler and Navier-Stokes equations for compressible flow [30]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_1}{\partial x_1} + \frac{\partial \rho v_2}{\partial x_2} = 0, \tag{2.12a}$$

$$\frac{\partial \rho v_1}{\partial t} + \frac{\partial \rho v_1^2}{\partial x_1} + \frac{\partial \rho v_1 v_2}{\partial x_2} + \frac{\partial p}{\partial x_1} = \frac{\partial \tau_{11}}{\partial x_1} + \frac{\partial \tau_{12}}{\partial x_2}, \tag{2.12b}$$

$$\frac{\partial \rho v_2}{\partial t} + \frac{\partial \rho v_1 v_2}{\partial x_1} + \frac{\partial \rho v_2^2}{\partial x_2} + \frac{\partial p}{\partial x_2} = \frac{\partial \tau_{12}}{\partial x_1} + \frac{\partial \tau_{22}}{\partial x_2}, \tag{2.12c}$$

$$\begin{aligned} &\frac{\partial E_t}{\partial t} + \frac{\partial \rho v_1 E_t}{\partial x_1} + \frac{\partial \rho v_2 E_t}{\partial x_2} + \frac{\partial v_1 p}{\partial x_1} + \frac{\partial v_2 p}{\partial x_2} = \\ &k\left(\frac{\partial q_1}{\partial x_1} + \frac{\partial q_2}{\partial x_2}\right) + \mu\left(\frac{\partial}{\partial x_1}(v_1\tau_{11} + v_2\tau_{12}) + \frac{\partial}{\partial x_2}(v_1\tau_{12} + v_2\tau_{22})\right), \end{aligned} \tag{2.12d}$$

where

$$q_i = -k\,\frac{\partial T}{\partial x_i}, \tag{2.12e}$$

$$\tau_{i,j} = \mu\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) - \frac{2}{3}\mu\,(\nabla \cdot v)\,\delta_{i,j}. \tag{2.12f}$$

SENSEI-Lite uses a curvilinear, structured-grid, finite volume method. Second-order spatial accuracy is achieved through a standard total variation diminishing scheme consisting of MUSCL reconstruction and selectable flux limiters [127]. The code is written in C++ and

MEX interfaces are used to call it from within time integrators implemented in Matlab. The primary function of the MEX code is to return the spatial residual for a given solution state; this is returned as a vector that is independent of any temporal information and can be used for building arbitrary time integration methods such as multi-stage ERK. This residual vector can optionally be returned as multiple vectors split according to the underlying equations, e.g., viscous and inviscid contributions from the Navier-Stokes equations as shown in (2.12); these vectors sum to the full residual and can also be used to integrate the equations independently. The Matlab client code is responsible for storing multiple solution state and update vectors as necessary, and can apply time-dependent source terms if desired.

2.4 Numerical experiments with a fluids code

2.4.1 Experimental setting

This section details the numerical experiments setup using MATLODE and SENSEI-Lite packages to study the performance of matrix-free time-stepping methods on unsteady flow problems. A reference solution is computed and stored by integrating the model using an explicit method with tight tolerances ($\sim 10^{-9}$) on errors. In each numerical experiment, the solution at the final integration time is compared against this reference solution, and the error is measured using the normalized $\mathcal{L}_2$ norm:

$$Error = \|y_N - y_{\text{ref}}\|_2 \qquad \text{where} \qquad y = [\rho, \rho v, \rho u, E_t]^T$$

Where y_N and y_{ref} are numerical and reference state vectors at the final integration time respectively. The error controller for adaptive time-step selection uses an embedded method to estimate an scaled scalar error based on the tolerances `AbsTol = RelTol` requested in the experiment. A detailed discussion of the error controller implementation can be found in [52, p. 168]

All experiments use matrix-free time-stepping methods. For each test problem we evaluate the eigenvalues of the Jacobian matrix in equation (2.7) to get information about the dynamic modes of the state variable evolution. A wide spread of the eigenvalues indicates the existence of both slow and fast dynamics, in other words, of "stiff" dynamics. The largest eigenvalue gives an estimate of the largest stable step-size in explicit methods. Eigenvalues are computed using Matlab's implementation of the implicitly restarted Arnoldi method that estimates the largest 1000 eigenvalues in magnitude for each test problem; this computation is also performed in matrix-free form.

2.4.2 Vortex-shedding cylinder test problem

The vortex-shedding cylinder test problem consists of a two-dimensional circular cylinder in a low subsonic flow (Mach 0.1) using a gas model for air at 5,000 ft altitude standard atmospheric conditions (278K, 84.31kPa). The viscosity for air is calculated based on local flow conditions using Sutherland's law. The model is a free-stream flow, using far-field boundary conditions set at over 100 chord lengths away from the surface of the cylinder to minimize interactions with the boundary. The default cylinder diameter is $8 \times 10^{-5}\, m$, which yields a Reynolds number of approximately 200 and results in a steady and predictable two-dimensional shedding of alternating vortices behind the cylinder. At this low Reynolds number there are no sub-grid-scale turbulence effects, so all physically accurate spatial and temporal scales in the solution can be directly modeled; therefore, the CFD code can solve the laminar Navier-Stokes equations with no underlying turbulence model. For all test problems, the flux scheme used is Roe's approximate Riemann solver and no flux limiter is employed in the MUSCL scheme as flux limiters were observed to be unnecessary and to generally reduce the accuracy of the FVM reconstruction in the continuous and smooth flow field around the cylinder. The parameters of this problem are modified to provide different tests, as summarized in Table 2.2. Care is taken so that the qualitative solution behavior (and appropriate Reynolds number) is maintained for all tests.

Table 2.2: Parameters of the vortex-shedding cylinder test problems.

Parameter	Description	Test problem 1 value	Test problem 2 value
$\rho(kg/m^3)$	Reference fluid density	1.0565	1.0565
S	Sutherland's coefficient	1.45E-6	2.9E-6
$T(K)$	Temperature	278	278
$v(m/s)$	Reference velocity	340	340
$L(m)$	Diameter	8E-5	8E-5
Re	Reynold's number	165.90	82.95

The first experiment is performed on the vortex-shedding cylinder test problem 1 with parameters given in Table 2.2. Figure 2.1 illustrates snapshots of the density component of the flow for cylinder test problem 1 at different times, showing the cyclic development of vortices behind the cylindrical object. This experiment uses fixed step sizes to study the temporal orders of convergence for each method, and the results are plotted in Figure 2.2. Numerical orders of convergence calculated for each method are reported in Table 2.3. One notable observation is the significantly lower numerical order of convergence for SDIRK method as a result of poor convergence of the Newton iteration, especially in the absence of preconditioners for the solution of linear systems. Readers interested in numerical experiments on a smaller problem that verify the theoretical order of convergence may consult the Appendix.

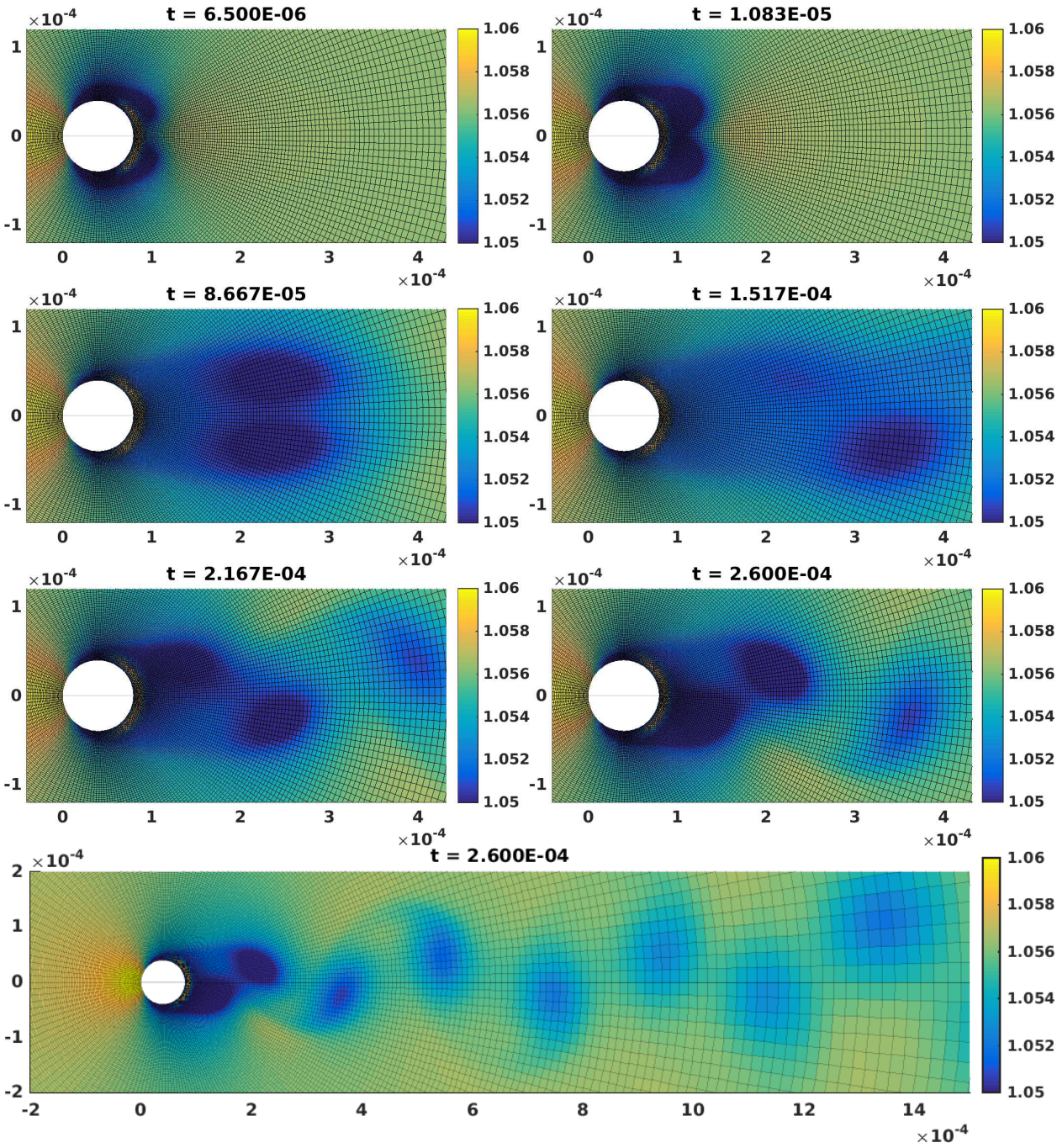


Figure 2.1: Snapshots of density component of the flow for vortex-shedding cylinder test problem 1 at different times.

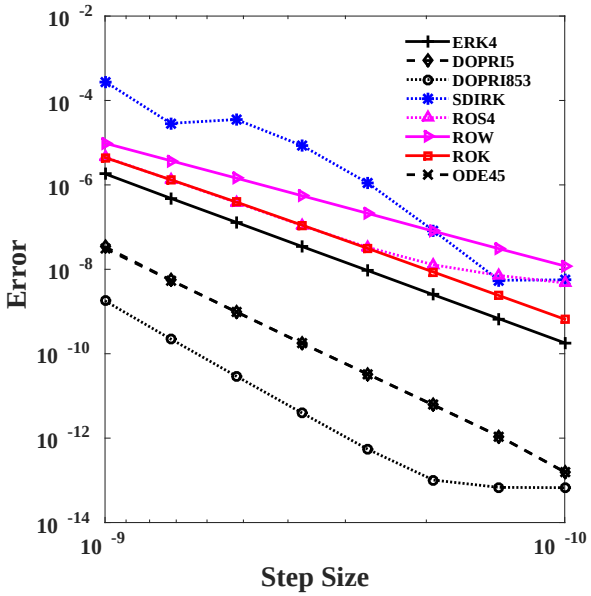


FIGURE 2.2: Convergence diagrams for the matrix-free methods of Table 2.1 using fixed step sizes over integration window $T = [0, 10^{-7}]$ second for cylinder test problem 1.

The cylinder test problem 2 with parameters given in Table 2.2, uses a different value for Sutherland's Law coefficient that translates into increased kinetic viscosity compared to vortex-shedding cylinder test problem 1. Figure 2.3 shows the numerical approximation of the first 1000 eigenvalues for the two test problems. The eigenvalues are computed using `Matlab`'s `eigs` function that uses an iterative Krylov-based iteration to approximate eigenvalues[118]. As indicated in Figure 2.3 the more viscous test problem shows larger negative eigenvalues, therefore, we expect stricter stability bounds on the step sizes for this test problem. For both problems the cluster of eigenvalues with the largest negative real parts consists of only a limited number of modes. This fact becomes relevant when we consider ROK methods that use a reduced order Jacobian for implicit integration. Throughout the numerical experiments the ROK method employs a reduced space of dimension four, unless otherwise specified.

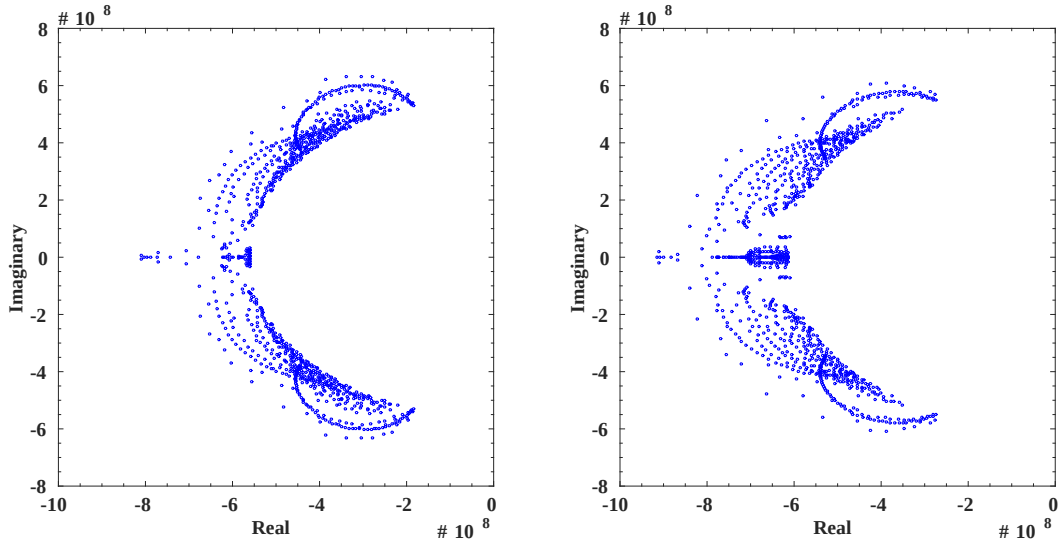
The stability of a method is related to its choice of timesteps in the adaptive time stepping framework. Figure 2.4 compares the step sizes of fourth order explicit Runge-Kutta method (ERK) to Rosenbrock-Krylov method of the same order for two different error tolerances. We can verify that the step sizes for the explicit method quickly reaches the upper bound set by the stability constraints regardless of the accuracy tolerance chosen for the method. On the other hand, by implicitly treating some of the stiff modes, the ROK method is able to achieve stable numerical integration for larger step sizes as is clear from Figures 2.4b and 2.4d. Furthermore, we notice that the step sizes scale well relative to the required accuracy.

Table 2.3: Numerical orders of convergence for various methods applied to the cylinder test problem 1.

Method	Numerical order	Theoretical order
ERK4	4.02	4
ERK5 (DOPRI5)	5.29	5
ERK5 (DOPRI853)	5.97	8
SDIRK	2.94	4
ROS4	3.11	4
ROW	2.96	3
ROK	3.85	4
ODE45	5.39	5

Figure 2.5 shows the Work-precision diagrams for the vortex-shedding cylinder test problems. Integration is performed over time window $T = [0, 2 \times 10^{-6}]$ seconds. These results lead to the following conclusions:

- As the adaptive time stepping method uses tighter tolerances, the integrator takes smaller steps leading to an increased number of total steps. This is the case for all integration methods presented here. However, the total number of steps for explicit methods does not change considerably for a wide range of tolerances, an effect observable in Figure 2.5 where the lines for explicit methods are nearly vertical. This is a result of the fact that for stiff problems the adaptive time steps are bounded by stability requirements rather than by accuracy constraints.
- Implicit methods are able to take fewer steps when the required solution tolerances are low, due to their improved stability properties. This is the case for ROS, ROW and SDIRK methods in Figure 2.5a and 2.5c. However, inspecting the runtime diagrams on Figures 2.5b and 2.5d reveals that the the increased cost of these methods makes them considerably less efficient.
- The effect of stiffness of the problem can also be seen in the timing reported in Figures 2.5b and 2.5d. We notice that while the explicit methods take about the same amount of time to complete the integration for all choices of solution tolerance, the runtime scales better with tolerances for implicit methods.
- ROK is the most efficient method for the cylinder test problems for error tolerances below 10^{-6}. This is an indication that ROK is able to capture sufficiently many stiff components in its Krylov subspace, and that by treating them implicitly the method is able to take larger time steps.



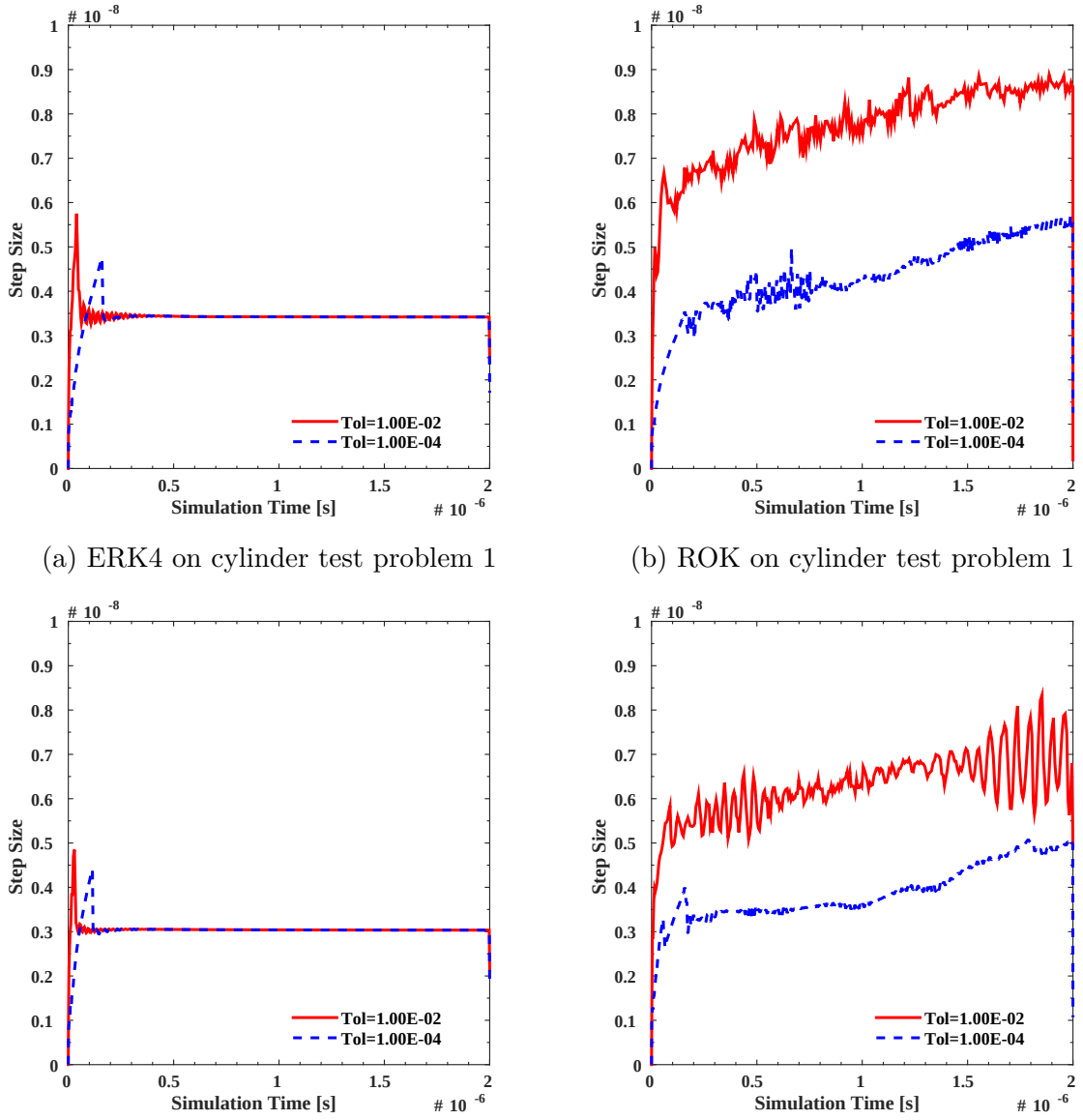
(a) Cylinder test problem 1 (b) Cylinder test problem 2

FIGURE 2.3: Eigenvalue distribution of the Jacobians of the right-hand-side functions for vortex-shedding cylinder test problems.

2.4.3 Flow over NACA0012 wing test problem

An analysis of unsteady flow over a NACA0012 airfoil is performed in addition to the cylinder test problem. Similar free-stream flow conditions are applied to this test problem, using atmospheric conditions at 5,000 ft. The Mach number is increased to 0.25; however, this flow is still within the subsonic flow regime. The chord length of this symmetric airfoil geometry is set to 0.001 meters, producing flow with a Reynolds number close to 5,000. Experimental analysis suggests that for the NACA0012 airfoil, a free-stream flow at this Reynolds number produces predominantly laminar flow over the airfoil and in its wake [55], allowing for turbulent effects to be neglected in this analysis. At an angle of attack of 15 degrees, vortices shed into the wake due to laminar separation of the flow over the upper surface of the airfoil generating an unsteady flow solution. As with the cylinder case, Roe's flux scheme is used with no flux limiter.

Figure 2.6 shows snapshots of the density field at different time moments. The distribution of Jacobian eigenvalues is shown in Figure 2.7, and indicates that a large number of fast eigenvalues are clustered together. The performance diagrams in Figure 2.8 indicate that the fastest methods here are Rosenbrock and ROW methods, until ROK and SDIRK become most efficient for errors below 10^{-4}. This test problem converges to a steady state solution, and this particular dynamics favors the implicit Rosenbrock methods, since the fast modes need only to be only damped out and not to be accurately solved.



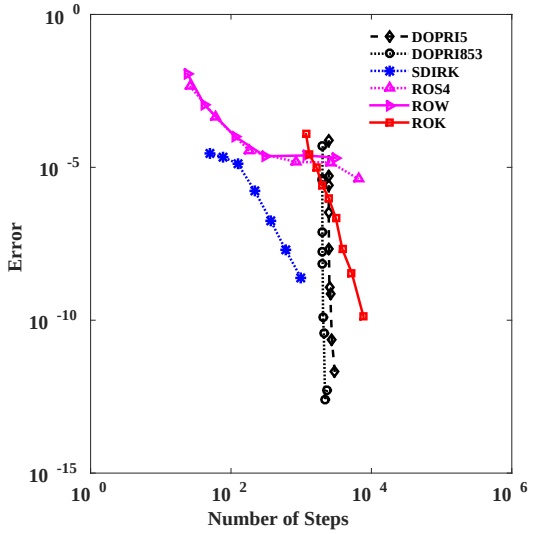
(a) ERK4 on cylinder test problem 1

(b) ROK on cylinder test problem 1

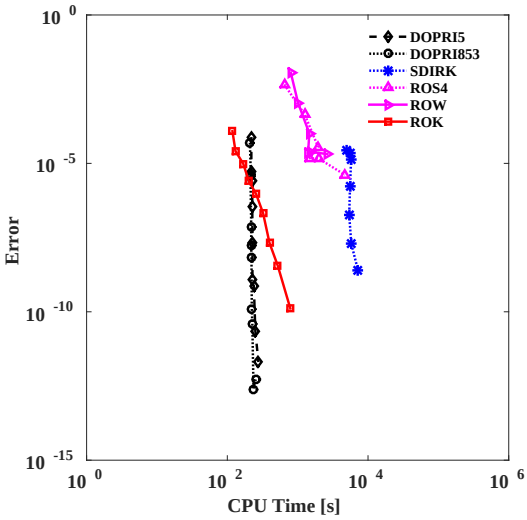
(c) ERK4 on cylinder test problem 2

(d) ROK on cylinder test problem 2

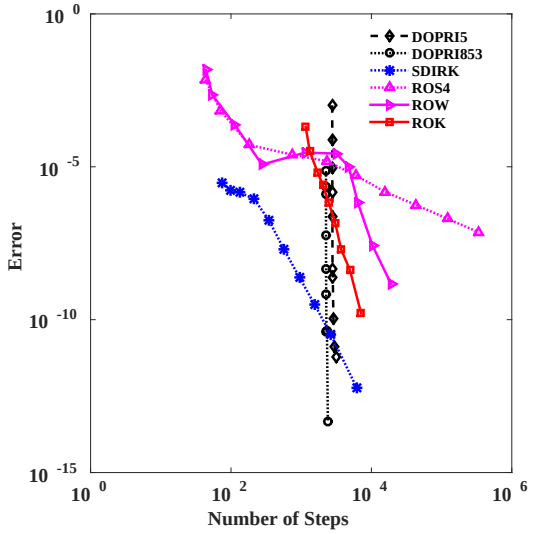
Figure 2.4: Adaptive time steps taken by the explicit Runge-Kutta method and by the ROK method for vortex shedding cylinder test problems.



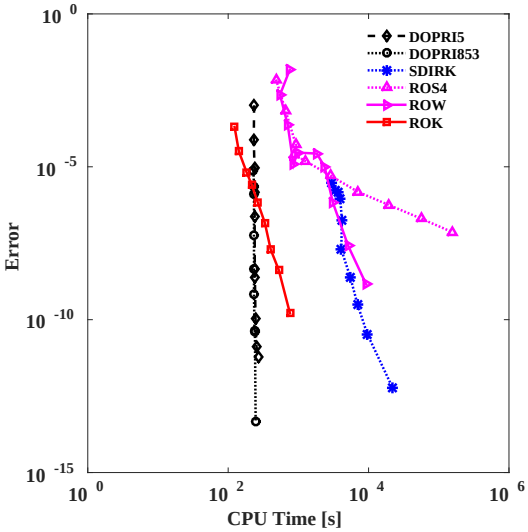
(a) Convergence diagram, cylinder test problem 1



(b) Work-precision diagram, cylinder test problem 1



(c) Convergence diagram, cylinder test problem 2



(d) Work-precision diagram, cylinder test problem 2

FIGURE 2.5: Relative performance of different integration methods applied to the cylinder test problems.

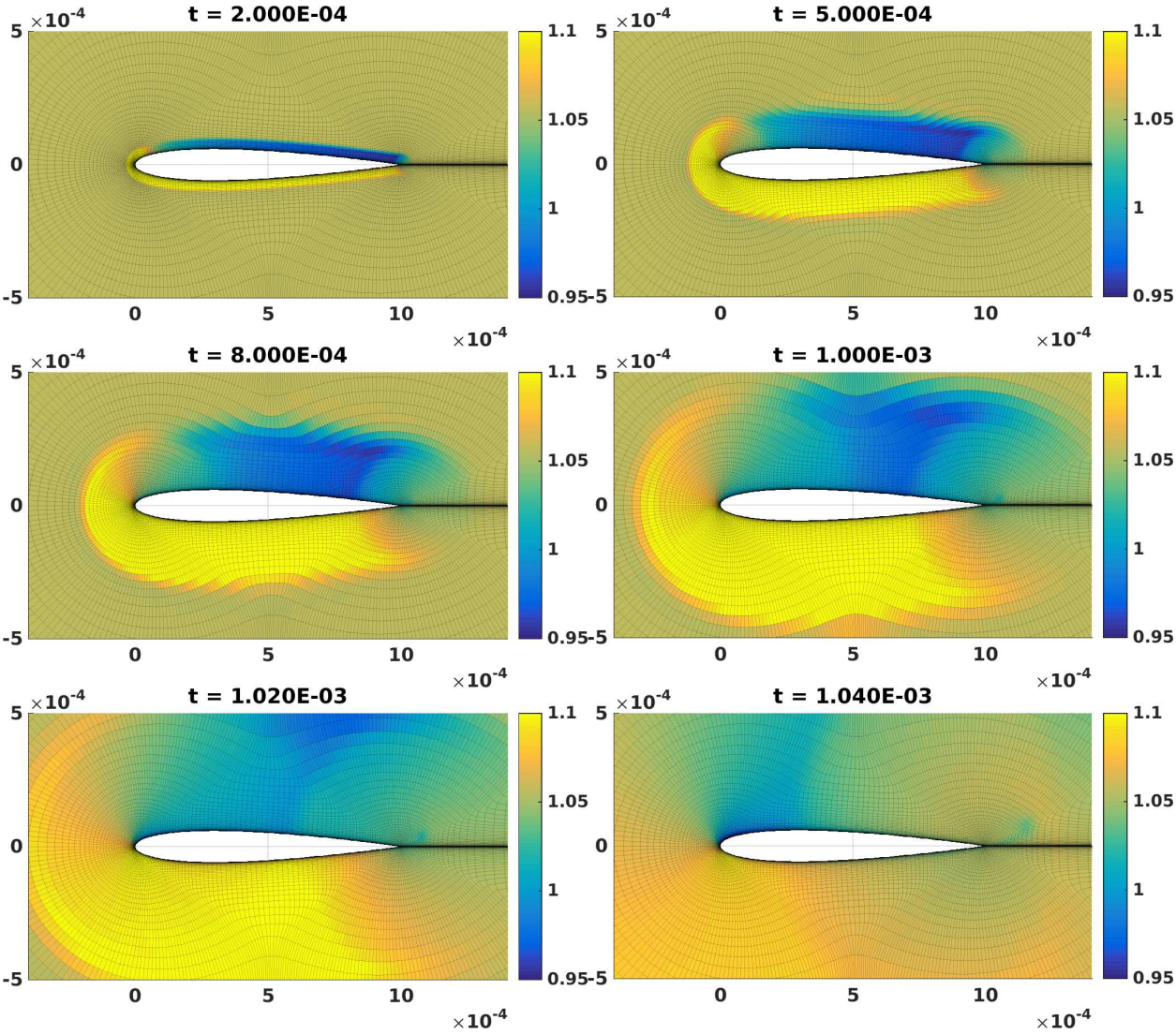


Figure 2.6: Snapshots of density component of the flow for NACA0012 at different times.

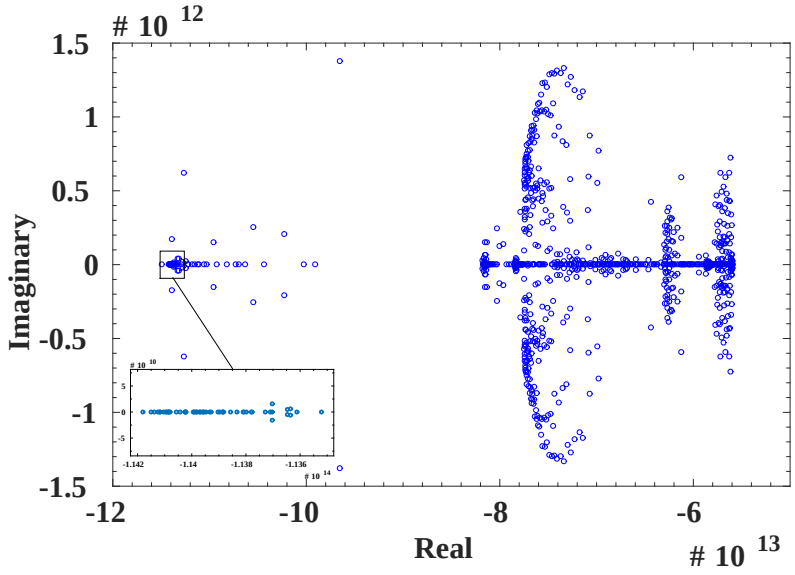
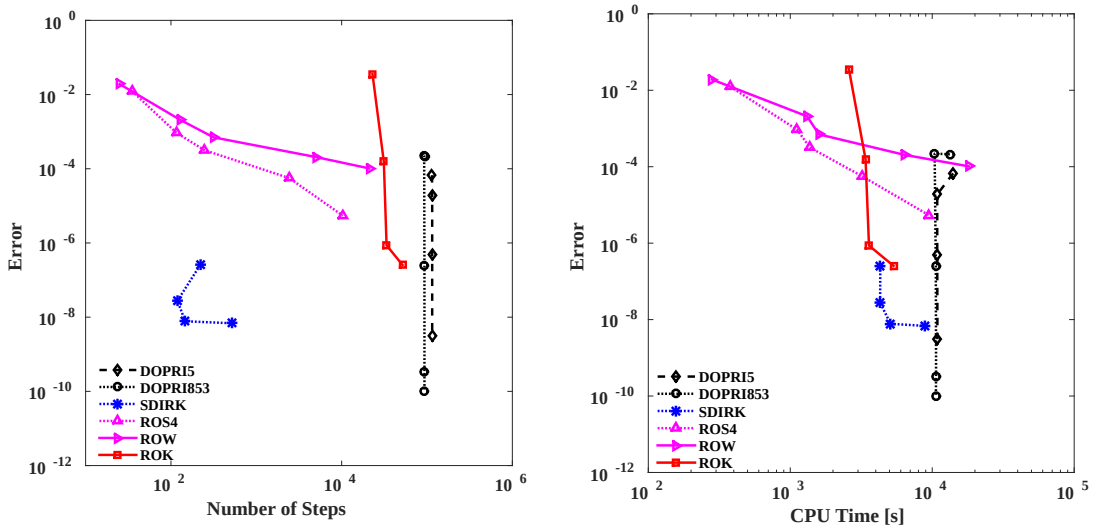


FIGURE 2.7: Eigenvalue distribution of the Jacobian for the NACA0012 test problem.



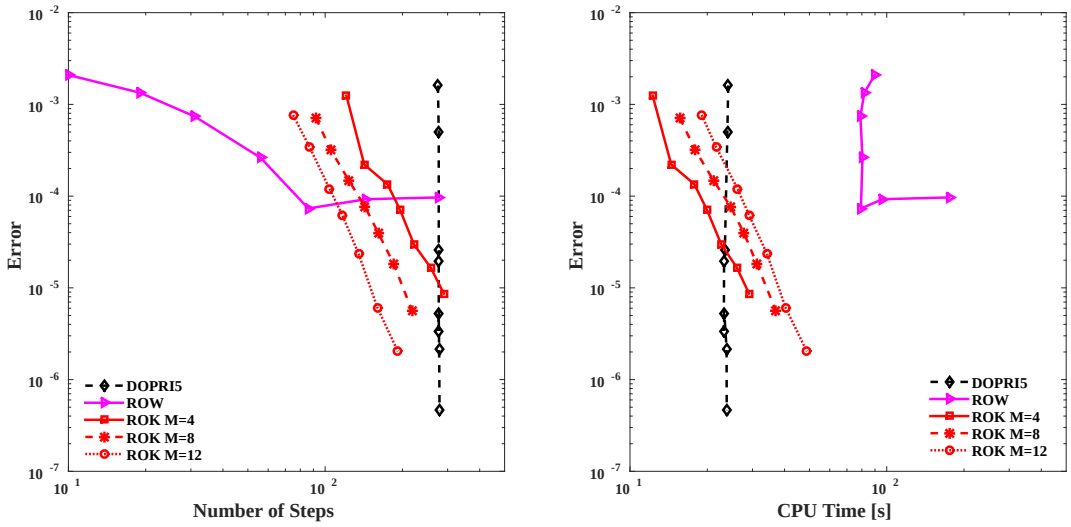
(a) Convergence diagram

(b) Work-precision diagram

FIGURE 2.8: Relative performance of different integration methods applied to the NACA0012 test problem.

2.4.4 Vortex shedding cylinder with iso-thermal conditions test problem

The final test problem uses the vortex shedding cylinder problem 2, with iso-thermal conditions at cylinder boundaries for a wall temperature of $33°K$. Among each class of explicit and implicit integrators we have selected the fastest representative methods. We have also chosen three variations of ROK method with 4, 8, and 12 Krylov basis vectors, respectively. The results in Figure 2.9 demonstrate that Krylov methods retain their computational superiority for a wide range of error tolerances. Furthermore, we observe that adding more basis vectors to the Krylov subspace requires extra cost (Figure 2.9b) and in turn increases step sizes slightly (Figure 2.9a), but ultimately the most efficient method is the one with minimum basis size, i.e. 4 vectors. Finally, it is worthwhile to point out that once the adaptive error controller of the step size reaches the GMRES tolerance, the error of JFNK methods does not decrease any further. This is seen in Figure 2.9 for the Rosenbrock-W method where the error curve flattens at an error level of about 5E-4.



(a) Convergence diagram (b) Work-precision diagram

Figure 2.9: Relative performance of different integration methods applied to cooled cylinder test problem with $T_{\text{wall}} = 33°K$.

2.5 Beyond traditional methods

In the following chapters we will investigate a number of strategies that allow us to take better advantage of both explicit and implicit methods when the physical system has suitable components for each type of these methods.

Chapter 3

Multirate GARK methods

3.1 Introduction

Many applications in science and engineering require the simulation of dynamical systems where different components evolve at different characteristic time scales. Clearly, these systems challenge traditional time discretizations that use a single time step for the entire system: either the fast components are resolved inaccurately, or the slow components are resolved with more accuracy than required, therefore increasing computational costs. Multirate methods discretize the fast components of a partitioned system using smaller time-steps and slow components with larger time-steps .

One can create multirate methods using traditional integrators along with different time steps for different components, where the coupling between these components is carefully constructed. Early efforts to develop multirate Runge-Kutta methods are due to Rice [88] and Andrus [5, 6]. In the first discussion of "multirate methods" Gear and Wells [39] propose pairing various linear multistep methods. This fundamental contribution already points to a number of challenges facing multirate methods such as coupling, automatic step size selection, and efficiency of the overall computational process. Other work to construct multirate linear multistep schemes includes [63]. Günther et al. [48, 50] developed multirate methods for partitioned Runge-Kutta schemes, as well as Rosenbrock-W methods [46] of order three that are well-suited for treatment of systems with both stiff and non-stiff variables. Similarly, Kværnø and Rentrop [69, 70] constructed explicit multirate Runge-Kutta methods of order three. Bartel et al. [7] propose one-step methods where internal stages are used to provide the coupling between the fast and slow components. Constantinescu and Sandu developed strong stability preserving (SSP) multirate methods of Runge-Kutta [26] and linear multistep [96] type that are suited for solving hyperbolic partial differential equations (PDEs).

Another approach, tracing back to Engstler et al. [36], derives multirate methods using Richardson extrapolation. Here, the solution is recursively improved on partitions of the system until required tolerances are satisfied. An important advantage of such schemes is the naturally available dynamic partitioning of the system. Constantinescu and Sandu [27, 28, 95] considered explicit and implicit base methods for multirate extrapolation methods and study their stability properties. Other multirate approaches include Galerkin [73], and combined multiscale [35] methodologies.

Günther and Sandu [47] built a class of multirate methods in based on the General Additive Runge–Kutta framework (GARK) [97]. Bremicker-Trübelhorn and Ortleb [13] developed third order multirate GARK (MrGARK) methods for fluid-structure interaction, and allowed for non-uniform fast steps in the order conditions of the methods.

This study develops a systematic design approach for constructing multirate methods in the MrGARK framework of Günther and Sandu [47, 97]. Several high-order schemes are constructed that combine implicit and explicit components for the fast and slow subsystems.

The chapter is organized as follows. Section 3.2 reviews the GARK framework and multirate GARK family. We study order conditions for high order MrGARK methods in section 3.3, and their scalar linear stability in section 3.4. Section 3.5 provides insight into fast-slow coupling requirements. Section 3.6 discusses the design criteria for practical methods, and section 3.7 studies error estimation and the adaptivity of both micro- and macro-steps. Newly constructed methods are listed in section 3.8, and method coefficients are detailed in section B.1. Numerical tests are performed on different test problems and the results are reported in section 3.9.

3.2 Multirate generalized additive Runge–Kutta schemes (MrGARK)

In this section we review some background on MrGARK methods.

3.2.1 GARK methods

The generalized additive Runge–Kutta (GARK) methods introduced in [49, 97] allow derivation of advanced multi-methods for solving *additively* partitioned systems of ordinary differential equations:

$$y' = f(y) = \sum_{m=1}^{N} f^{\{m\}}(y), \qquad y(t_0) = y_0, \tag{3.1}$$

where the right-hand side function $f : \mathbb{R}^d \to \mathbb{R}^d$ is split into N different partitions based on properties such as stiffness, non-linearity, dynamical behavior, and evaluation cost. We note that additive partitioning includes the case of *component partitioning* in which the state vector is split into a number subsets.

A GARK method advances the numerical solution as follows [97]:

$$Y_i^{\{q\}} = y_n + h \sum_{m=1}^{N} \sum_{j=1}^{s^{\{m\}}} a_{i,j}^{\{q,m\}} f^{\{m\}} \left(Y_j^{\{m\}} \right) \qquad \substack{q=1,\ldots,N,\\ i=1,\ldots,s^{\{q\}},} \tag{3.2a}$$

$$y_{n+1} = y_n + h \sum_{q=1}^{N} \sum_{i=1}^{s^{\{q\}}} b_i^{\{q\}} f^{\{q\}} \left(Y_i^{\{q\}} \right). \tag{3.2b}$$

We note that if the right hand side function accepts a time argument such as $y' = f(t, y)$ the stages evaluations are also done at consistent arguments $f^{\{q\}} \left(t_{n-1} + c_i^{\{q\}} h, Y_i^{\{q\}} \right)$.

3.2.2 Multirate GARK methods

In the case of a two-way partitioned system eq. (3.1) with slow component $\{\mathfrak{s}\}$, and fast component $\{\mathfrak{f}\}$ we have:

$$y' = f(y) = f^{\{\mathfrak{s}\}}(y) + f^{\{\mathfrak{f}\}}(y), \qquad y(t_0) = y_0. \tag{3.3}$$

A multirate GARK method [97] integrates the slow component with a Runge–Kutta method $(A^{\{\mathfrak{s},\mathfrak{s}\}}, b^{\{\mathfrak{s}\}})$ and a large step size H, and the fast component with another Runge–Kutta method $\left(A^{\{\mathfrak{f},\mathfrak{f}\}}, b^{\{\mathfrak{f}\}}\right)$ and a small step size $h = H/M$. Here $M \geq 1$ represents the (integer) number of fast steps that are executed for each of the slow steps. MrGARK methods are formally derived in [47]. One step of the method utilizes $s^{\{\mathfrak{s}\}}$ slow stages, denoted by $Y_i^{\{\mathfrak{s}\}}$, and $M\, s^{\{\mathfrak{f}\}}$ fast stages, denoted by $Y_i^{\{\mathfrak{f},\lambda\}}$:

$$\begin{aligned} Y_i^{\{\mathfrak{s}\}} &= y_n + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{s},\mathfrak{s}\}} f^{\{\mathfrak{s}\}} \left(Y_j^{\{\mathfrak{s}\}} \right) \\ &\quad + h \sum_{\lambda=1}^{M} \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{s},\mathfrak{f},\lambda\}} f^{\{\mathfrak{f}\}} \left(Y_j^{\{\mathfrak{f},\lambda\}} \right), \qquad i = 1, \ldots, s^{\{\mathfrak{s}\}}, \end{aligned} \tag{3.4a}$$

$$\begin{aligned} Y_i^{\{\mathfrak{f},\lambda\}} &= \widetilde{y}_{n+(\lambda-1)/M} + H \sum_{j=1}^{s^{\{\mathfrak{s}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} f^{\{\mathfrak{s}\}} \left(Y_j^{\{\mathfrak{s}\}} \right) \\ &\quad + h \sum_{j=1}^{s^{\{\mathfrak{f}\}}} a_{i,j}^{\{\mathfrak{f},\mathfrak{f}\}} f^{\{\mathfrak{f}\}} \left(Y_j^{\{\mathfrak{f},\lambda\}} \right), \qquad \substack{i=1,\ldots,s^{\{\mathfrak{f}\}}\\ \lambda=1,\ldots,M}, \end{aligned} \tag{3.4b}$$

$$\widetilde{y}_{n+\lambda/M} = \widetilde{y}_{n+(\lambda-1)/M} + h \sum_{i=1}^{s^{\{\mathfrak{f}\}}} b_i^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}} \left(Y_i^{\{\mathfrak{f},\lambda\}} \right). \tag{3.4c}$$

The full step is then formed with:

$$y_{n+1} = \widetilde{y}_{n+M/M} + H \sum_{i=1}^{s^{\{\mathfrak{s}\}}} b_i^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\big(Y_i^{\{\mathfrak{s}\}}\big). \tag{3.4d}$$

Let the coupling between the two methods be described by $A^{\{\mathfrak{s},\mathfrak{f},\lambda\}}$ and $A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}$ for $\lambda \in \{1, 2, \cdots, M\}$. The GARK Butcher tableau for the method (3.4) is [47]:

$$\begin{array}{c|c} \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} \\ \hline \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} & \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline \mathbf{b}^{\{\mathfrak{f}\}\,T} & \mathbf{b}^{\{\mathfrak{s}\}\,T} \end{array} := \begin{array}{cccc|c} \frac{1}{M}A^{\{\mathfrak{f},\mathfrak{f}\}} & 0 & \cdots & 0 & A^{\{\mathfrak{f},\mathfrak{s},1\}} \\ \frac{1}{M}\mathbf{1}\,b^{\{\mathfrak{f}\}\,T} & \frac{1}{M}A^{\{\mathfrak{f},\mathfrak{f}\}} & \cdots & 0 & A^{\{\mathfrak{f},\mathfrak{s},2\}} \\ \vdots & & \ddots & & \vdots \\ \frac{1}{M}\mathbf{1}\,b^{\{\mathfrak{f}\}\,T} & \frac{1}{M}\mathbf{1}\,b^{\{\mathfrak{f}\}\,T} & \cdots & \frac{1}{M}A^{\{\mathfrak{f},\mathfrak{f}\}} & A^{\{\mathfrak{f},\mathfrak{s},M\}} \\ \hline \frac{1}{M}A^{\{\mathfrak{s},\mathfrak{f},1\}} & \frac{1}{M}A^{\{\mathfrak{s},\mathfrak{f},2\}} & \cdots & \frac{1}{M}A^{\{\mathfrak{s},\mathfrak{f},M\}} & A^{\{\mathfrak{s},\mathfrak{s}\}} \\ \hline \frac{1}{M}b^{\{\mathfrak{f}\}\,T} & \frac{1}{M}b^{\{\mathfrak{f}\}\,T} & \cdots & \frac{1}{M}b^{\{\mathfrak{f}\}\,T} & b^{\{\mathfrak{s}\}\,T} \end{array}. \tag{3.5}$$

Remark 3.1 (Slow and fast stage numbers)**.** The structure of the Butcher tableau implies that the fast stage ℓ of the fast micro-step λ corresponds to row $(\lambda-1)s^{\{\mathfrak{f}\}} + \ell$ in eq. (3.5), and the slow stage j corresponds to row $Ms^{\{\mathfrak{f}\}} + j$ in eq. (3.5).

Definition 3.2 (Telescopic MrGARK schemes)**.** A *telescopic MrGARK* method [47] uses the same base scheme for the slow and fast partitions:

$$A^{\{\mathfrak{f},\mathfrak{f}\}} = A^{\{\mathfrak{s},\mathfrak{s}\}} = A, \qquad b^{\{\mathfrak{f}\}} = b^{\{\mathfrak{s}\}} = b. \tag{3.6}$$

In this study we will focus on schemes with telescopic property since it allows a simple extension of the MrGARK method to an arbitrary number of partitions (time scales) using successively larger time steps, each an integer multiple of the previous one.

3.3 Order conditions for multirate GARK methods

We consider the following *internal consistency* conditions [47, 97] to ensure that fast and slow right-hand side evaluations are performed at the same points in time:

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbb{1}^{\{\mathfrak{s}\}} = \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbb{1}^{\{\mathfrak{s}\}} := \mathbf{c}^{\{\mathfrak{s}\}}, \qquad \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbb{1}^{\{\mathfrak{f}\}} = \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbb{1}^{\{\mathfrak{f}\}} := \mathbf{c}^{\{\mathfrak{f}\}}, \tag{3.7}$$

where $\mathbb{1}^{\{\sigma\}} := [1, 1, \cdots, 1]^T \in \mathbb{R}^{s^{\{\sigma\}}}$.

Assume that the internal consistency conditions eq. (3.7) hold, and further assume that each individual method $\left(A^{\{\mathfrak{s},\mathfrak{s}\}}, b^{\{\mathfrak{s}\}}\right)$ and $\left(A^{\{\mathfrak{f},\mathfrak{f}\}}, b^{\{\mathfrak{f}\}}\right)$ has at least order 4. Then the MrGARK scheme (3.4)–(3.5) has order four if and only if the following coupling conditions hold [47]:

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}}, \qquad \text{(order 3)} \tag{3.8a}$$

$$\frac{1}{6} = \mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}}, \qquad \text{(order 3)} \tag{3.8b}$$

$$\frac{1}{8} = \mathbf{b}^{\{\mathfrak{f}\}\,T}\,\left(\mathbf{c}^{\{\mathfrak{f}\}} \times \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}}\right), \qquad \text{(order 4)} \tag{3.8c}$$

$$\frac{1}{8} = \mathbf{b}^{\{\mathfrak{s}\}\,T}\,\left(\mathbf{c}^{\{\mathfrak{s}\}} \times \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}}\right), \qquad \text{(order 4)} \tag{3.8d}$$

$$\frac{1}{12} = \mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\left(\mathbf{c}^{\{\mathfrak{s}\}} \times \mathbf{c}^{\{\mathfrak{s}\}}\right), \qquad \text{(order 4)} \tag{3.8e}$$

$$\frac{1}{12} = \mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\left(\mathbf{c}^{\{\mathfrak{f}\}} \times \mathbf{c}^{\{\mathfrak{f}\}}\right), \qquad \text{(order 4)} \tag{3.8f}$$

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}}, \qquad \text{(order 4)} \tag{3.8g}$$

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}}, \qquad \text{(order 4)} \tag{3.8h}$$

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}}, \qquad \text{(order 4)} \tag{3.8i}$$

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}}, \qquad \text{(order 4)} \tag{3.8j}$$

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}}, \qquad \text{(order 4)} \tag{3.8k}$$

$$\frac{1}{24} = \mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}}. \qquad \text{(order 4)} \tag{3.8l}$$

Here "$\times$" denotes component-wise multiplication of two vectors.

Without imposing any special structure on coupling matrices, we can rewrite eq. (3.8) using the block structure shown in eq. (3.5). Considering the following intermediate simplifications:

$$\mathbf{c}^{\{\mathfrak{f}\}} = \frac{1}{M}\left[c^{\{\mathfrak{f}\}} + (\lambda-1)\,\mathbb{1}\right]_{\lambda=1,\dots,M}, \tag{3.9a}$$

$$\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \left[\frac{1}{M^2}\left(\frac{(\lambda-1)^2}{2}\mathbb{1} + (\lambda-1)c^{\{\mathfrak{f}\}} + A^{\{\mathfrak{f},\mathfrak{f}\}}\,c^{\{\mathfrak{f}\}}\right)\right]_{\lambda=1,\dots,M}, \tag{3.9b}$$

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \frac{1}{M^2}\sum_{\lambda=1}^{M} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}}\,\left((\lambda-1)\mathbb{1} + c^{\{\mathfrak{f}\}}\right), \tag{3.9c}$$

$$\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \frac{1}{M}\left[\sum_{k=1}^{\lambda-1}\mathbb{1}\,b^{\{\mathfrak{f}\}\,T}\,A^{\{\mathfrak{f},\mathfrak{s},k\}}\,c^{\{\mathfrak{s}\}} + A^{\{\mathfrak{f},\mathfrak{f}\}}\,A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}\,c^{\{\mathfrak{s}\}}\right]_{\lambda=1,\dots,M}. \tag{3.9d}$$

The order conditions eq. (3.8) can be written in terms of base methods and coupling coefficients as follows:

$$\frac{M}{6} = \sum_{\lambda=1}^{M} b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} c^{\{\mathfrak{s}\}}, \qquad \text{(order 3)} \qquad (3.10a)$$

$$\frac{M^2}{6} = \sum_{\lambda=1}^{M} b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} \left((\lambda-1)\mathbb{1} + c^{\{\mathfrak{f}\}}\right), \qquad \text{(order 3)} \qquad (3.10b)$$

$$\begin{aligned}\frac{M^2}{8} &= \sum_{\lambda=1}^{M} (\lambda-1)\, b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} c^{\{\mathfrak{s}\}} \\ &\quad + \sum_{\lambda=1}^{M} b^{\{\mathfrak{f}\}\,T} \left(c^{\{\mathfrak{f}\}} \times A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} c^{\{\mathfrak{s}\}}\right),\end{aligned} \qquad \text{(order 4)} \qquad (3.10c)$$

$$\frac{M^2}{8} = b^{\{\mathfrak{s}\}\,T} \sum_{\lambda=1}^{M} \left(c^{\{\mathfrak{s}\}} \times \left(A^{\{\mathfrak{s},\mathfrak{f}\,\lambda\}}\left((\lambda-1)\,\mathbb{1} + c^{\{\mathfrak{f}\}}\right)\right)\right), \qquad \text{(order 4)} \qquad (3.10d)$$

$$\frac{M}{12} = \sum_{\lambda=1}^{M} b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} c^{\{\mathfrak{s}\}\times 2}, \qquad \text{(order 4)} \qquad (3.10e)$$

$$\begin{aligned}\frac{M^3}{12} &= \sum_{\lambda=1}^{M} b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} c^{\{\mathfrak{f}\}\times 2} + \sum_{\lambda=1}^{M} (\lambda-1)^2 b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}}\, \mathbb{1} \\ &\quad + 2\sum_{\lambda=1}^{M} (\lambda-1) b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} c^{\{\mathfrak{f}\}},\end{aligned} \qquad \text{(order 4)} \qquad (3.10f)$$

$$\frac{M^2}{24} = \sum_{\lambda=1}^{M} b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{s}\}} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} \left((\lambda-1)\mathbb{1} + c^{\{\mathfrak{f}\}}\right), \qquad \text{(order 4)} \qquad (3.10g)$$

$$\frac{M}{24} = \sum_{\lambda=1}^{M} b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} c^{\{\mathfrak{s}\}}, \qquad \text{(order 4)} \qquad (3.10h)$$

$$\begin{aligned}\frac{M^3}{24} &= \sum_{\lambda=1}^{M} \frac{(\lambda-1)^2}{2} b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}}\, \mathbb{1} \\ &\quad + \sum_{\lambda=1}^{M} (\lambda-1) b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} c^{\{\mathfrak{f}\}} + \sum_{\lambda=1}^{M} b^{\{\mathfrak{s}\}\,T} A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} A^{\{\mathfrak{f},\mathfrak{f}\}} c^{\{\mathfrak{f}\}},\end{aligned} \qquad \text{(order 4)} \qquad (3.10i)$$

$$\frac{M^2}{24} = \sum_{\lambda=1}^{M} \sum_{k=1}^{\lambda-1} b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{s},k\}} c^{\{\mathfrak{s}\}} + \sum_{\lambda=1}^{M} b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{f}\}} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} c^{\{\mathfrak{s}\}}, \qquad \text{(order 4)} \qquad (3.10j)$$

$$\frac{M}{24} = \sum_{\lambda=1}^{M} b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} A^{\{\mathfrak{s},\mathfrak{s}\}} c^{\{\mathfrak{s}\}}, \qquad \text{(order 4)} \qquad (3.10k)$$

$$\frac{M^3}{24} = \sum_{\lambda=1}^{M} \sum_{k=1}^{M} b^{\{\mathfrak{f}\}\,T} A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} A^{\{\mathfrak{s},\mathfrak{f},k\}} \left((k-1)\mathbb{1} + c^{\{\mathfrak{f}\}}\right). \qquad \text{(order 4)} \qquad (3.10l)$$

Remark 3.3 (Design process)**.** A practical design procedure for MrGARK schemes is to first select the base slow and fast methods of desired order, and then solve the order conditions eq. (3.10) for the coupling coefficients $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$ and $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$.

3.4 Linear stability analysis

Following [47, 97] we consider the following scalar model problem:

$$y' = \lambda^{\{\mathfrak{f}\}}\, y + \lambda^{\{\mathfrak{s}\}}\, y. \tag{3.11}$$

where the ratio of the fast to the slow variable is M, the step size ratio. Denote $z^{\{\mathfrak{f}\}} = H\,\lambda^{\{\mathfrak{f}\}}$, $z^{\{\mathfrak{s}\}} = H\,\lambda^{\{\mathfrak{s}\}}$, and

$$s = Ms^{\{\mathfrak{f}\}} + s^{\{\mathfrak{f}\}}, \quad Z = \begin{bmatrix} z^{\{\mathfrak{f}\}}\mathbf{I}_{Ms^{\{\mathfrak{f}\}}\times Ms^{\{\mathfrak{s}\}}} & \mathbf{0} \\ \mathbf{0} & z^{\{\mathfrak{s}\}}\mathbf{I}_{s^{\{\mathfrak{s}\}}\times s^{\{\mathfrak{s}\}}} \end{bmatrix}. \tag{3.12}$$

It was shown in [47, 97] that application of MrGARK method eq. (3.5) to eq. (3.11) leads to the solution

$$y_{n+1} = R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) y_n,$$

with the stability function:

$$R\left(z^{\{\mathfrak{f}\}}, z^{\{\mathfrak{s}\}}\right) = 1 + \mathbf{b}_{\text{GARK}}^T \cdot Z \cdot \left(\mathbf{I}_{s\times s} - \mathbf{A}_{\text{GARK}} \cdot Z\right)^{-1} \cdot \mathbb{1}_{s\times 1}. \tag{3.13}$$

In order to visualize the stability region of a method, we choose:

$$z^{\{\mathfrak{f}\}} = M\,\rho\, e^{-i\,\theta^{\{\mathfrak{f}\}}}, \qquad z^{\{\mathfrak{s}\}} = \rho\, e^{-i\,\theta^{\{\mathfrak{s}\}}}, \qquad \frac{\pi}{2} \le \theta^{\{\mathfrak{f}\}}, \theta^{\{\mathfrak{s}\}} \le \frac{3\pi}{2}, \tag{3.14}$$

such that the ratio of the variable magnitudes matches the ratio of the step sizes. This reduces eq. (3.13) to a function of three real variables by assuming the fast eigenvalue is M times larger in magnitude than the slow eigenvalue. The stability region is the volume in the $\{\theta^{\{\mathfrak{f}\}}, \theta^{\{\mathfrak{s}\}}, \rho\}$ space where the magnitude of the stability function (3.13) is less than or equal to one. Stability regions for each method developed here are provided in section B.1. For some of the methods the region decreases with increasing M. The plots for different values of M correspond to different test problems: when M increases, the scale separation of the test problem also increases (we apply smaller micro-steps to faster problems). The increasing stiffness of the fast component restricts the macro-step, a phenomenon known as stiffness leakage. Coupling coefficients whose magnitude increases rapidly with M can lead to a degradation of stability.

3.5 Decoupled MrGARK methods

We now discuss in detail the structure of the coupling coefficient matrices, and how this structure defines the way the fast and slow stage computations eq. (3.4) are carried out, and therefore determines the practicality of the MrGARK method. In order to construct practical MrGARK methods we need to avoid complex couplings between multiple fast and slow stages. To this end we define *decoupled MrGARK methods*.

Definition 3.4 (Decoupled MrGARK methods)**.** An MrGARK method is decoupled if the computation of its stages proceeds in sequence, such that each slow stage uses only information from other slow stages and the already computed fast stages, and vice-versa. There is no coupling that requires fast and slow stages to be solved together. Any form of implicitness is entirely within the fast or within the slow system.

3.5.1 Structure of the slow-fast coupling (including fast information into the slow stage calculations)

We first introduce a notation for the order of computation of the slow stages with respect to the fast stages. Consider the j-th slow stage (3.4a) at abscissa $c_j^{\{\mathfrak{s}\}}$, i.e., the row $M\, s^{\{\mathfrak{f}\}} + j$ in the Butcher tableau eq. (3.5). We denote its order of computation by (L_j, I_j), i.e., the j-th slow stage is computed immediately after the I_j-th stage of the L_j-th micro-step is computed. This means that the first $L_j - 1$ micro-steps have been completed, and the L_j-th micro-step has partially progressed to compute stage I_j, when the j-th slow stage is evaluated.

When the slow stage $c_j^{\{\mathfrak{s}\}}$ is evaluated after the last stage of micro-step $\lambda - 1$, but before the before the first stage of the λ-th micro-step, we have $(L_j, I_j) = (\lambda - 1, s^{\{\mathfrak{f}\}})$. Equivalently, this situation can be represented by $(L_j, I_j) = (\lambda, 0)$.

In order to construct practical MrGARK methods we require that the evaluation of the j-th slow stage depends only on those fast stages that have been completed; in the GARK Butcher tableau eq. (3.5), the j-th slow stage depends only on the rows $1 : (L_j - 1)s^{\{\mathfrak{f}\}} + I_j$.

The j-th row of the slow-fast coupling matrix:

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} = \frac{1}{M}\left[A^{\{\mathfrak{s},\mathfrak{f},1\}} \quad \dots \quad A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} \quad \dots \quad A^{\{\mathfrak{s},\mathfrak{f},M\}}\right] \in \mathbb{R}^{s^{\{\mathfrak{s}\}} \times M s^{\{\mathfrak{f}\}}} \tag{3.15a}$$

contains the coefficients that bring fast information into the computation of slow stage j. A decoupled MrGARK scheme enjoys the following properties:

- The coupling matrices $A^{\{\mathfrak{s},\mathfrak{f},\lambda\}}$ with the completed microsteps $\lambda = 1 : L_j - 1$ can have full rows:

$$a_{j,k}^{\{\mathfrak{s},\mathfrak{f},\lambda\}} \neq 0, \quad \text{for} \quad \lambda = 1 : L_j - 1.$$

- Rows of the coupling matrix $A^{\{\mathfrak{s},\mathfrak{f},L_j\}}$ can have non-zero entries only in the first I_j positions:

$$\begin{aligned} a_{j,k}^{\{\mathfrak{s},\mathfrak{f},L_j\}} &\neq 0, \qquad k = 1, \cdots, I_j, \\ a_{j,k}^{\{\mathfrak{s},\mathfrak{f},L_j\}} &= 0, \qquad k = I_j + 1, \cdots, s^{\{\mathfrak{f}\}}. \end{aligned}$$

- The coupling matrices $A^{\{\mathfrak{s},\mathfrak{f},\lambda\}}$ with the future microsteps $\lambda = L_j + 1 : M$ are zero:

$$a_{j,k}^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = 0, \quad \text{for} \quad \lambda = L_j + 1 : M.$$

Consequently, for decoupled MrGARK schemes, the slow-fast coupling matrix is lower block triangular:

$$\left(\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\right)_{j,k} = 0, \quad \text{for} \quad 1 \leq j \leq s^{\{\mathfrak{s}\}}, \quad (L_j - 1)s^{\{\mathfrak{f}\}} + I_j + 1 \leq k \leq Ms^{\{\mathfrak{f}\}}, \tag{3.15b}$$

where we used the fact that the fast stage I_j of micro-step L_j has GARK stage number $(L_j - 1)s^{\{\mathfrak{f}\}} + I_j$ in the Butcher tableau eq. (3.5).

3.5.2 Structure of the fast-slow coupling (including slow information into the fast stage calculations)

We next introduce a notation for the order of computation of the fast stages with respect to the slow stages. We denote by $J_{L,i}$ the index of the last slow stage computed before starting the fast stage i of micro-step L. Specifically, the fast stage i at micro-step L is computed after the slow stage $J_{L,i}$, but before the slow stage $J_{L,i} + 1$. For a decoupled MrGARK this stage can only use information from slow stages 1 to $J_{L,i}$. Consequently, the i-th row of the coupling matrix $A^{\{\mathfrak{f},\mathfrak{s},\lambda\}}$ can have nonzero entries only in the first $J_{L,i}$ columns:

$$a_{i,j}^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = 0, \quad \text{for} \quad J_{L,i} + 1 \leq j \leq s^{\{\mathfrak{s}\}}.$$

The coupling matrix:

$$\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}} = \left[A^{\{\mathfrak{f},\mathfrak{s},1\}\,T} \quad \dots \quad A^{\{\mathfrak{f},\mathfrak{s},\lambda\}\,T} \quad \dots \quad A^{\{\mathfrak{f},\mathfrak{s},M\}\,T}\right]^T \in \mathbb{R}^{Ms^{\{\mathfrak{f}\}} \times s^{\{\mathfrak{s}\}}} \tag{3.16a}$$

has a block lower triangular structure:

$$\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}_{(L-1)s^{\{\mathfrak{f}\}}+i,j} = 0 \quad \text{for} \quad J_{L,i} + 1 \leq j \leq s^{\{\mathfrak{s}\}}, \tag{3.16b}$$

where we used the fact that the fast stage i of micro-step L has GARK stage number $(L - 1)s^{\{\mathfrak{f}\}} + i$ in the Butcher tableau eq. (3.5).

3.5.3 Relation between the coupling matrices

From eqs. (3.15) and (3.16) we see that the two coupling matrices $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$ and $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$ have related sparsity structures. For *decoupled MrGARK methods* the sparsity structures are complementary, in the sense that one must have zeros in the entries where the other matrix has nonzero elements:

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} \times \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}\,T} = \mathbf{0}_{s^{\{\mathfrak{s}\}}\times Ms^{\{\mathfrak{f}\}}}, \tag{3.17}$$

where $\times$ denotes element-by-element multiplication. This is schematically illustrated in fig. 3.1a.

For *coupled MrGARK methods* the sparsity structures can overlap, in the sense that they both can have non-zeros entries in the same location:

$$\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}} \times \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}\,T} \neq \mathbf{0}_{s^{\{\mathfrak{s}\}}\times Ms^{\{\mathfrak{f}\}}}. \tag{3.18}$$

This is illustrated in fig. 3.1c. The overlapping non-zero coupling coefficients, indicated by dashed boxes in fig. 3.1c, imply that the corresponding fast and slow stages need to be computed together, in a step that involves the entire non-partitioned system. Coupled methods are not pursued further in this study.

3.5.4 Order of the slow and fast stage evaluation

The sparsity structure of the coupling matrices $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$ and $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$ determines the order in which the fast and slow stages can be evaluated such as to respect the data dependencies between them. Using the notation defined in section 3.5.1 and section 3.5.2, the general order in which stages are evaluated is follows:

1. Start by evaluating the fast stages i for which $a_{1,i}^{\{\mathfrak{s},\mathfrak{f}\}} \neq 0$; these are the fast stages needed by the computation of the first slow stage $c_1^{\{\mathfrak{s}\}}$. If the entire first row $a_{1,:}^{\{\mathfrak{s},\mathfrak{f}\}} = 0$ then proceed with evaluating the first slow stage $c_1^{\{\mathfrak{s}\}}$.
2. After stage I_j of the L_j-th micro-step is computed, the j-th stage $c_j^{\{\mathfrak{s}\}}$ of the slow method is evaluated.
3. Continue with evaluating stages of the fast method, and stop after stage I_{j+1} of the L_{j+1}-st micro-step to compute stage $c_{j+1}^{\{\mathfrak{s}\}}$ of the slow method.
4. Continue until all fast and slow stages are evaluated.

Remark 3.5 (Time ordering)**.** Assuming that the slow stage abscissae are increasing, and that fast stage abscissae are non-decreasing in time:

$$c_1^{\{\mathfrak{s}\}} < \cdots < c_{s^{\{\mathfrak{s}\}}}^{\{\mathfrak{s}\}}, \qquad c_1^{\{\mathfrak{f}\}} \leq \cdots \leq c_{s^{\{\mathfrak{f}\}}}^{\{\mathfrak{f}\}}, \tag{3.19}$$

a natural order to evaluate the MrGARK stages follows the time ordering of their abscissae. Note that the j-th slow stage approximates the solution at time $T_{Ms^{\{\mathrm{f}\}}+j} = t_n + c_j^{\{\mathrm{s}\}} H$, while the j-th fast stage of the L-th microstep approximates the solution at time $T_{(L-1)s^{\{\mathrm{f}\}}+j} = t_n + (L - 1 + c_j^{\{\mathrm{f}\}})\,(H/M)$. The pairs (L_j, I_j) are chosen such that the stages are evaluated in increasing order of their approximation times:

$$\frac{L_j - 1 + c_{I_j}^{\{\mathrm{f}\}}}{M} \le c_j^{\{\mathrm{s}\}} \le \frac{L_j - 1 + c_{I_j+1}^{\{\mathrm{f}\}}}{M} \quad \text{for} \quad 1 \le I_j < s^{\{\mathrm{f}\}}, \tag{3.20a}$$

$$\frac{L_j + c_{s^{\{\mathrm{f}\}}}^{\{\mathrm{f}\}}}{M} \le c_j^{\{\mathrm{s}\}} \le \frac{L_j + 1 + c_1^{\{\mathrm{f}\}}}{M} \quad \text{for} \quad I_j = s^{\{\mathrm{f}\}}, \tag{3.20b}$$

$$\text{or} \quad \frac{L_j - 1 + c_{s^{\{\mathrm{f}\}}}^{\{\mathrm{f}\}}}{M} \le c_j^{\{\mathrm{s}\}} \le \frac{L_j + c_1^{\{\mathrm{f}\}}}{M} \quad \text{for} \quad I_j = 0.$$

Remark 3.6 (Simpler time ordering)**.** It is possible to simplify the time ordering such that the slow stages are evaluated at the end of full micro-steps. The stage $c_j^{\{\mathrm{s}\}}$ is evaluated after the end of micro-step $\lambda - 1$, but before micro-step λ, if $(\lambda - 1)/M \le c_j^{\{\mathrm{s}\}} < \lambda/M$, in which case $L_j := \lambda$ and $I_j = 0$. When $c_j^{\{\mathrm{s}\}} \ge 1$ we take $L_j = M, I_j = 0$.

3.5.5 Reordering the GARK Butcher tableau

In the standard form of the GARK Butcher tableau eq. (3.5) the first $Ms^{\{\mathrm{f}\}}$ stages are for the fast method, and stages $Ms^{\{\mathrm{f}\}} + 1$ to $Ms^{\{\mathrm{f}\}} + s^{\{\mathrm{s}\}}$ are for the slow method. Let $\mathtt{IC}$ be the vector of GARK tableau stage indices sorted in the order of computations, as summarized in fig. 3.2. A renumbering of stages leads to a row and column permutation of the Butcher tableau eq. (3.5). The reordered Butcher matrix is $\mathbf{A}_{\mathrm{GARK}}(\mathtt{IC}, \mathtt{IC})$. The reordering is illustrated in fig. 3.1. We distinguish the following cases:

- If the reordered Butcher tableau is strictly lower triangular then the MrGARK method is explicit, and each stage uses only previously computed information.

- If the reordered Butcher tableau is lower triangular, with some non-zero diagonal entries, then the MrGARK method is implicit, but decoupled. The non-zero diagonal entries correspond to implicit fast or slow stages. This case is illustrated in fig. 3.1b.

- Finally, if the reordered Butcher tableau has block-diagonal entries then the MrGARK method is coupled, in the sense that several fast and slow stages need to be solved together, in a coupled manner. This case is illustrated in fig. 3.1d.

As an example let us consider the GARK Butcher matrix for the explicit method EX2-EX2 2(1)[A] introduced in section B.1.1 for $M = 3$:

$$\mathbf{A} = \left(\begin{array}{cccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{9} & 0 & 0 & 0 & 0 & 0 & \frac{2}{9} & 0 \\ \frac{1}{4} & \frac{3}{4} & 0 & 0 & 0 & 0 & \frac{11}{60} & \frac{3}{20} \\ \frac{1}{4} & \frac{3}{4} & \frac{2}{9} & 0 & 0 & 0 & -\frac{19}{180} & \frac{9}{20} \\ \frac{1}{4} & \frac{3}{4} & \frac{1}{4} & \frac{3}{4} & 0 & 0 & \frac{31}{60} & \frac{3}{20} \\ \frac{1}{4} & \frac{3}{4} & \frac{1}{4} & \frac{3}{4} & \frac{2}{9} & 0 & -\frac{79}{180} & \frac{9}{20} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & 0 \end{array}\right), \quad \mathbf{A}(\text{IC}, \text{IC}) = \left(\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{9} & \frac{2}{9} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3} & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{11}{60} & \frac{1}{4} & \frac{3}{4} & \frac{3}{20} & 0 & 0 & 0 & 0 \\ -\frac{19}{180} & \frac{1}{4} & \frac{3}{4} & \frac{9}{20} & \frac{2}{9} & 0 & 0 & 0 \\ \frac{31}{60} & \frac{1}{4} & \frac{3}{4} & \frac{3}{20} & \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ -\frac{79}{180} & \frac{1}{4} & \frac{3}{4} & \frac{9}{20} & \frac{1}{4} & \frac{3}{4} & \frac{2}{9} & 0 \end{array}\right),$$

where $\text{IC} = [7, 1, 2, 8, 3, 4, 5, 6]$. The permuted Butcher tableau verifies the explicit nature of the method, and also provides a progression of stage computations that is practically easy to implement.

3.6 Design of practical decoupled MrGARK methods

In this section we discuss several desirable properties of practical MrGARK methods, and the design methodologies to incorporate them in the construction of new high order schemes.

3.6.1 Design principles

The following set of design principles incorporates properties that are important for ensuring the practicality of MrGARK schemes:

1. Practical MrGARK methods need to be *high order* (have an order of accuracy larger than two), and *generic* in the multirate ratio (have the same order of accuracy for any M.) This typically requires that the coupling coefficients are functions of M. We have addressed these requirements via the order conditions derived in section 3.3.

2. Methods where the fast and slow base schemes are both either explicit or implicit should be *telescopic* eq. (3.6) in order to be easily applicable to multi-partitioned systems with multiple time scales. All explicit-explicit methods proposed here are telescopic.

3. In order to maintain computational efficiency, a complex coupling between multiple fast and slow stages needs to be avoided. Here the focus is on *decoupled* multirate methods satisfying equation eq. (3.17), as they are far less computationally demanding than coupled methods. However, the overall stability of the scheme may be affected by decoupling.

4. Methods should be optimized for general-purpose time integration, i.e. have small principal error terms and large stability regions. Note that the principal error for

MrGARK methods is a function of M. If not properly controlled, the coupling errors can grow with M, thereby reducing the overall efficiency of the method. All schemes derived herein have coefficients optimized for small principal error and large stability region.

5. Consider the case where one of the base methods $q \in \{\mathfrak{f}, \mathfrak{s}\}$ is implicit. A favorable property of the entire MrGARK method is stiff accuracy [47]:

$$\mathbf{e}_{s^{\{q\}}}^T \mathbf{A}^{\{q,\mathfrak{f}\}} = \mathbf{b}^{\{\mathfrak{f}\}\,T}, \qquad \mathbf{e}_{s^{\{q\}}}^T \mathbf{A}^{\{q,\mathfrak{s}\}} = \mathbf{b}^{\{\mathfrak{s}\}\,T}. \tag{3.21}$$

   This property implies that the base implicit Runge–Kutta method is stiffly accurate, and that the MrGARK stability function approaches zero as the q-th component of the system becomes infinitely stiff [47]. All schemes derived herein having an implicit base method enjoy the property eq. (3.21).

6. Practical use of MrGARK methods demands an error control mechanism that is capable of adapting both the step size H and the multirate ratio M. Therefore the error control problem in multirate integration is fundamentally more complex than in traditional, single rate integration. We fully address the error control issue in section 3.7.

7. It is desirable to derive multirate methods with reduced coupling errors by requiring that both the main and the embedded methods satisfy higher order coupling conditions. This strategy isolates the dominant local truncation errors to the solution of the slow and fast components and greatly simplifies the task of adaptively choosing H and M, as discussed in section 3.7.

   Definition 3.7 (Naturally adaptive schemes)**.** A MrGARK scheme of order p is *naturally adaptive* if the fast and slow local truncation error terms (corresponding to N-trees with nodes of the same color) are $\mathcal{O}(h^{p+1})$, and the coupling local truncation error terms (corresponding to N-trees with nodes of different colors) are $\mathcal{O}(h^{p+2})$.

   We construct *naturally adaptive* second and third order schemes in section B.1.

8. Methods of type S are optimized for simplicity and stability. First, by design, one seeks to maximize the sparsity of the coupling coefficient matrices, such as to simplify the implementation and decrease data dependencies. Next, the coupling coefficients can potentially become large in magnitude when M increases, and this can lead to large cancellation errors in the context of finite precision arithmetic, as well as to a degradation of numerical stability. It is desirable that the coupling coefficients are optimized such that their magnitude remains bounded for large values M.

3.6.2 Design process

The order conditions eq. (3.8) and the additional constraints associated with the design principles of section 3.6.1 lead to large, nonlinear systems of equations that need to be

solved for the method coefficients. The resulting coupling coefficients $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$ and $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$ are *functions* of the micro-step number λ and the multi-rate step size ratio M. The proposed approach for designing MrGARK methods of order p is a three-step process, as follows.

1. The first step is to construct optimized base slow $\left(A^{\{\mathfrak{s},\mathfrak{s}\}}, b^{\{\mathfrak{s}\}}\right)$ and fast $\left(A^{\{\mathfrak{f},\mathfrak{f}\}}, b^{\{\mathfrak{f}\}}\right)$ schemes of order p. Implicit base methods are selected to be stiffly-accurate SDIRK schemes. Explicit base methods are chosen to have small principal error components. When both the slow and the fast methods are of the same type (explicit or implicit) we choose the same discretization coefficients $A^{\{\mathfrak{s},\mathfrak{s}\}} = A^{\{\mathfrak{f},\mathfrak{f}\}}$, $b^{\{\mathfrak{s}\}} = b^{\{\mathfrak{f}\}}$ such as to obtain a telescopic MrGARK method.

2. The second step is to define the sparsity patterns of the coupling matrices $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$ and $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$. These patterns determine the computational flow of the method and its implementation complexity, and influence the overall stability and accuracy properties. We manually test various sparsity patterns to balance all of these properties.

3. The third step computes the coupling coefficients $\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}$ and $\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}$ such as to satisfy the coupling conditions eq. (3.8) up to order p. Any free parameters in the family are used to minimize the Euclidean norm of the residuals of the order $p+1$ coupling conditions; for naturally adaptive MrGARK methods all these residuals are cancelled. In this work the solution of order conditions and the minimization of the error coefficients are carried out with Mathematica Version 11.2.

An alternative to this derivation strategy is a monolithic constrained optimization procedure that minimizes the residuals of the order $p+1$ coupling conditions, subject to solving the conditions eq. (3.8) up to order p, and subject to structural constraints such as decoupling (3.17) and stiff accuracy (3.21).

The proposed three-step procedure is preferable for two reasons. First, we expect MrGARK methods to be applied to problems with a rather weak coupling between partitions, where the primary sources of error are the base methods and not the coupling. Our approach gives precedence to the base errors first. Second, the procedure is practical as it reduces the number of nonlinear equations that need to be solved together during the design.

3.7 Error estimation and adaptive MrGARK methods

Adaptivity of traditional (single rate) methods adjusts the step size such as to ensure the desired accuracy of the solution at a minimal computational effort. In the context of Runge–Kutta schemes a second "embedded" method is used to provide an aposteriori estimate of the local truncation error [52, CH II.4]. The step size is adjusted to bring the local error estimate to the user prescribed level using the asymptotic relation that this error is proportional to $\propto H^{p+1}$.

Adaptivity of multirate methods is more complex, as there are two independent parameters that control the solution accuracy and efficiency: the macro-step size H and the multirate ratio M (or, equivalently, the macro-step H and the micro-step h.) In order to achieve adaptivity of both H and M we propose to construct error estimates using not one, but several embedded methods, such as to obtain additional information about the structure of the local truncation error. Analytical asymptotic formulas are used to quantify the dependency of the local error terms on both H and M. Armed with this information, we develop several criteria for adaptively selecting both the macro-step size and the multirate ratio.

3.7.1 Local error structure for a second order MrGARK method

In order to understand the structure of the local truncation error we first consider a second order, internally consistent MrGARK scheme. The principal error terms of order $\mathcal{O}(H^3)$ are associated with the third order conditions, and are provided in table 3.1.

Error type	Error term	Elementary differential	Order condition	Residuals for IM2-EX2 2(1)[A]
Slow	$e_{3,1}^{\{\mathfrak{s}\}}$	$f_{y,y}^{\{\mathfrak{s}\}}\left(f^{\{\mathfrak{s}\}}, f^{\{\mathfrak{s}\}}\right)$	$\mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{c}^{\{\mathfrak{s}\}\times 2} = \frac{1}{3}$	0
Slow	$e_{3,2}^{\{\mathfrak{s}\}}$	$f_y^{\{\mathfrak{s}\}} f_y^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}$	$\mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \frac{1}{6}$	$\frac{1}{6}$
Fast	$e_{3,1}^{\{\mathfrak{f}\}}$	$f_{y,y}^{\{\mathfrak{f}\}}\left(f^{\{\mathfrak{f}\}}, f^{\{\mathfrak{f}\}}\right)$	$\mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{c}^{\{\mathfrak{f}\}\times 2} = \frac{1}{3}$	$\frac{4-3\sqrt{2}}{12M^2}$
Fast	$e_{3,2}^{\{\mathfrak{f}\}}$	$f_y^{\{\mathfrak{f}\}} f_y^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}}$	$\mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \frac{1}{6}$	$\frac{4-3\sqrt{2}}{6M^2}$
Coupling	$e_{3,1}^{\{\mathfrak{c}\}}$	$f_y^{\{\mathfrak{s}\}} f_y^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}}$	$\mathbf{b}^{\{\mathfrak{s}\}\,T}\,\mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}\,\mathbf{c}^{\{\mathfrak{f}\}} = \frac{1}{6}$	$\frac{3\sqrt{2}-3-M}{12M}$
Coupling	$e_{3,2}^{\{\mathfrak{c}\}}$	$f_y^{\{\mathfrak{f}\}} f_y^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}$	$\mathbf{b}^{\{\mathfrak{f}\}\,T}\,\mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}\,\mathbf{c}^{\{\mathfrak{s}\}} = \frac{1}{6}$	$\frac{1}{6}$

Table 3.1: Principal error terms of $\mathcal{O}(H^3)$ for second order MrGARK schemes.

As an example, consider the second order MrGARK method IM2-EX2 2(1)[A] from section B.1.4. We note from table 3.1 that the third order residuals for this method are all asymptotically bounded as M increases, and the local truncation error behaves like:

$$\begin{aligned}
\text{LTE} = &\left(0 \cdot f_{y,y}^{\{\mathfrak{s}\}}\left(f^{\{\mathfrak{s}\}}, f^{\{\mathfrak{s}\}}\right) + \frac{1}{6} \cdot f_y^{\{\mathfrak{s}\}} f_y^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\right) H^3 \\
&+ \left(\frac{4-3\sqrt{2}}{12M^2} \cdot f_{y,y}^{\{\mathfrak{f}\}}\left(f^{\{\mathfrak{f}\}}, f^{\{\mathfrak{f}\}}\right) + \frac{4-3\sqrt{2}}{6M^2} \cdot f_y^{\{\mathfrak{f}\}} f_y^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}}\right) H^3 \\
&+ \left(\frac{3\sqrt{2}-3-M}{12M} \cdot f_y^{\{\mathfrak{s}\}} f_y^{\{\mathfrak{f}\}} f^{\{\mathfrak{f}\}} + \frac{1}{6} \cdot f_y^{\{\mathfrak{f}\}} f_y^{\{\mathfrak{s}\}} f^{\{\mathfrak{s}\}}\right) H^3 + \mathcal{O}\left(H^4\right), \\
\|\text{LTE}\| \approx &\left(K_1 + K_2 \frac{1}{M} + K_3 \frac{1}{M^2}\right) H^3 + \mathcal{O}\left(H^4\right).
\end{aligned} \tag{3.22}$$

In some cases, when there are sufficient degrees of freedom left after solving the order conditions, it is possible for a method of order p to either cancel out coupling errors of order

$\mathcal{O}(H^{p+1})$ (fig. 3.3a) or to minimize them to assure better coupling behavior (fig. 3.3b). fig. 3.3 shows how leading error coefficients change with M for two optimized MrGARK methods.

3.7.2 General structure of the MrGARK local truncation error

As the analysis in section 3.7.1 reveals, the local truncation error of the MrGARK method eq. (3.4) has three components associated with the slow integration, the fast integration, and with the coupling:

$$\text{LTE} = \text{LTE}^{\{\mathfrak{s}\}} + \text{LTE}^{\{\mathfrak{f}\}} + \text{LTE}^{\{\mathfrak{c}\}}. \tag{3.23}$$

For a method of order p the slow truncation error is that of applying one step with the base slow Runge–Kutta method with a step size H:

$$\left\|\text{LTE}^{\{\mathfrak{s}\}}\right\| \approx C^{\{\mathfrak{s}\}} \cdot H^{p+1} + \mathcal{O}\left(H^{p+2}\right). \tag{3.24a}$$

The fast truncation error is that of applying M consecutive steps with the base fast Runge–Kutta method with a step size H/M. We use the global error estimate [52, CH II.3]:

$$\left\|\text{LTE}^{\{\mathfrak{f}\}}\right\| \leq \left(\frac{H}{M}\right)^p \frac{C'}{L}\left(\exp\{(L\,H)\} - 1\right) = \left(\frac{H}{M}\right)^p \frac{C'}{L}\left(LH + \mathcal{O}\left(H^2\right)\right),$$

where $\left\|\partial f^{\{\mathfrak{f}\}}/\partial y\right\| \leq L$ and C' is a constant, to obtain:

$$\left\|\text{LTE}^{\{\mathfrak{f}\}}\right\| \leq \frac{C^{\{\mathfrak{f}\}}}{M^p} \cdot H^{p+1} + \mathcal{O}\left(H^{p+2}\right). \tag{3.24b}$$

The principal terms of the coupling errors correspond to N-trees having $p+1$ nodes of two colors. A tree with k fast nodes and $p+1-k$ slow nodes corresponds to a residual term constructed from multiplying k fast matrix blocks in the Butcher tableau (3.5) and $p+1-k$ slow matrix blocks. Note that each of the fast matrix blocks ($\mathbf{b}^{\{\mathfrak{f}\}\,T}, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}, \mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}$) in (3.5) carries a scaling factor of $1/M$. Assuming that the coupling coefficients remain bounded for large M, a product of k fast blocks is $\mathcal{O}\left(1/M^k\right)$. Since we have coupling trees with $p \geq k \geq 1$ fast nodes, the corresponding products of fast blocks have scalings ranging from $1/M$ to $1/M^p$. Residuals contain sums of M elementary coupling blocks as seen in (3.10); sums of M fast blocks have scaling factors ranging from 1 to $1/M^{p-1}$, but some of the sums can have a small, fixed number of terms. In addition, the magnitude of the coupling coefficients, as resulted from the solution of order conditions, can scale as M^ℓ (for some $\ell \geq 0$). Consequently, a generic expression for the local coupling error is:

$$\left\|\text{LTE}^{\{\mathfrak{c}\}}\right\| \approx \left(\sum_{i=-\ell}^{p} C_i^{\{\mathfrak{c}\}}\, M^{-i}\right) \cdot H^{p+1} + \mathcal{O}\left(H^{p+2}\right), \tag{3.24c}$$

While the slow and fast errors are simple functions of H and M, the dependency of the coupling error on M is more difficult to describe even with access to the residuals of each

of the $p+1$ order conditions. These residuals can be positive or negative. Moreover, the evolution of the coupling error as a function of M depends not only on residuals, but also on elementary differentials, since the constants $C_i^{\{\mathfrak{c}\}}$ in (3.24c) are linear combinations of products of method residuals and norms of elementary differentials. Therefore, in an adaptive method, the effect of changing M on the coupling error is more difficult to quantify.

3.7.3 Use of multiple embedded methods to estimate the local truncation error

Following the traditional Runge–Kutta strategy, we look to use embedded methods to obtain estimates of the local truncation error. Specifically, we design pairs of main weight vectors $(b^{\{\mathfrak{f}\}}, b^{\{\mathfrak{s}\}})$ and embedded weight vectors $(\widehat{b}^{\{\mathfrak{f}\}}, \widehat{b}^{\{\mathfrak{s}\}})$ that produce solutions of different orders when paired with $(\mathbf{A}^{\{\mathfrak{f},\mathfrak{f}\}}, \mathbf{A}^{\{\mathfrak{f},\mathfrak{s}\}}, \mathbf{A}^{\{\mathfrak{s},\mathfrak{f}\}}, \mathbf{A}^{\{\mathfrak{s},\mathfrak{s}\}})$. The choice of the embedded weights is made such that additional function evaluations are avoided.

In the traditional Runge–Kutta approach, a single embedded method is used to provide a single estimate of the error, and a step size scaling factor is computed (for the next step) such that the corresponding scaled error estimate satisfies the accuracy requirements at hand. In multirate integration the solution accuracy depends on two parameters, H and M, that can be adjusted independently. We have seen in section 3.7.2 that changes in these parameters affect differently the slow, fast, and coupling components of the local truncation error.

The proposed strategy to select H and M adaptively relies on multiple embedded methods to independently estimate different parts of the error. Specifically, consider an MrGARK scheme that produces a main solution y_{n+1} of order p, i.e., cancels all residuals for two-trees with up to p nodes. We seek to build an embedded solution $\widehat{y}_{n+1}^{\{\mathfrak{s}\}}$ that cancels all residuals up to order $p-1$, as well as all fast and all coupling residuals of order p. Similarly, we seek to build a second embedded solution $\widehat{y}_{n+1}^{\{\mathfrak{f}\}}$ that cancels all residuals up to order $p-1$, as well as all slow and all coupling residuals of order p. The three components of the local truncation error can then be estimated as follows:

$$\begin{aligned} &\mathbf{LTE}_{n+1} \approx y_{n+1} - \widehat{y}_{n+1}, \quad \mathbf{LTE}_{n+1}^{\{\mathfrak{s}\}} \approx y_{n+1} - \widehat{y}_{n+1}^{\{\mathfrak{s}\}}, \quad \mathbf{LTE}_{n+1}^{\{\mathfrak{f}\}} \approx y_{n+1} - \widehat{y}_{n+1}^{\{\mathfrak{f}\}}, \\ &\mathbf{LTE}_{n+1}^{\{\mathfrak{c}\}} = \mathbf{LTE}_{n+1} - \mathbf{LTE}_{n+1}^{\{\mathfrak{s}\}} - \mathbf{LTE}_{n+1}^{\{\mathfrak{f}\}} \approx \widehat{y}_{n+1}^{\{\mathfrak{s}\}} + \widehat{y}_{n+1}^{\{\mathfrak{f}\}} - \widehat{y}_{n+1} - y_{n+1}. \end{aligned} \tag{3.25}$$

The construction of three different embedded methods that capture precisely some components of the error can be very difficult to achieve for high order schemes. For this reason we now discuss a simplified error estimation strategy that utilizes only two embedded methods. Consider an MrGARK scheme with primary weights $(b^{\{\mathfrak{f}\}}, b^{\{\mathfrak{s}\}})$ that produces a main solution y_{n+1} of order p. The residuals corresponding to two-trees with up to p nodes are zero. We construct three different embedded solutions, as follows:

- A generic embedded solution $\widehat{y}_{n+1}$ of order $p-1$, obtained with weights $(\widehat{b}^{\{\mathfrak{f}\}}, \widehat{b}^{\{\mathfrak{s}\}})$, that

captures all the error terms and is used to approximate the overall local truncation error. The residuals associated with two-trees with up to $p-1$ nodes are zero, while the residuals of two-trees with p nodes can be nonzero.

- An embedded solution $\widehat{y}_{n+1}^{\{\mathfrak{s}\}}$, generated with the weights $(b^{\{\mathfrak{f}\}}, \widehat{b}^{\{\mathfrak{s}\}})$, captures all of the slow error, part of the coupling error, and none of the fast error. The weight vector $b^{\{\mathfrak{f}\}}$ cancels the residuals for all two-trees with up to p nodes with a fast-colored root. The weight vector $\widehat{b}^{\{\mathfrak{s}\}}$ cancels the residuals for all two-trees with up to $p-1$ nodes with a slow-colored root. However, the residuals of the two-trees with p nodes and a slow-colored root, and the associated $\mathcal{O}(H^p)$ error terms, can be nonzero. Trees with a slow-colored root correspond to either slow or to coupling trees, and the solution difference $y_{n+1} - \widehat{y}_{n+1}^{\{\mathfrak{s}\}}$ captures the sum of the corresponding errors terms.

- Similarly, an embedded solution $\widehat{y}_{n+1}^{\{\mathfrak{f}\}}$, generated with $(\widehat{b}^{\{\mathfrak{f}\}}, b^{\{\mathfrak{s}\}})$, captures all of the fast error, part of the coupling error, and none of the slow error.

In general, these mixed embeddings do not exactly isolate the slow, fast, and coupling error, but they can serve as approximations in (3.25). Note that the resulting approximate $\text{LTE}^{\{\mathfrak{s}\}}$ and $\text{LTE}^{\{\mathfrak{f}\}}$ contain the parts of the coupling error corresponding to trees of order p with slow-colored roots and with fast colored-roots, respectively. Consequently the $\text{LTE}^{\{\mathfrak{c}\}}$ estimated by the solution difference in (3.25) is zero.

However, for naturally adaptive methods (proposition 3.7), the simplified error estimation strategy does separate the main components of the fast and slow errors, as explained next. For a naturally adaptive method of order p with weights $(b^{\{\mathfrak{f}\}}, b^{\{\mathfrak{s}\}})$ the residuals corresponding to all two-trees with up to p nodes are zero, and the coupling residuals for two-trees with $p+1$ nodes are also zero. The non-zero residuals of order p correspond to either fast or low trees with p nodes.

A naturally adaptive embedded solution $\widehat{y}_{n+1}$ of order $p-1$, obtained with weights $(\widehat{b}^{\{\mathfrak{f}\}}, \widehat{b}^{\{\mathfrak{s}\}})$, cancels all residuals of order up to $p-1$; it also cancels the coupling residuals for two-trees of up to p nodes. Consequently, the difference $y_{n+1} - \widehat{y}_{n+1}^{\{\mathfrak{s}\}}$ contains only $\mathcal{O}(H^p)$ terms corresponding to slow-colored trees, and $y_{n+1} - \widehat{y}_{n+1}^{\{\mathfrak{f}\}}$ contains only $\mathcal{O}(H^p)$ terms corresponding to fast-colored trees. Consequently, naturally adaptive embedded methods *do* isolate the slow and the fast errors. This property allows to construct more accurate adaptivity mechanisms.

3.7.4 Controlling errors by adapting both the macro-step and the micro-step

Based on the understanding of the structure of errors in the MrGARK framework, we now look into practical approaches to adaptivity of multirate methods. The following error esti-

mates are available via the set of embedded methods:

$$\widehat{\varepsilon}_{n+1} := \left\|y_{n+1} - \widehat{y}_{n+1}\right\|_{\text{err}}, \quad \widehat{\varepsilon}^{\{\mathfrak{s}\}}_{n+1} := \left\|y_{n+1} - \widehat{y}^{\{\mathfrak{s}\}}_{n+1}\right\|_{\text{err}}, \quad \widehat{\varepsilon}^{\{\mathfrak{f}\}}_{n+1} := \left\|y_{n+1} - \widehat{y}^{\{\mathfrak{f}\}}_{n+1}\right\|_{\text{err}},$$

where the error is measured by the following relative error norm [52, CH II.4]:

$$\left\|x - y\right\|_{\text{err}} := \sqrt{\frac{1}{d}\sum_{i=1}^{d}\left(\frac{x_i - y_i}{\text{AbsTol}_i + \text{RelTol}_i \cdot \max\left(|(x_i|, |y_i|\right)}\right)^2}.$$

Several heuristic strategies that use these error estimates to adapt both H and M are discussed below.

Balancing error strategy

In this approach, we first use the estimated total truncation error $\widehat{\varepsilon}_{n+1}$ to control the macro-step size using the traditional mechanism based on the asymptotic error behavior [52, CH II.4]:

$$\widehat{\varepsilon}_{n+2} \leq 1 \qquad \Rightarrow \qquad H_{\text{new}} = \text{fac} \cdot H \cdot \left(\widehat{\varepsilon}_{n+1}\right)^{-\frac{1}{p}},$$

where $\text{fac} < 1$ is a safety factor.

Next, the multirate ratio M is adjusted such that the estimated slow and fast error components over the next step are equal to each other, i.e., the slow and fast contributions to the error are balanced. Assuming $q = \min\left(p, \hat{p}\right)$ and using the asymptotic formulas (3.24) we have that:

$$\begin{aligned} \widehat{\varepsilon}^{\{\mathfrak{s}\}}_{n+2} &= \widehat{\varepsilon}^{\{\mathfrak{f}\}}_{n+2}, \\ C^{\{\mathfrak{s}\}} \cdot H^{q+1}_{\text{new}} &\approx \frac{C^{\{\mathfrak{f}\}}}{M^{q}_{\text{new}}} \cdot H^{q+1}_{\text{new}}, \\ \widehat{\varepsilon}^{\{\mathfrak{s}\}}_{n+1} \cdot \frac{H^{q+1}_{\text{new}}}{H^{q+1}} &\approx \widehat{\varepsilon}^{\{\mathfrak{f}\}}_{n+1} \cdot \frac{M^{q}}{M^{q}_{\text{new}}} \cdot \frac{H^{q+1}_{\text{new}}}{H^{q+1}}, \end{aligned} \qquad \Rightarrow \qquad M_{\text{new}} \approx M \cdot \left(\frac{\widehat{\varepsilon}^{\{\mathfrak{f}\}}_{n+1}}{\widehat{\varepsilon}^{\{\mathfrak{s}\}}_{n+1}}\right)^{\frac{1}{q}}. \tag{3.26}$$

In this strategy, as well as the others, M can be rounded up, down, or to the nearest integer. Also in practice, the re-scaling of H and M for the next step can be bounded up and down in order to avoid large jumps and oscillations. This adaptivity strategy serves as a very simple heuristic. However, the choices of the new H and of the new M are made independently of each other, and the approach does not account for how their interaction impacts the error; the only mechanism for controlling the coupling error is the change in H.

Efficiency optimization strategy

This approach focuses on the important aspect of the overall cost of multirate integration. Evaluation of the slow and fast partitions can have very different computational costs in

some applications. Moreover, an implicit method (e.g., applied to solve the fast component) is likely to be much more expensive than an explicit method (e.g., applied to the slow component). We require the adaptive selection of H and M to satisfy the error tolerance criteria at a minimal overall computational cost.

Let $t^{\{\mathfrak{s}\}}$ and $t^{\{\mathfrak{f}\}}$ represent the computational costs of a slow macro-step and a fast micro-step, respectively. We define the computational efficiency of a step as the progress made during the step (H) divided by the total cost of executing step ($t^{\{\mathfrak{s}\}} + M_{\text{new}} t^{\{\mathfrak{f}\}}$).

The new values of H and M are selected such as to achieve the desired accuracy while maximizing the computational efficiency. This requires solving the following constrained optimization problem to minimize the inverse of efficiency:

$$\begin{aligned} &\min_{H_{\text{new}}, M_{\text{new}}} \frac{t^{\{\mathfrak{s}\}} + M_{\text{new}}\, t^{\{\mathfrak{f}\}}}{H_{\text{new}}}, \\ &\text{subject to } \widehat{\varepsilon}_{n+2} = 1. \end{aligned} \tag{3.27}$$

Expanding the constraint yields:

$$1 = \widehat{\varepsilon}_{n+2} \approx \widehat{\varepsilon}_{n+1}^{\{\mathfrak{s}\}} \cdot \frac{H_{\text{new}}^{q+1}}{H^{q+1}} + \widehat{\varepsilon}_{n+1}^{\{\mathfrak{f}\}} \cdot \frac{M^q}{M_{\text{new}}^q} \cdot \frac{H_{\text{new}}^{q+1}}{H^{q+1}},$$

where we have used the fact that, for naturally adaptive methods, the coupling component of the local truncation error is negligible in the asymptotic regime. The constraint equation can be solved explicitly for H_{new}:

$$H_{\text{new}} = H \cdot \left(\widehat{\varepsilon}_{n+1}^{\{\mathfrak{s}\}} + \widehat{\varepsilon}_{n+1}^{\{\mathfrak{f}\}} \cdot \frac{M^q}{M_{\text{new}}^q} \right)^{-\frac{1}{q+1}}. \tag{3.28}$$

After eliminating the constraint the optimization problem eq. (3.27) simplifies to:

$$\min_{M_{\text{new}}} \frac{t^{\{\mathfrak{s}\}} + M_{\text{new}}\, t^{\{\mathfrak{f}\}}}{H} \left(\widehat{\varepsilon}_{n+1}^{\{\mathfrak{s}\}} + \widehat{\varepsilon}_{n+1}^{\{\mathfrak{f}\}} \cdot \frac{M^q}{M_{\text{new}}^q} \right)^{\frac{1}{q+1}}. \tag{3.29}$$

Note this is an integer minimization problem. One can solve it as a continuous optimization problem, then round the result to an integer to find the optimal M_{new}. Afterwards H_{new} is computed from eq. (3.28).

Remark 3.8 (Timing)**.** The CPU times $t^{\{\mathfrak{s}\}}$ and $t^{\{\mathfrak{f}\}}$ can be evaluated online by timing the slow macro-steps and the fast micro-steps during their execution. The algorithm adjusts automatically if these compute times vary during the application lifetime.

3.8 New high-order MrGARK methods

Using the design process outlined above we construct several MrGARK methods for use in practical applications. We use the naming convention *FASTf-SLOWs p(p̂)[type]*, where p

is the method order, $\widehat{p}$ is the embedded order, f is the number of stages in the fast base method, and s is the number of stages in the slow base method. Each component method is either explicit or implicit: FAST, SLOW $\in$ {EX,IM}. We distinguish between methods of type A (optimized for accuracy and for better step size control) and methods of type S (optimized for simplicity and for stability), therefore *type* $\in$ {A,S}. The newly developed methods are as follows:

- EX2-EX2 2(1)[A], EX2-EX2 2(1)[S], EX3-EX3 3(2)[A], EX4-EX4 3(2)[A], EX3-EX3 3(2)[S], and EX5-EX5 4(3)[A] are explicit-explicit, schemes of order two, three, three and four respectively.
- EX2-IM2 2(1)[A], EX3-IM3 3(2)[A], and EX6-IM5 4(3)[A] are methods with an explicit fast part and an implicit slow part of order two, three, and four respectively.
- IM2-EX2 2(1)[A], IM3-EX3 3(2)[A], and IM6-EX5 4(3)[A] are methods with an implicit fast part and an explicit slow part of order two, three, and four, respectively.

The coefficients of these methods are given in section B.1.

3.9 Numerical experiments

In this section we carry out numerical experiments to validate and test the newly derived MrGARK methods. The choice of the partitioning is based on prior knowledge of the computational cost and stiffness of each partition. Readers interested in automatic partitioning of the right hand side function may refer to [51, p.21]

3.9.1 Additive partitioning tests

The first experiment is carried out using a two-dimensional unsteady convection-diffusion equation [34, Ch. 3] in a square spatial domain $\Omega = \{0 \leq x, y \leq 1\}$ and with a circular wind profile:

$$u_t - \varepsilon \nabla^2 u + w \cdot \nabla u = 0 \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega, \quad w = \begin{bmatrix} 2y(1-x^2) \\ -2x(1-y^2) \end{bmatrix}. \tag{3.30a}$$

A Streamline Upwind Petrov-Galerkin (SUPG) spatial discretization is used, which leads to a semi-discrete system of linear ODEs:

$$\mathbf{M}^h u_t^h = \mathbf{A}\, u^h + (\vec{n} + \vec{n}^{stab})\, u^h, \tag{3.30b}$$

where $\mathbf{M}^h$ and $\mathbf{A}$ are mass and stiffness matrices and $\vec{n} + \vec{n}^{stab}$ represent linear forms of the convective term and SUPG stabilization.

In the multirate experiments we designate the first term in eq. (3.30b) as the slow component, $f^{\{\mathfrak{s}\}} = (\mathbf{M}^h)^{-1}\,\mathbf{A}\,u^h$, and the second term as the fast component, $f^{\{\mathfrak{f}\}} = (\mathbf{M}^h)^{-1}\,(\vec{n}+\vec{n}^{stab})\,u^h$. In practice the splitting choice is informed by inspecting the spectral radius of the right hand side operators. The weak form of the PDE and the corresponding MrGARK schemes are implemented in the FEniCS package [3], which is used to carry out the convergence experiments.

The convergence diagrams for this test are shown in fig. 3.4. The numerical orders of convergence for all schemes match their theoretical orders for the multirate step ratios tested. Figure 3.5 shows the evolution of the model solution over time.

3.9.2 Timing experiments

Timing experiments are performed in MATLAB using the Gray-Scott model [72]. Here, we are interested in the application of MrGARK methods to additive splittings of the right hand side into linear and nonlinear terms. This reaction-diffusion PDE is:

$$\begin{cases} u_t = \nabla \cdot (\varepsilon_u\,\nabla u) - u\,v^2 + \mathfrak{f}(1-u), \\ v_t = \nabla \cdot (\varepsilon_v\,\nabla v) + u\,v^2 - (\mathfrak{f}+\mathfrak{k})\,v. \end{cases} \tag{3.31}$$

The domain is the unit square discretized with second order finite differences. The reaction parameters are $\mathfrak{k} = 0.0520$ and $\mathfrak{f} = 0.0180$.

In the following experiments the nonlinear reaction terms on the right hand side are considered the fast system, and the diffusion terms are regarded as the slow one.

A first version of the model, used to test explicit MrGARK methods, includes nonlinear diffusion terms:

$$\varepsilon_u = 0.0625\; e^{-\frac{u}{100}}\sin(\pi x)\sin(\pi y), \quad \varepsilon_v = 0.0312\; e^{-\frac{v}{100}}\sin(\pi x)\sin(\pi y). \tag{3.32}$$

Figure 3.6 presents the evolution of quantity u over time. Figures 3.7a and 3.7b present performance diagrams for a fixed-step time integration of Gray-Scott model with nonlinear diffusion. The MrGARK method EX3-EX3 3(2)[A] from section B.1.5 with different multirate step ratios is used. Notice that as M increases, it is possible to increase the macro-step size without violating the CFL conditions. The results shown in fig. 3.7a indicate that, for a fixed error level, the multirate method shows better performance than single rate one for $M = 2$ and 3.

A second version of the model, used to test EX-IM MrGARK methods, includes linear diffusive terms with parameters $\varepsilon_u = 0.0625$ and $\varepsilon_v = 0.0312$. The diffusion is considered the slow process and is treated implicitly; the constant diffusion operator is leveraged in the

solution of linear stage equations. Figures 3.7c and 3.7d show the performance diagrams for the MrGARK method EX3-IM3 3(2)[A] from section B.1.8 compared to single rate implicit method of the same order. In fig. 3.7d we note the reduced order of convergence for the single rate implicit method. Being a DIRK scheme, this method has a Newton-Krylov iteration for each stage that involves Jacobian-vector products of the full right hand side. Here, as in many practical cases, these are approximated by finite differences. Therefore, the quality of the approximation and the Krylov solver affect the convergence rate and efficiency of the SR-IM method [102]. On the other hand, the stages for multirate methods use only the Jacobian of the linear term, and show full order of convergence and faster computations than both single rate implicit and explicit methods. As M increases the performance of the multirate method improves incurring negligible increase in error.

3.9.3 Numerical experiments for H and M adaptivity

We experiment with different adaptivity strategies described in section 3.7.4 using the Gray-Scott model eq. (3.31) with nonlinear diffusion term eq. (3.32). All experiments in this section are performed for a time span of $T = [0, 2]$ seconds.

Figure 3.8 shows the automatically selected macro-step size (H) and multirate step ratio (M) for the naturally adaptive method EX4-EX4 3(2)[A]. The efficiency optimization strategy (section 3.7.4) is used to select both H and M. As the relative cost of evaluating the fast and slow right hand side functions changes in figs. 3.8a to 3.8c, the choice of M varies to keep the overall efficiency high. The costs of slow and fast system are computed by timing their respective right hand side evaluations. The one-dimensional integer optimization in eq. (3.29) is then approximated by limiting the possible increase or decrease of M_{new} to

$$\max(1, M - 1) \leq M_{\text{new}} \leq M + 2,$$

which reduces the computational cost of the optimization and increases the robustness of the algorithm.

In fig. 3.9 the adaptivity strategy based on balancing the fast and slow errors (section 3.7.4) is tested. By interchanging the roles of the fast and slow partitions the choice of multirate step ratio M changes from the maximum allowed value of ten to its minimum allowed value of two, in an attempt to keep the fast and slow errors balanced. The macro-step size selection by the traditional error controller is the same in both cases.

Finally, it is instructive to see how different H-M adaptivity strategies developed in section 3.7.4 compare against the conventional H adaptivity. We carry out this experiment using the Gray-Scott model. Figure 3.10a shows the error in the final solution scaled by the computation time when the EX2-EX2 2(1)[A] method from section B.1.1 is used. Figure 3.10a repeats the experiment using the EX5-EX5 4(3)[A] method from section B.1.10. In these experiments the efficient single rate base method is employed when the adaptive strategy selects $M = 1$. The parameters of the adaptivity algorithm such as macro-step

rejection factor are optimized for maximum efficiency. The results indicate that the H-M adaptivity strategies performed better than the classical H error control.

3.9.4 Component partitioning experiment

We consider the BSVD reaction-diffusion problem on the unit square Ω [53] with boundary $\partial\Omega$ and outward boundary normal vector $\vec{n}$:

$$u_t = \nabla \cdot (D(x,y)\nabla u) + 10(1-u^2)(u+0.6), \tag{3.33a}$$

$$u(x,y,0) = 2\exp\big\{\big(-10(x-0.5)^2 - 10(y+0.1)^2\big)\big\}, \qquad x,y \in \Omega, \tag{3.33b}$$

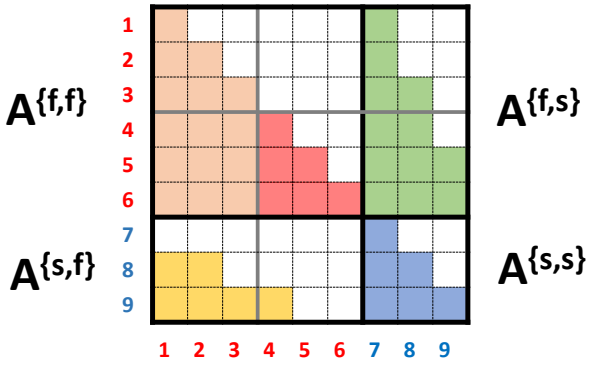
$$D(x,y) = 0.1\sum_{i=1}^{3} e^{-100(x-0.5)^2+(y-y_i)^2}, \tag{3.33c}$$

$$D(x,y)\nabla u \cdot \vec{n} = 0, \qquad x,y \in \partial\Omega,\ t \in [0,t_F]. \tag{3.33d}$$

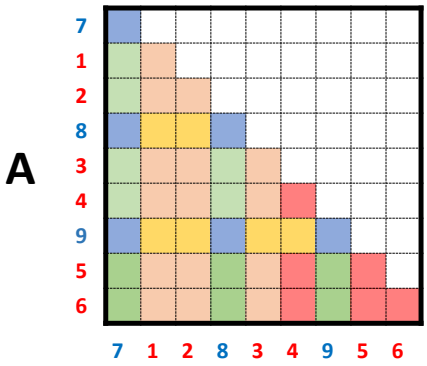
The parameters values are $y_1 = 0.6, y_2 = 0.75, y_3 = 0.9$ as prescribed in [53]. The domain is partitioned into two sub-domains:

$$\Omega_1 = \{(x,y) \in \Omega \ :\ |x-0.5| \leq 0.125,\ y \leq 0.125\}, \qquad \Omega_2 = \Omega \backslash \Omega_1.$$

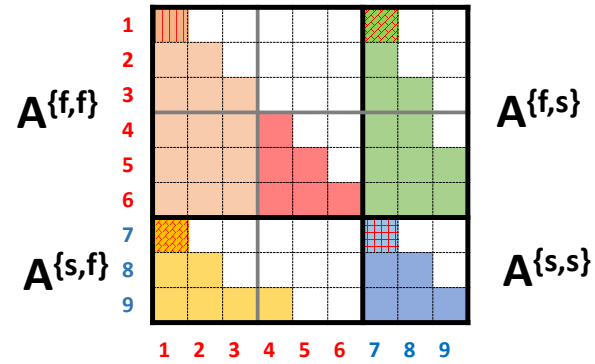
A continuous finite element semi-discretization in space with Lagrange polynomial basis of order four leads to a two-way partitioned system of ODEs. We discretized this model using the FEniCS package [3] and the evolution of its solution over the time span of $T = [0,5]$ seconds is shown in fig. 3.11. The subsystem with fewer degrees of freedom (DOFs) is designated as the fast one, and the remaining variables are considered to be the slow system. In this experiment the ratio of slow to fast DOFs is 28. fig. 3.12 shows the error for fixed macro-step time integration using explicit-explicit multirate methods of orders 2, 3 and 4. The largest macro-steps are chosen close to the maximum stable step size for the method. Results reported in fig. 3.12 show that, although the numerical order of convergence of the methods fluctuate slightly, they follow their theoretical values closely. Note that the errors are reported only for stable macro-step sizes for different methods and multirate step ratios.



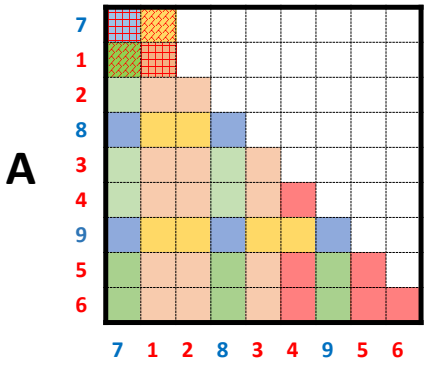
(a) Butcher tableau of a decoupled MrGARK. The coupling matrices have complementary sparsity patterns.



(b) Permuted Butcher tableau of a decoupled MrGARK shows a lower diagonal structure.



(c) Butcher tableau of a coupled MrGARK. The dashed entries of the coupling matrices violate the sparsity complementarity.



(d) Permuted Butcher tableau of a coupled MrGARK. The first slow and the first fast steps, dashed, need to computed together in a coupled manner.

FIGURE 3.1: Example of decoupled and coupled MrGARK with $M = 2$ and $s = 3$. Blue is the slow method, pink the first fast step, dark pink the second fast step, and green and yellow are the couplings. The permuted versions of the tableaus reflect the sequential order of stage computations. Note the entry above the diagonal for the coupled, permuted tableau due to the non-complementary coupling structure.

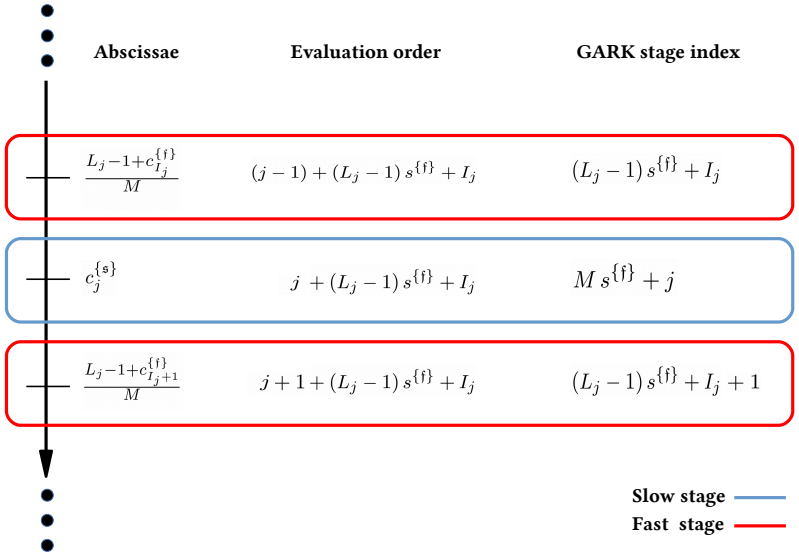
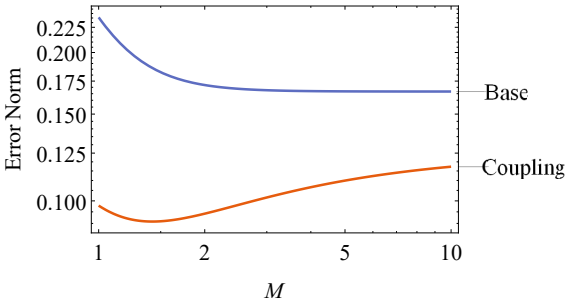
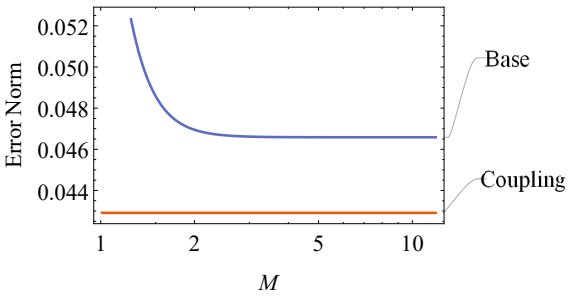


FIGURE 3.2: An example of the order of computing stages in MrGARK with stage sequence $\{(L_j, I_j)\}$.



(a) EX2-EX2 2(1)[A]. The base error is $\mathcal{O}(H^3)$, but the coupling is $\mathcal{O}(H^4)$ since the method is naturally adaptive.



(b) EX3-EX3 3(2)[A]. Both the base and coupling errors are $\mathcal{O}(H^4)$.

FIGURE 3.3: Behavior of the local truncation error components for two MrGARK schemes as the multirate ratio M increases.

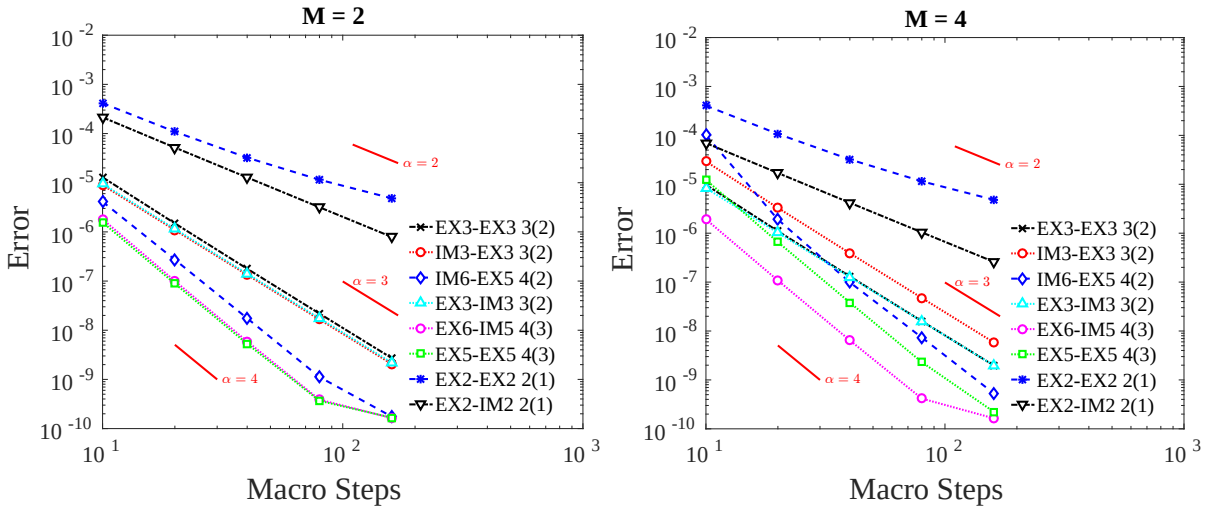


FIGURE 3.4: Convergence plots for the unsteady convection-diffusion test (3.30) over the time span $T = [0, 10]$ seconds. A fixed macro-step time integration is carried out with varying multirate step ratios M using MrGARK type A methods.

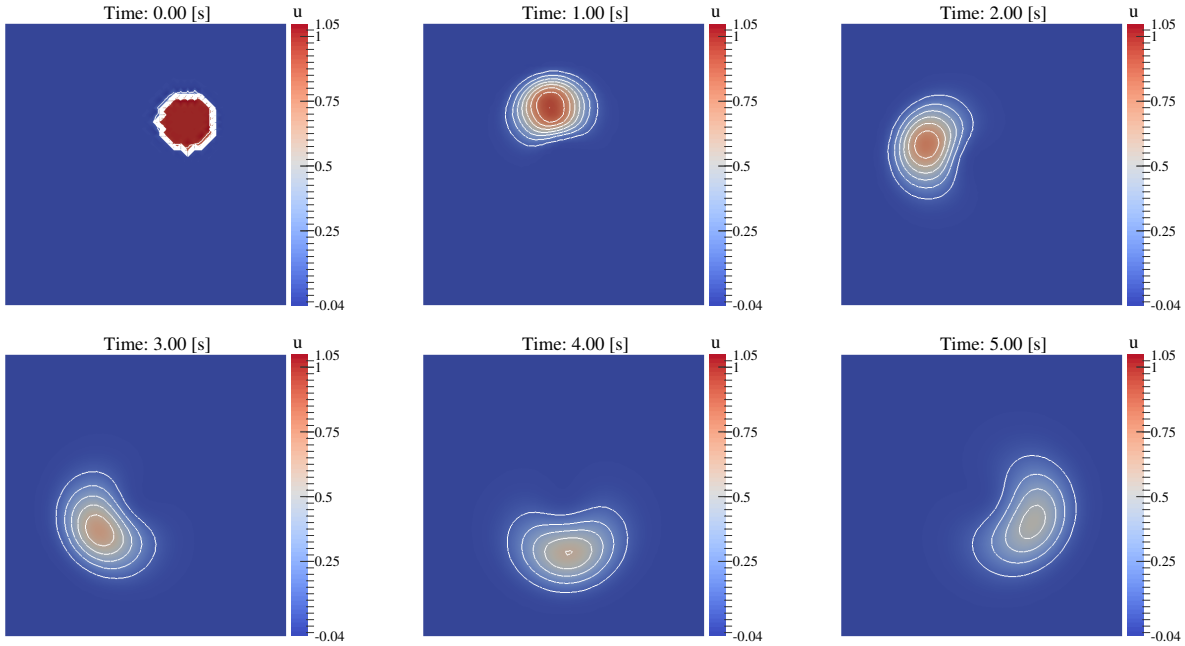


FIGURE 3.5: Evolution of the unsteady convection-diffusion problem (3.30) solution in time.

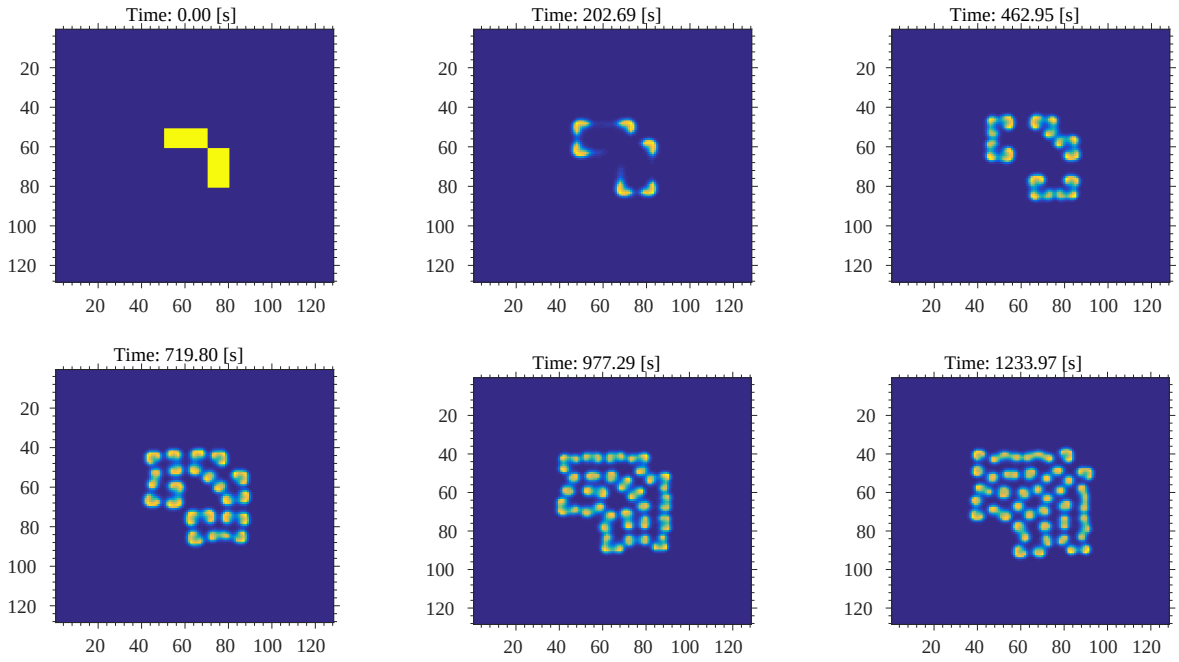
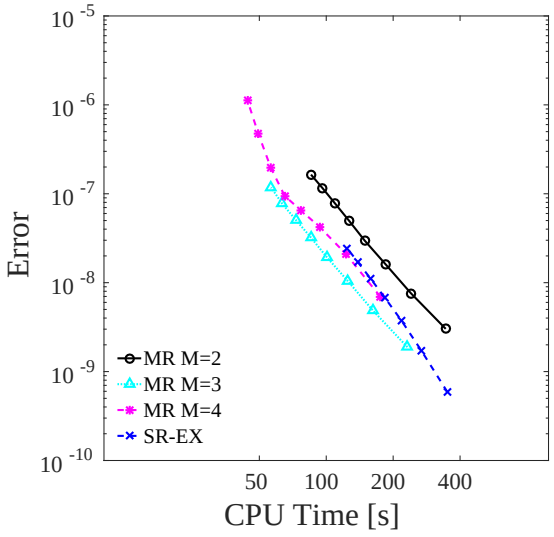
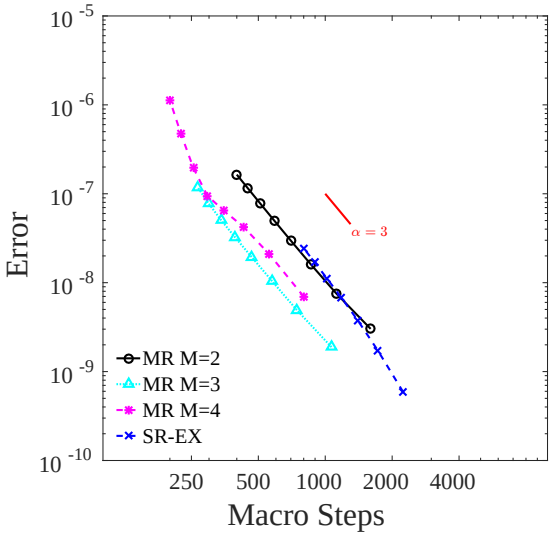


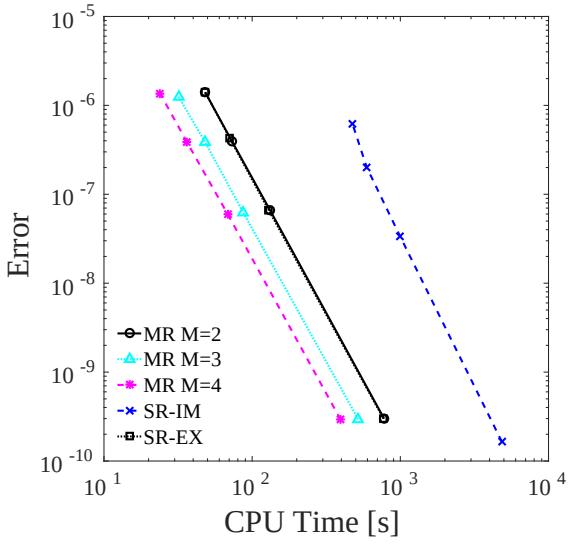
Figure 3.6: Evolution of Gray-Scott model (3.31) solution u in time. A nonlinear diffusion is used, and integration is performed with explicit MrGARK methods.



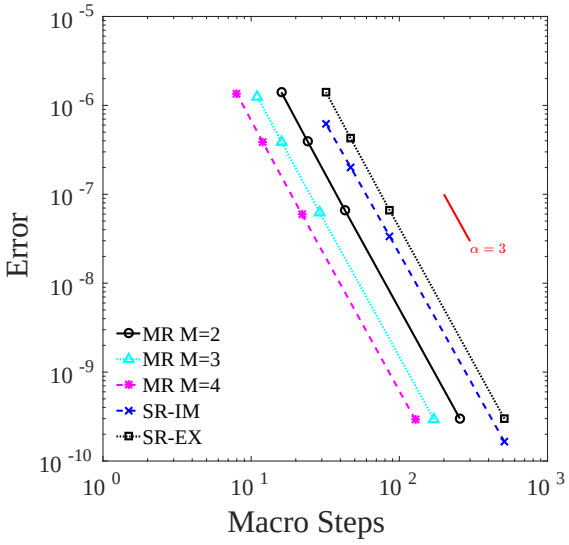
(a) Convergence plot for EX3-EX3 3(2)[A]



(b) Performance plot for EX3-EX3 3(2)[A]

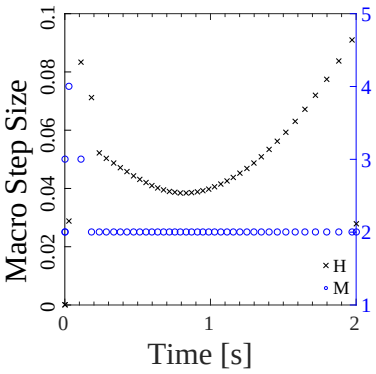


(c) Performance plot for EX3-IM3 3(2)[A]

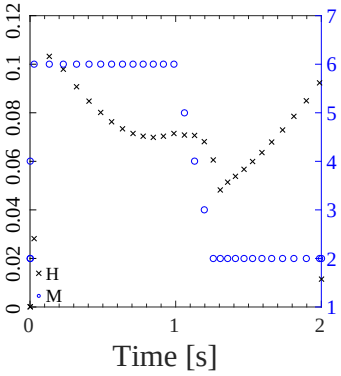


(d) Convergence plot for EX3-IM3 3(2)[A]

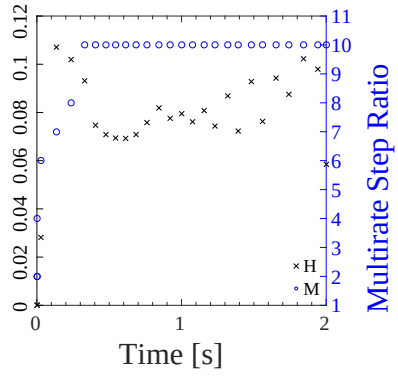
FIGURE 3.7: Results for MrGARK schemes applied to Gray-Scott model eq. (3.31).



(a) $t^{\{s\}}/t^{\{f\}} = 15$.

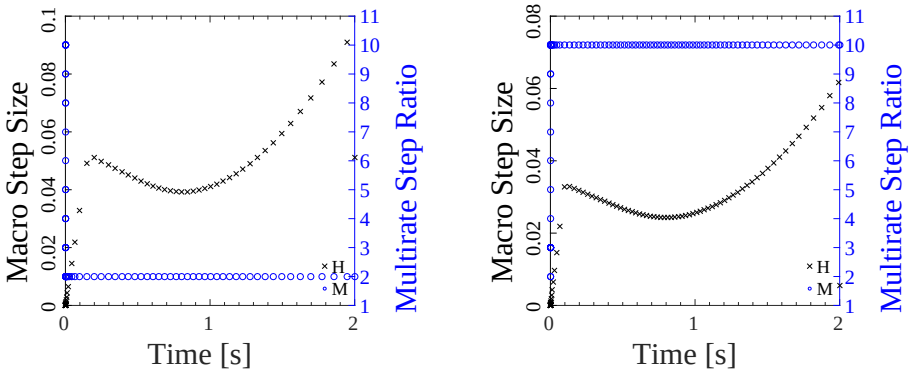


(b) $t^{\{s\}}/t^{\{f\}} = 20$.



(c) $t^{\{s\}}/t^{\{f\}} = 25$.

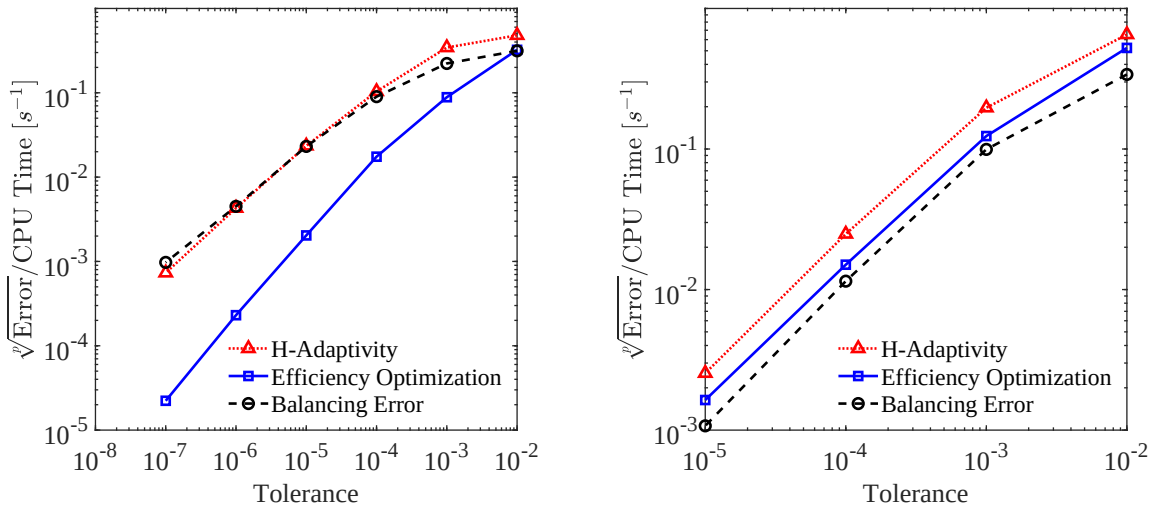
FIGURE 3.8: Automatically selected macro-step size and multirate step ratio for the Gray-Scott model eq. (3.31) integrated over the time span $T = [0, 2]$ using EX4-EX4 3(2)[]A] method. The efficiency optimization strategy (section 3.7.4) is used with AbsTol = RelTol = 10^{-4}.



(a) Gray-Scott model with reaction as the fast system

(b) Gray-Scott model with diffusion as the fast system

FIGURE 3.9: Automatically selected macro-step size and multirate step ratio for the Gray-Scott model eq. (3.31) integrated over the time span $T = [0, 2]$ seconds using EX4-EX4 3(2)[A] method. The strategy of balancing the slow and fast errors (section 3.7.4) is used with AbsTol = RelTol = 10^{-2}.



(a) EX2-EX2 2(1)[A] method.

(b) EX5-EX5 4(3)[A] method.

FIGURE 3.10: Efficiency of different adaptivity strategies for explicit-explicit methods applied to Gray-Scott model with a compute time ratio $t^{\{s\}}/t^{\{f\}} = 2$.

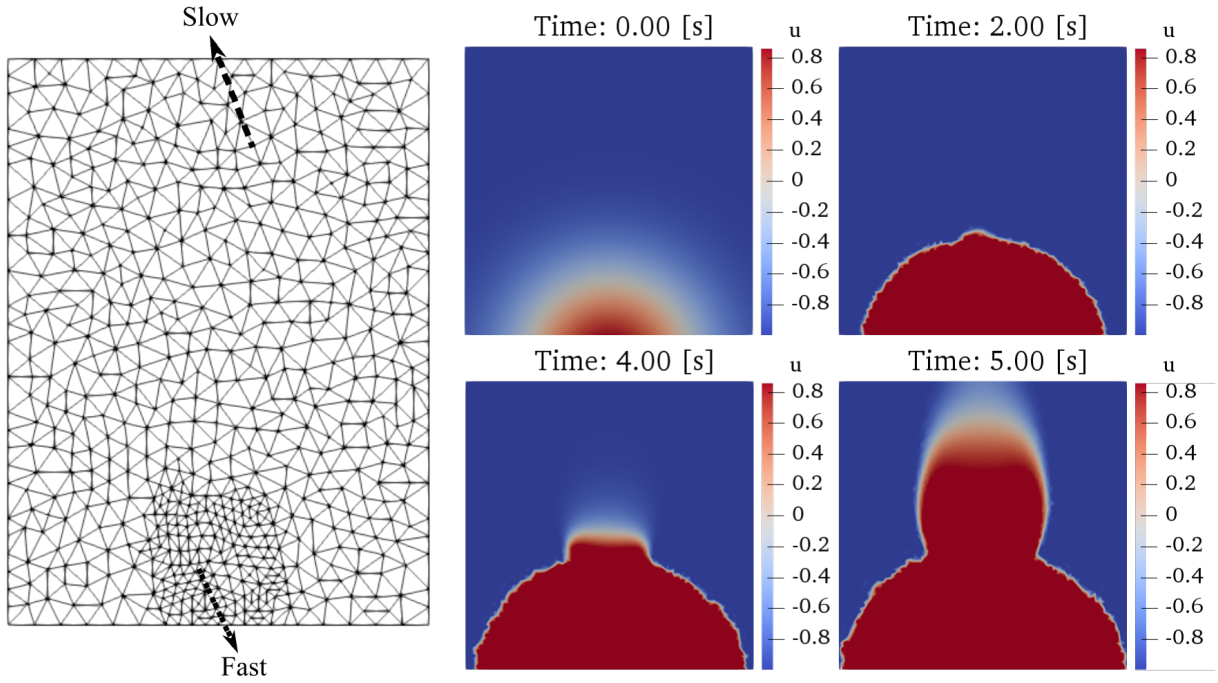


FIGURE 3.11: Evolution of the BSVD problem (3.33) solution in time.

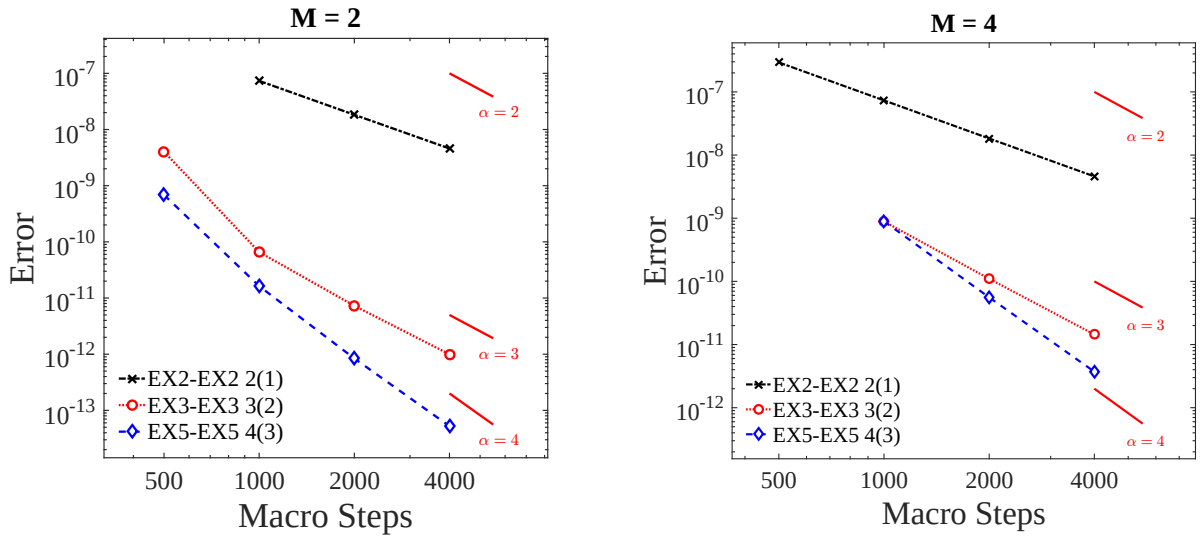


FIGURE 3.12: Convergence results for BSVD test eq. (3.33) over the time span $T = [0, 0.2]$ seconds. A fixed macro-step time integration is carried out with varying multirate step ratios M using explicit MrGARK type A methods.

Chapter 4

Alternating directions implicit integration in GLM framework

4.1 Introduction

We are concerned with solving the initial value problem:

$$y'(t) = f(y) = \sum_{\sigma=1}^{N} f^{\{\sigma\}}(y), \quad y(t_0) = y_0, \tag{4.1}$$

where the right hand side function $f : \mathbb{R}^d \to \mathbb{R}^d$ is additively split into N partitions. Systems such as eq. (4.1) emerge from method of lines semi-discretization of PDEs when all spatial derivatives are approximated by their discretization. In many cases, the right hand side function includes discrete self-adjoint operators performing spatial derivatives in different directions. The sparsity structure of these operators is similar, for example, in the case when a fixed-stencil finite difference method is used to resolve spatial derivatives. Implicit time-stepping methods are preferred to propagate stiff differential equations in time, but they require working with large Jacobian matrices. Implicit-Explicit (IMEX) methods allow us to integrate non-stiff parts of the system more efficiently, however, more can be achieved by designing specialized time-stepping methods for certain classes of problems. Depending on the choice of discretization, we can use the tensor product structure of derivative operators to only work with one-dimensional Jacobian matrices much smaller than the full Jacobian, applying directional derivatives in different directions in turn.

Alternating Directions Implicit (ADI) schemes for parabolic problems were first introduced in the works of Douglas [32], Douglas and Rachford [33], and Peaceman and Rachford [79]. Closely related to this field is the body of work on operator splitting schemes [119, 129, 130] and Approximate Matrix Factorizations (AMF) applied to Rosenbrock-W [42, 43, 44] and LIRK methods [133]. Another important development is the Fractional Step Runge–Kutta framework [14, 15] investigating the link between directional methods and IMEX schemes.

Early analysis of convergence of stiff ODEs can be traced back to Prothero-Robinson [82]. Ostermann *et* al. formally show the fractional order phenomenon is related to changes in the behavior of local truncation error in stiff systems [77]. Methods of high stage order are known to alleviate this drawback [11, 81]. The General Linear Method (GLM) framework

[18, 22, 59] encompasses many of these methods and facilitates creation of new ones for novel applications. The theory of partitioned GLMs was formalized in [132] and different families of methods based on this structure have been reported in [25, 131, 134, 136]. More recent high order IMEX-GLMs found in the literature [58, 106, 107] are based on Diagonally Implicit Multistage Integration Methods (DIMSIMs), Two-Step Runge–Kutta methods, and Peer methods providing various accuracy and stability enhancements.

The goal of this chapter is to extend the capabilities of ADI schemes to high order GLMs, creating methods resilient to order reduction while leveraging the efficiency of alternating implicit integration. The chapter is organized as follows: We start by reviewing the partitioned GLM framework in section 4.2, introduce the structure of ADI-GLMs in section 4.3, study their order conditions in section 4.4, and investigate their stability in section 4.5. We comment on design principles and implementation aspects in section 4.6 followed by numerical experiments in section 4.7. B.2 includes the coefficients of the new methods, and B.3 presents stability plots.

4.2 Traditional and partitioned General Linear Methods

A traditional GLM with s internal and r external stages represented by Butcher tableau:

$$\begin{array}{c|c|c} \mathbf{c} & \mathbf{A} & \mathbf{U} \\ \hline & \mathbf{B} & \mathbf{V} \end{array}, \tag{4.2}$$

advances the numerical solution to eq. (4.1) with timestep h according to:

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

$$\xi_i^{[n]} = h\sum_{j=1}^{s} b_{i,j} f(Y_j) + \sum_{j=1}^{r} v_{i,j} \xi_j^{[n-1]}, \quad i = 1, \dots, r, \tag{4.3b}$$

where the matrix notation of coefficients is used:

$$\begin{aligned} &\mathbf{A} := [a_{i,j}] \in \mathbb{R}^{s\times s}, \quad \mathbf{U} := [u_{i,j}] \in \mathbb{R}^{s\times r}, \quad \mathbf{B} := [b_{i,j}] \in \mathbb{R}^{r\times s}, \\ &\mathbf{V} := [v_{i,j}] \in \mathbb{R}^{r\times r}, \quad \mathbf{W} := [w_{i,j}] = [\mathbf{w}_0 \ \cdots \ \mathbf{w}_p] \in \mathbb{R}^{r\times (p+1)}, \end{aligned} \tag{4.4}$$

where matrix $\mathbf{W}$ determines the relation between external stages and derivatives of the exact solution such that for a method of order p:

$$\xi_i^{[n]} = \sum_{k=0}^{p} w_{i,k} h^k y^{(k)}(t_n) + \mathcal{O}\big(h^{p+1}\big).$$

GLM framework is extensive and well-established. Readers interested in theoretical foundation of these methods are referred to the literature [18, 22, 59].

IMEX-GLMs are extensions of traditional GLMs that treat partitions of the right hand side with different methods while keeping a single set of internal and external stages. One step of an IMEX-GLM formally reads as:

$$Y_i = h\sum_{\sigma=1}^{N}\sum_{j=1}^{s} a_{i,j}^{\{\sigma\}} f^{\{\sigma\}}(Y_j) + \sum_{j=1}^{r} u_{i,j}\,\xi_j^{[n-1]}, \quad i = 1,\dots,s, \tag{4.5a}$$

$$\xi_i^{[n]} = h\sum_{\sigma=1}^{N}\sum_{j=1}^{s} b_{i,j}^{\{\sigma\}} f^{\{\sigma\}}(Y_j) + \sum_{j=1}^{r} v_{i,j}\xi_j^{[n-1]}, \quad i = 1,\dots,r. \tag{4.5b}$$

4.3 Formulation of ADI-GLMs

We rely on the theory of IMEX-GLMs as reported in [25, 132, 134] to design partitioned GLMs suited for ADI integration. The goal is to construct GLMs that apply implicit integration to individual partitions of the right hand side function in eq. (4.1), while using an explicit coupling to the other components. We seek to achieve high stage order while benefiting from the low computational cost of directional implicit methods.

Definition 4.1 (ADI-GLM schemes)**.** One step of an N-way partitioned ADI-GLM applied to eq. (4.1) is defined as:

$$Y_i^{\{\mu\}} = h\sum_{\sigma=1}^{N}\sum_{j=1}^{s} a_{i,j}^{\{\mu,\sigma\}} f^{\{\sigma\}}(Y_j^{\{\sigma\}}) + \sum_{\sigma=1}^{N}\sum_{j=1}^{r} u_{i,j}^{\{\mu,\sigma\}}\xi_j^{\{\sigma\}[n-1]}, \tag{4.6a}$$
$$i = 1,\dots,s, \qquad \mu = 1,\dots,N,$$

$$\xi_i^{\{\mu\}[n]} = h\sum_{\sigma=1}^{N}\sum_{j=1}^{s} b_{i,j}^{\{\mu,\sigma\}} f^{\{\sigma\}}(Y_j^{\{\sigma\}}) + \sum_{\sigma=1}^{N}\sum_{j=1}^{r} v_{i,j}^{\{\mu,\sigma\}}\xi_j^{\{\sigma\}[n-1]}, \tag{4.6b}$$
$$i = 1,\dots,r, \qquad \mu = 1,\dots,N.$$

Here, we are interested in applying different combinations of explicit and diagonally implicit methods to the right hand side partitions and storing the resulting internal and external stages separately.

If the method is order p, the external stages are related to derivatives of y by:

$$\xi_i^{\{\mu\}[n]} = w_{i,0}^{\{\mu\}} y(t_n) + \sum_{\sigma=1}^{N}\sum_{k=1}^{p} w_{i,k}^{\{\mu,\sigma\}} h^k (f^{\{\sigma\}})^{(k-1)}\left(y\left(t_n\right)\right) + \mathcal{O}\left(h^{p+1}\right), \tag{4.7}$$

$$\mathbf{W}^{\{\mu,\sigma\}} := [\mathbf{w}_0^{\{\mu\}} \ \cdots \ \mathbf{w}_p^{\{\mu,\sigma\}}] \in \mathbb{R}^{r\times(p+1)}. \tag{4.8}$$

The method is stage order q if internal stages are approximations of the exact solution at abscissa points $\mathbf{c}^{\{\mu\}}$:

$$Y_i^{\{\mu\}} = y(t_{n-1} + \mathbf{c}_i^{\{\mu\}} h) + \mathcal{O}\big(h^{q+1}\big). \tag{4.9}$$

4.4 Construction of ADI-GLMs

We start by considering a pair of explicit and implicit GLMs with the same number of external and internal stages:

$$\begin{array}{c|c|c} \mathbf{c}^{\{E\}} & \mathbf{A}^{\{E\}} & \mathbf{U}^{\{E\}} \\ \hline & \mathbf{B}^{\{E\}} & \mathbf{V}^{\{E\}} \end{array}, \quad \begin{array}{c|c|c} \mathbf{c}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{U}^{\{I\}} \\ \hline & \mathbf{B}^{\{I\}} & \mathbf{V}^{\{I\}} \end{array}. \tag{4.10}$$

We construct ADI-GLMs using a collection of IMEX-GLMs each performing implicit integration in a specific direction. A preconsistent IMEX-GLM has order p and stage order $q \in \{p, p-1\}$ if and only if the following conditions hold:

$$\frac{\mathbf{c}^{\{\sigma\}\times k}}{k!} - \frac{\mathbf{A}\mathbf{c}^{\{\sigma\}\times(k-1)}}{(k-1)!} - \mathbf{U}^{\{\sigma\}}\mathbf{w}_k^{\{\sigma\}} = 0, \tag{4.11a}$$
$$k = \{1, \dots, q\}, \quad \sigma \in \{E, I\},$$
$$\sum_{l=0}^{k} \frac{\mathbf{w}_{k-l}^{\{\sigma\}}}{l!} - \frac{\mathbf{B}^{\{\sigma\}}\mathbf{c}^{\{\sigma\}\times(k-1)}}{(k-1)!} - \mathbf{V}^{\{\sigma\}}\mathbf{w}_k^{\{\sigma\}} = 0, \tag{4.11b}$$
$$k = \{1, \dots, p\}, \quad \sigma \in \{E, I\}.$$

The structure of the Butcher tableau for an ADI-GLM depends on the number of partitions and number of stiff partitions that require implicit treatment. Here, we focus on three practical examples and more elaborate designs follow the same principles. The Butcher tableau for a 3-way partitioned ADI-GLM with alternating implicit stages in all partitions is:

$$\begin{array}{c|ccc|ccc} \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{A}^{\{E\}} & \mathbf{U} & \mathbf{0} & \mathbf{0} \\ \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{0} & \mathbf{U} & \mathbf{0} \\ \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{0} & \mathbf{0} & \mathbf{U} \\ \hline & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{B}^{\{E\}} & \mathbf{V} & \mathbf{0} & \mathbf{0} \\ & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{0} & \mathbf{V} & \mathbf{0} \\ & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{0} & \mathbf{0} & \mathbf{V} \end{array}. \tag{4.12}$$

When only two partitions are stiff, the non-stiff partition is carried through explicitly:

$$\begin{array}{c|ccc|ccc} \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{A}^{\{E\}} & \mathbf{U} & \mathbf{0} & 0 \\ \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{0} & \mathbf{U} & 0 \\ \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{0} & 0 & \mathbf{U} \\ \hline & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{B}^{\{E\}} & \mathbf{V} & \mathbf{0} & 0 \\ & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{0} & \mathbf{V} & 0 \\ & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{0} & 0 & \mathbf{V} \end{array}. \tag{4.13}$$

We notice immediately that $Y_i^{\{3\}} \equiv Y_i^{\{2\}}$, therefore one only computes two types of stage vectors, and the second is used as an argument for the explicit integration of the third, non-stiff component.

In a similar fashion, a 2-way partitioned ADI-GLM is described by:

$$\begin{array}{c|cc|cc} \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{U} & \mathbf{0} \\ \mathbf{c} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{0} & \mathbf{U} \\ \hline & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{V} & \mathbf{0} \\ & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{0} & \mathbf{V} \end{array}. \tag{4.14}$$

Remark 4.2. Comparing eqs. (4.12) to (4.14) with eq. (4.6), notice that we have chosen:

$$\mathbf{c}^{\{E\}} = \mathbf{c}^{\{I\}} = \mathbf{c}, \tag{4.15a}$$

$$\mathbf{U}^{\{E\}} = \mathbf{U}^{\{I\}} = \mathbf{U}, \tag{4.15b}$$

$$\mathbf{V}^{\{E\}} = \mathbf{V}^{\{I\}} = \mathbf{V}. \tag{4.15c}$$

This selection is practically useful in creating IMEX-GLMs with unified internal stages. In the context of ADI-GLMs this choice allows us to keep the number of internal and external stages as low as the number of stiff partitions.

Remark 4.3. We have also decoupled computations involving the external stages:

$$\mathbf{U}^{\{\sigma,\mu\}} = \begin{cases} \mathbf{0} & \sigma \neq \mu \\ \mathbf{U} & \sigma = \mu \end{cases}, \qquad \mathbf{V}^{\{\sigma,\mu\}} = \begin{cases} \mathbf{0} & \sigma \neq \mu \\ \mathbf{V} & \sigma = \mu \end{cases}. \tag{4.16}$$

Theorem 4.4. *The ADI-GLMs eq. (4.6) subject to eq. (4.15) and eq. (4.16) is stage order q and order p, hereafter denoted by order (q,p), if and only if individual methods (4.10) are order (q,p).*

Proof. We first assume that the ADI-GLM is order (q,p) such that eqs. (4.7) and (4.9) hold. Since all internal stages $Y_i^{\{\sigma\}}$ share the same abscissa, from eq. (4.9) we have:

$$Y_i^{\{\sigma\}} = Y_i^{\{\mu\}} + \mathcal{O}\big(h^{q+1}\big), \qquad \sigma,\mu \in \{1,\dots,N\}. \tag{4.17}$$

Therefore, we can replace $Y_j^{\{\sigma\}}$ with $Y_j^{\{\mu\}}$ in eq. (4.6a) without changing the order. The resulting method is an IMEX-GLM with

$$\begin{array}{c|ccc|c} \mathbf{c} & \mathbf{A}^{\{\mu,1\}} & \dots & \mathbf{A}^{\{\mu,N\}} & \mathbf{U} \\ \hline & \mathbf{B}^{\{\mu,1\}} & \dots & \mathbf{B}^{\{\mu,N\}} & \mathbf{V} \end{array}, \qquad \mu \in \{1,\dots,N\}. \tag{4.18}$$

From IMEX-GLM order conditions [132, 135] method (4.18) is order (q,p) if and only if individual methods

$$\begin{array}{c|c|c} \mathbf{c} & \mathbf{A}^{\{\mu,\sigma\}} & \mathbf{U} \\ \hline & \mathbf{B}^{\{\mu,\sigma\}} & \mathbf{V} \end{array}, \qquad \mu \in \{1,\dots,N\}, \qquad \sigma \in \{1,\dots,N\}.$$

are order (q,p). This means that the methods in eq. (4.10) have to be order (q,p).

The *if* part of the theorem can be proven along the same line of reasoning. Assuming individual methods (4.10) are order (q,p) the IMEX-GLM (4.18) is order (q,p). Internal stage values in eq. (4.5a) can be replaced by an approximation of the same order as in eq. (4.17) to create the internal stages for ADI-GLM. Since the order of internal stages has not changed, external stages also remain order p. This concludes the proof. □

Remark 4.5. A corollary to proposition 4.4 is that in the case of ADI-GLM (4.13), we can forgo computing $\left(Y_i^{\{3\}}, \xi^{\{3\}[n]}\right)$ stages without losing accuracy. Furthermore, this choice will not affect the stability since the stiff partitions are still treated implicitly and the integration of the non-stiff partition already appears in stage computations.

4.5 Stability of ADI-GLMs

Applying the ADI-GLM (4.12) to the linear scalar test equation:

$$u' = \lambda_x u + \lambda_y u + \lambda_z u, \tag{4.19}$$

and using eq. (4.6) leads to the following directional stages:

$$Y^{\{1\}} = \eta_x \mathbf{A}^{\{I\}} Y^{\{1\}} + \eta_y \mathbf{A}^{\{E\}} Y^{\{2\}} + \eta_z \mathbf{A}^{\{E\}} Y^{\{3\}} + \mathbf{U}\xi^{\{1\}[n-1]}, \tag{4.20a}$$
$$Y^{\{2\}} = \eta_x \mathbf{A}^{\{I\}} Y^{\{1\}} + \eta_y \mathbf{A}^{\{I\}} Y^{\{2\}} + \eta_z \mathbf{A}^{\{E\}} Y^{\{3\}} + \mathbf{U}\xi^{\{2\}[n-1]}, \tag{4.20b}$$
$$Y^{\{3\}} = \eta_x \mathbf{A}^{\{I\}} Y^{\{1\}} + \eta_y \mathbf{A}^{\{I\}} Y^{\{2\}} + \eta_z \mathbf{A}^{\{I\}} Y^{\{3\}} + \mathbf{U}\xi^{\{3\}[n-1]}, \tag{4.20c}$$
$$\xi^{\{1\}[n]} = \eta_x \mathbf{B}^{\{I\}} Y^{\{1\}} + \eta_y \mathbf{B}^{\{E\}} Y^{\{2\}} + \eta_z \mathbf{B}^{\{E\}} Y^{\{3\}} + \mathbf{V}\xi^{\{1\}[n-1]}, \tag{4.20d}$$
$$\xi^{\{2\}[n]} = \eta_x \mathbf{B}^{\{I\}} Y^{\{1\}} + \eta_y \mathbf{B}^{\{E\}} Y^{\{2\}} + \eta_z \mathbf{B}^{\{E\}} Y^{\{3\}} + \mathbf{V}\xi^{\{2\}[n-1]}, \tag{4.20e}$$
$$\xi^{\{3\}[n]} = \eta_x \mathbf{B}^{\{I\}} Y^{\{1\}} + \eta_y \mathbf{B}^{\{I\}} Y^{\{2\}} + \eta_z \mathbf{B}^{\{I\}} Y^{\{3\}} + \mathbf{V}\xi^{\{3\}[n-1]}, \tag{4.20f}$$

where $\eta_x = h\lambda_x, \eta_y = h\lambda_y, \eta_z = h\lambda_z$. Defining auxiliary notations $\mathbf{Z} = \text{blkdiag}\left(\eta_x \mathbf{I}_{s\times s}, \eta_y \mathbf{I}_{s\times s}, \eta_z \mathbf{I}_{s\times s}\right)$ and $\xi^{[n]} = \left(\xi^{\{1\}[n]}, \xi^{\{2\}[n]}, \xi^{\{3\}[n]}\right)^T$, the stability matrix is defined as:

$$\xi^{[n]} = \mathbf{M}(\eta_x, \eta_y, \eta_z)\, \xi^{[n-1]}, \tag{4.21a}$$

$$\mathbf{M}(\eta_x, \eta_y, \eta_z) = \widetilde{\mathbf{V}} + \widetilde{\mathbf{B}}\, \mathbf{Z}\Big(\mathbf{I}_{3s\times 3s} - \widetilde{\mathbf{A}}\mathbf{Z}\Big)^{-1}\widetilde{\mathbf{U}}, \tag{4.21b}$$

where:

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} & \mathbf{A}^{\{E\}} \\ \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{E\}} \\ \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} & \mathbf{A}^{\{I\}} \end{bmatrix}, \qquad \widetilde{\mathbf{U}} = \mathbf{I}_{3\times 3} \otimes \mathbf{U}, \tag{4.22a}$$

$$\widetilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} & \mathbf{B}^{\{E\}} \\ \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{E\}} \\ \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} & \mathbf{B}^{\{I\}} \end{bmatrix}, \qquad \widetilde{\mathbf{V}} = \mathbf{I}_{3\times 3} \otimes \mathbf{V}. \tag{4.22b}$$

When the eigenvalues of the system (4.19) are equal in all directions such that $\eta_x = \eta_y = \eta_z = \eta$ the stability matrix becomes:

$$\widehat{\mathbf{M}}(\eta) = \mathbf{M}(\eta, \eta, \eta) = \widetilde{\mathbf{V}} + \eta\widetilde{\mathbf{B}}\Big(\mathbf{I}_{3s\times 3s} - \eta\widetilde{\mathbf{A}}\Big)^{-1}\widetilde{\mathbf{U}}. \tag{4.23}$$

Equation (4.23) provides practical means for assessment and optimization of stability of ADI-GLMs.

Remark 4.6. The stability regions for individual explicit and implicit methods are defined as:

$$\mathcal{S}^{\{\sigma\}} = \left\{\eta \in \mathbb{C} :\ \mathbf{M}^{\{\sigma\}}(\eta) \text{ power bounded}\right\}, \tag{4.24a}$$

$$\mathbf{M}^{\{\sigma\}}(\eta) = \mathbf{V}^{\{\sigma\}} + \eta\mathbf{B}^{\{\sigma\}}\left(\mathbf{I}_{s\times s} - \eta\mathbf{A}^{\{\sigma\}}\right)^{-1}\mathbf{U}^{\{\sigma\}}, \quad \sigma \in \{E, I\}. \tag{4.24b}$$

The stability region of a 3-way partition method is defined as:

$$S = \{\eta_x, \eta_y, \eta_z \in \mathbb{C} :\ \mathbf{M}(\eta_x, \eta_y, \eta_z) \text{ power bounded}\}. \tag{4.25}$$

Remark 4.7. To investigate the stability of ADI-GLMs we define real and complex stability regions as:

$$\mathcal{S}_{\text{Real}} = \{\eta_x, \eta_y \in \mathbb{R} :\ \mathbf{M}(\eta_x, \eta_y, \max(\eta_x, \eta_y)) \text{ power bounded}\}, \tag{4.26a}$$

$$\mathcal{S}_{\text{Cplx}} = \left\{\eta \in \mathbb{C} :\ \widehat{\mathbf{M}}(\eta) \text{ power bounded}\right\}. \tag{4.26b}$$

Remark 4.8 (Stability as all partitions become infinitely stiff)**.** Consider the stability matrix eq. (4.23) when the eigenvalues in each direction simultaneously approach $-\infty$:

$$\lim_{\eta\to-\infty}\widehat{\mathbf{M}}(\eta)=\begin{bmatrix}\mathbf{V}^{\{I\}}-\left(\mathbf{B}^{\{I\}}-\mathbf{B}^{\{E\}}\right)\left(\mathbf{A}^{\{I\}}-\mathbf{A}^{\{E\}}\right)^{-1}\mathbf{U} & * \\ \mathbf{0} & \mathbf{M}^{\{I\}}(-\infty)\end{bmatrix}. \tag{4.27}$$

Due to the block triangular structure of this matrix, the eigenvalues of eq. (4.27) are the eigenvalues of the diagonal blocks and the entries in the upper right block can be ignored.

Consider the case $p=q=r=s$. We will further assume $\mathbf{w}_0^{\{I\}}=\mathbf{w}_0^{\{E\}}$, which comes at no loss of generality since we can always pick an equivalent formulation of the base methods where this holds. Using the difference of the order conditions of the base methods, we have that

$$\left(\mathbf{A}^{\{I\}}-\mathbf{A}^{\{E\}}\right)\mathbf{C}+\mathbf{U}\left(\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}\right)=\mathbf{0}, \tag{4.28a}$$

$$\left(\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}\right)\boldsymbol{\mu}-\left(\mathbf{B}^{\{I\}}-\mathbf{B}^{\{E\}}\right)\mathbf{C}-\mathbf{V}\left(\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}\right)=\mathbf{0}, \tag{4.28b}$$

where

$$\mu_{i,j}=\begin{cases}0 & i>j\\ \frac{1}{(j-i)!} & i\le j\end{cases},\qquad \mathbf{C}=\begin{bmatrix}\mathbb{1}_s & \mathbf{c} & \frac{\mathbf{c}^2}{2} & \cdots & \frac{\mathbf{c}^{p-1}}{(p-1)!}\end{bmatrix}. \tag{4.29}$$

Now we have that

$$\begin{aligned}&\left(\mathbf{V}^{\{I\}}-\left(\mathbf{B}^{\{I\}}-\mathbf{B}^{\{E\}}\right)\left(\mathbf{A}^{\{I\}}-\mathbf{A}^{\{E\}}\right)^{-1}\mathbf{U}\right)\left(\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}\right)\\ &=\mathbf{V}^{\{I\}}\left(\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}\right)+\left(\mathbf{B}^{\{I\}}-\mathbf{B}^{\{E\}}\right)\mathbf{C}\\ &=\left(\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}\right)\boldsymbol{\mu}.\end{aligned} \tag{4.30}$$

Thus, the upper left block of eq. (4.27) is similar to $\boldsymbol{\mu}$ provided $\mathbf{W}^{\{I\}}_{:,1:p}-\mathbf{W}^{\{E\}}_{:,1:p}$ is non-singular. In this case, eq. (4.27) is not power bounded because the 1 eigenvalue of $\boldsymbol{\mu}$ is defective. We note that this is not an issue when only a single eigenvalue becomes infinitely stiff.

In B.3 we provide plots of different stability regions for ADI-GLMs.

4.6 Design and implementation of ADI-GLMs

We have chosen the GLMs to be DIMSIMs [19] in order to reduce the number of free parameters in the design and simplify the order conditions. We require:

$$a_{i,i}^{\{I\}}=\gamma,\quad a_{i,i}^{\{E\}}=0,\quad a_{i,j}^{\{\sigma\}}=0,\quad \text{for}\quad j>i,\qquad \sigma\in\{E,I\}, \tag{4.31a}$$

$$\mathbf{U}^{\{\sigma\}}=\mathbf{I}_{s\times r},\qquad \sigma\in\{E,I\}, \tag{4.31b}$$

$$\mathbf{V}^{\{\sigma\}}=\mathbb{1}_r^T v,\quad v^T\mathbb{1}_r=1,\qquad \sigma\in\{E,I\}. \tag{4.31c}$$

linearly implicit GLMs derived here have $p = q = r = s$. The design process starts with choosing the abscissa vector $\mathbf{c}$. The remaining free parameters are coefficients of $\mathbf{A}^{\{E\}}$, $\mathbf{A}^{\{I\}}$, and v. For the new second and third order schemes, we picked existing, L-stable, type 2 DIMSIMs for $\mathbf{A}^{\{I\}}$ and v. Then, we choose $\mathbf{A}^{\{E\}}$ by numerically optimizing the area of the $\mathcal{S}_{\text{Cplx}}$ and $\mathcal{S}^{\{E\}}$ stability regions using Mathematica. At fourth order, we performed the same optimization for $\mathbf{A}^{\{E\}}$, however, we were unable to achieve satisfactory stability when using an existing type 2 DIMSIM for the implicit base method. Instead, we derived a new A(83°)-stable DIMSIM for which the ADI-GLM stability was acceptable.

Once $\mathbf{A}^{\{E\}}$ and $\mathbf{A}^{\{I\}}$ and v are determined, $\mathbf{B}^{\{E\}}$ and $\mathbf{B}^{\{I\}}$ are given using DIMSIM formulas [21, 61]. $\mathbf{W}^{\{I\}}$ and $\mathbf{W}^{\{E\}}$ are computed by solving eq. (4.11) and used in the starting procedure to generate initial values of the external stages at the beginning of the time-stepping loop in eq. (4.6). The starting procedure consists of integrating the system eq. (4.1) exactly over a short time-span $[0, (p-1)H]$ and using function values

$$f_k^{\{\sigma\}} := f^{\{\sigma\}}(y(kH)), \quad k = \{0, \ldots, p-1\}, \quad \sigma \in \{1, \ldots, N\}, \tag{4.32}$$

to approximate, via finite differences, the higher order derivatives needed in eq. (4.7). Readers interested in further details about the starting procedure may consult [23, 132]. The ending procedure for GLMs produces the high order approximation to $y(t_f)$ at the final time using stage values. All ADI-GLMs designed have the property that $\mathbf{c}_s = 1$, therefore, the last computed internal stage may be used as the final value in the integration with no further calculation required:

$$y_{t_f} = Y_s^{\{N\}} = h \sum_{\sigma=1}^{N} \sum_{j=1}^{s} a_{s,j}^{\{N,\sigma\}} f^{\{\sigma\}}(Y_j^{\{\sigma\}}) + \sum_{j=1}^{r} u_{s,j}^{\{N,\sigma\}} \xi_j^{\{\sigma\}[n-1]}. \tag{4.33}$$

Remark 4.9 (The ADI character of the methods)**.** The Butcher tableau for ADI-GLMs can be permuted to reflect the order of computation of stages in practice. In general, an ADI-GLM proceeds with computing internal stages:

$$\left\{ Y_1^{\{1\}}, Y_1^{\{2\}}, \ldots, Y_1^{\{N\}}, \ldots, Y_s^{\{1\}}, \ldots, Y_s^{\{2\}}, \ldots, Y_s^{\{N\}} \right\}, \tag{4.34}$$

after which external stage updates are computed. Let us consider the application of the second order ADI-GLM eq. (4.35a) to eq. (4.14). We reorder the tableau according to the

permutation list $\mathcal{P} = \{1, 3, 2, 4\}$ to get the permuted tableau eq. (4.35b).

$$\begin{array}{c|c|c} \mathbf{c} & \mathbf{A} & \mathbf{U} \\ \hline & \mathbf{B} & \mathbf{V} \end{array} = \left[\begin{array}{c|cccc|cccc} 0 & \frac{5}{8} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \frac{1}{4} & \frac{5}{8} & \frac{1}{2} & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{5}{8} & 0 & \frac{5}{8} & 0 & 0 & 0 & 1 & 0 \\ 1 & \frac{1}{4} & \frac{5}{8} & \frac{1}{4} & \frac{5}{8} & 0 & 0 & 0 & 1 \\ \hline & \frac{1}{2} & -\frac{5}{32} & -\frac{3}{128} & \frac{5}{128} & -\frac{5}{16} & \frac{21}{16} & 0 & 0 \\ & 0 & \frac{27}{32} & \frac{13}{128} & \frac{85}{128} & -\frac{5}{16} & \frac{21}{16} & 0 & 0 \\ & -\frac{3}{128} & \frac{5}{128} & -\frac{3}{128} & \frac{5}{128} & 0 & 0 & -\frac{5}{16} & \frac{21}{16} \\ & \frac{13}{128} & \frac{85}{128} & \frac{13}{128} & \frac{85}{128} & 0 & 0 & -\frac{5}{16} & \frac{21}{16} \end{array}\right], \tag{4.35a}$$

$$\begin{array}{c|c|c} \mathbf{c} & \mathbf{A}_{\mathcal{P},\mathcal{P}} & \mathbf{U}_{\mathcal{P},:} \\ \hline & \mathbf{B}_{:,\mathcal{P}} & \mathbf{V} \end{array} = \left[\begin{array}{c|cccc|cccc} 0 & \frac{5}{8} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{5}{8} & \frac{5}{8} & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & \frac{1}{4} & \frac{1}{2} & \frac{5}{8} & 0 & 0 & 1 & 0 & 0 \\ 1 & \frac{1}{4} & \frac{1}{4} & \frac{5}{8} & \frac{5}{8} & 0 & 0 & 0 & 1 \\ \hline & \frac{1}{2} & -\frac{3}{128} & -\frac{5}{32} & \frac{5}{128} & -\frac{5}{16} & \frac{21}{16} & 0 & 0 \\ & 0 & \frac{13}{128} & \frac{27}{32} & \frac{85}{128} & -\frac{5}{16} & \frac{21}{16} & 0 & 0 \\ & -\frac{3}{128} & -\frac{3}{128} & \frac{5}{128} & \frac{5}{128} & 0 & 0 & -\frac{5}{16} & \frac{21}{16} \\ & \frac{13}{128} & \frac{13}{128} & \frac{85}{128} & \frac{85}{128} & 0 & 0 & -\frac{5}{16} & \frac{21}{16} \end{array}\right]. \tag{4.35b}$$

We observe how the lower triangular structure of $\mathbf{A}_{\mathcal{P},\mathcal{P}}$ defines successive implicit stages in different directions while using previously computed stage values explicitly.

4.7 Numerical Experiments

In this section, we investigate numerically the accuracy and stability of ADI-DIMSIMs using 2D and 3D time-dependent parabolic PDEs. Up to this point, we have only considered autonomous problems, however, ADI-GLM extends to non-autonomous systems by evaluating the right hand side functions at the consistent times $t_{n-1} + \mathbf{c}h$. For a 3D problem we use

the equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + g(x,y,z,t), \tag{4.36a}$$

$$\begin{aligned} g(x,y,z,t) = {} & e^t(1-x)x(1-y)y(1-z)z \\ & + 2e^t(1-x)x(1-y)y + 2e^t(1-x)x(1-z)z \\ & + 2e^t(1-y)y(1-z)z - 6e^t \\ & + e^t\left(\left(x+\frac{1}{3}\right)^2 + \left(y+\frac{1}{4}\right)^2 + \left(z+\frac{1}{2}\right)^2\right), \end{aligned} \tag{4.36b}$$

with Dirichlet boundary conditions according to the exact solution:

$$u(x,y,z,t) = e^t(1-x)x(1-y)y(1-z)z \tag{4.37}$$

$$+ e^t\left(\left(x+\frac{1}{3}\right)^2 + \left(y+\frac{1}{4}\right)^2 + \left(z+\frac{1}{2}\right)^2\right). \tag{4.38}$$

The spatial discretization uses second order finite differences on the unit cube domain $D := \{x, y, z \in [0,1]\}$ with a uniform mesh with N_p points in each direction. We use the parameter N_p in the experiments to change the stiffness of directional derivatives. Note that using a uniform mesh allows us to factorize a tridiagonal 1D Jacobian matrix once and use it to efficiently to compute directional stages.

To verify the temporal order of convergence for the new methods, we integrate the problem over a time-span $t = [0,1]$ and record the relative ℓ_2 error at final time versus number of time steps. Figures 4.1a to 4.1c, verify the theoretical order for a range of mesh sizes. We compare ADI-DIMSIMs with an ADI scheme based on a fourth order IMEX Runge–Kutta method reported in [97, Example 3]. We note the deterioration in the order as the problem becomes more stiff with decreasing mesh size in fig. 4.1d.

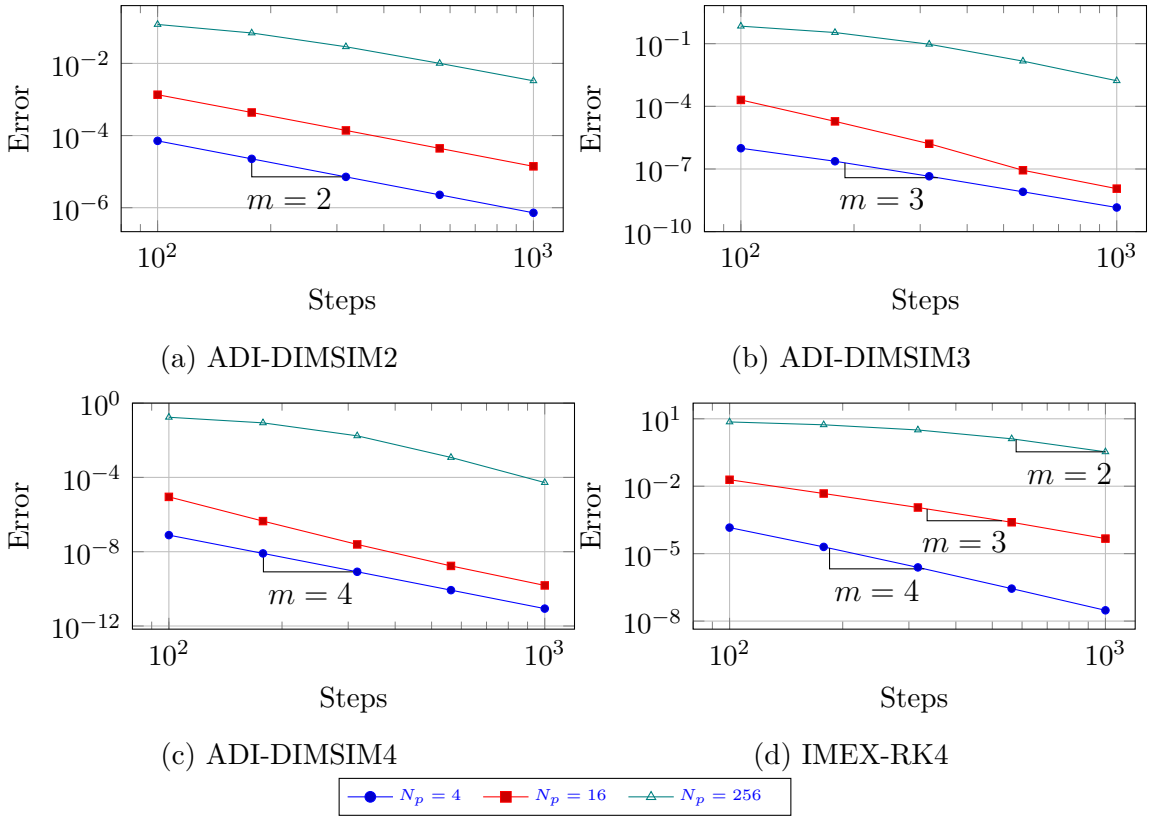
For a 2D numerical experiment the following problem is used on unit square domain $D = \{x, y \in [0,1]\}$, with the same spatial discretization and integrated over the same time-span:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + h(x,y,t), \tag{4.39a}$$

$$\begin{aligned} h(x,y,t) = {} & e^t(1-x)x(1-y)y + e^t\left(\left(x+\frac{1}{3}\right)^2 + \left(y+\frac{1}{4}\right)^2 - 4\right) \\ & + 2e^t(1-x)x + 2e^t(1-y)y, \end{aligned} \tag{4.39b}$$

with Dirichlet boundary conditions according to the exact solution:

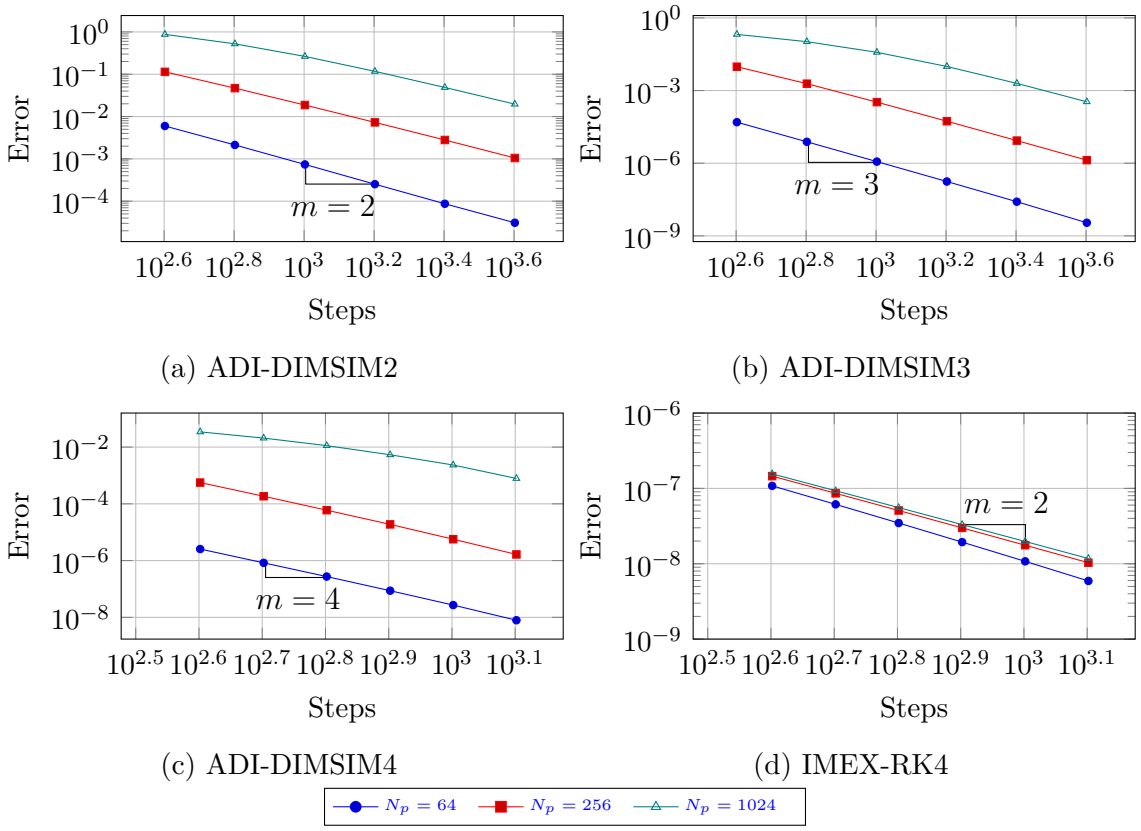
$$u(x,y,t) = e^t(1-x)x(1-y)y + e^t\left(\left(x+\frac{1}{3}\right)^2 + \left(y+\frac{1}{4}\right)^2\right). \tag{4.40}$$



(a) ADI-DIMSIM2 (b) ADI-DIMSIM3

(c) ADI-DIMSIM4 (d) IMEX-RK4

Figure 4.1: Convergence plots for ADI-DIMSIMs on 3D test problem compared to IMEX-RK4 method



(a) ADI-DIMSIM2

(b) ADI-DIMSIM3

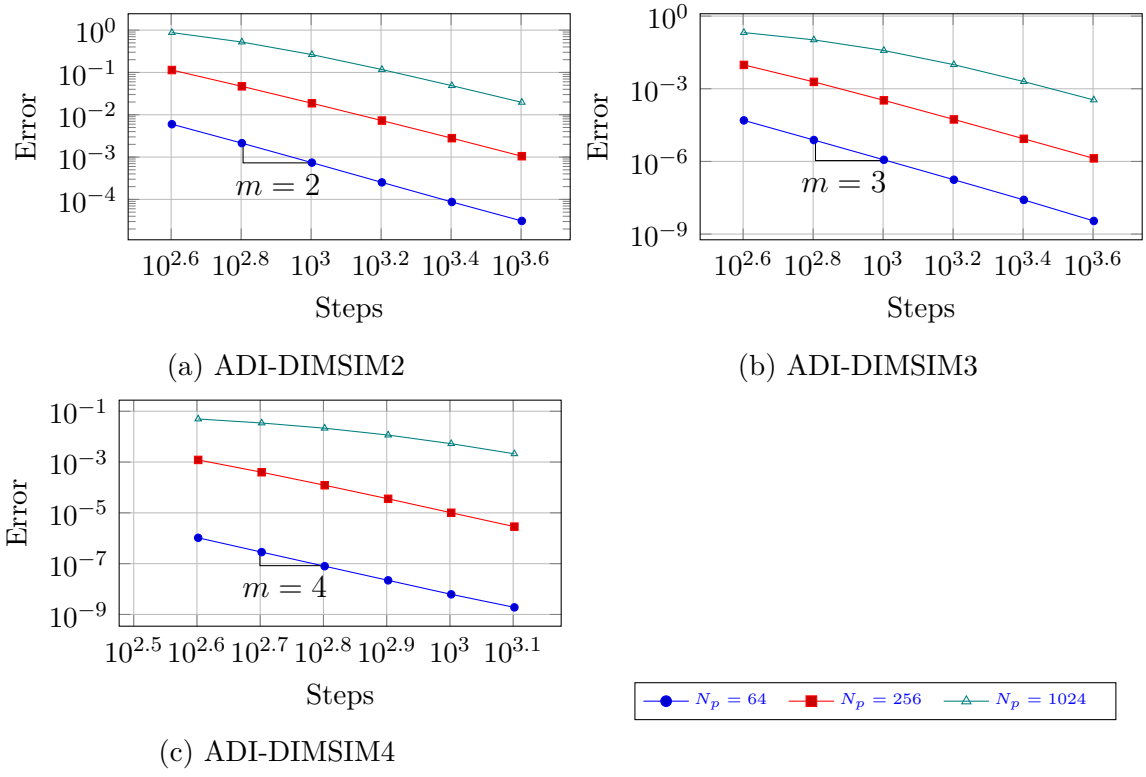
(c) ADI-DIMSIM4

(d) IMEX-RK4

FIGURE 4.2: Convergence plots for ADI-DIMSIMs on 2D test problem compared to IMEX-RK4 method

Figure 4.2 shows convergence plots for this experiment. Once again, we observe the order reduction for the IMEX-RK4 method in fig. 4.2d while ADI-DIMSIMs retain their convergence order in figs. 4.2a to 4.2c.

For a third set of experiments, we examine solutions of eq. (4.39), this time considering the forcing term $g(x, y, t)$ as a third partition to be treated explicitly in the entire integration. This means that the Butcher tableau in eq. (4.13) is used for these experiments. Figure 4.3 summarizes the results with close to theoretical order of ADI-DIMSIMs.



(a) ADI-DIMSIM2

(b) ADI-DIMSIM3

(c) ADI-DIMSIM4

Figure 4.3: Convergence plots for ADI-DIMSIMs on 2D test problem with an explicit partition

Chapter 5

Linearly-implicit General Linear Methods

5.1 Introduction

We are concerned with solving stiff initial value problems of the form

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

Implicit methods used to solve eq. (5.1), such as backward differentiation formula (BDF) and diagonally implicit Runge–Kutta (DIRK) methods, require solving nonlinear system of equations to compute implicitly-defined stage values. The nonlinear solves can be very expensive and often dominate the cost of a step. Linearly implicit methods, on the other hand, reduce the computational cost by posing stages as the solution to a *linear* systems [75, 76, 91]. A limitation of this process is that the accuracy of the method hinges on the availability of the exact Jacobian of the system. In response to this limitation, W-methods have been developed with additional order conditions such that terms involving the Jacobian are eliminated from the error expansion up to a certain order. As a result, these methods provide a flexible choice for the Jacobian of the system: the closer the approximate Jacobian is to the exact one, the better the stability of the method, while the order of convergence is unchanged. Rosenbrock-W (ROW) methods [117] are examples of this class of time stepping methods in the Runge–Kutta framework. [87, 109].

If the approximate Jacobian is considered to span a Krylov subspace of the exact Jacobian, ROW order conditions can be simplified to derive new methods with minimally implicit linear system solves. The resulting family of Rosenbrock–Krylov methods are introduced in [122] and further explored in [40, 124] for improved subspace selection. Another recent development is partitioned linearly implicit Runge–Kutta methods allowing Rosenbrock-type base method for multi-method integrations [99].

There are a number of incentives to extend ROW methods beyond the Runge–Kutta family. ROW methods can encounter order reduction when stages are solved using iterative solvers [123]. Furthermore, linearly-implicit methods, in general, are vulnerable to order reduction when used for solving PDEs with time-dependent boundary conditions as studied in [78]. Various improvements for ROW methods applied to parabolic problems have been proposed

in [1, 4, 71] resulting in methods stratified for stiff problems.

A more systematic approach is to avoid low stage orders. Rosenbrock methods with an explicit first stage can raise the stage order to two [93]. Glandon *et* al. [41] propose linearly implicit multi-step methods and the order conditions to derive methods both for exact and approximate Jacobians. As A-stability becomes unattainable for high order linear multi-step methods, one can also turn to multi-stage multi-step schemes to build high order, high stage order linearly implicit methods. A family of linearly-implicit two-step Runge-Kutta methods is reported in [104, 105, 128] with features such as step-size adaptation using variable coefficients, parallel stage computations, and super-convergence of the output variable. As discussed in [80] these methods when considered in their Nordsieck form are closely related to generic GLMs.

We are inspired by these contributions to further study new possibilities for linearly implicit W-methods stemming from GLMs. Our goal is to provide a new framework that facilitates creation of linearly-implicit methods for a variety of applications by providing the order condition theory and designing some examples of new families of methods based on existing GLMs. We will do away with the stage order restriction of Runge-Kutta methods and discuss its benefits on the asymptotic errors when our methods are applied to stiff problems.

Our approach bears similarities to IMEX GLMs. Partitioned GLMs, including IMEX, have been proposed in [24, 25, 132, 134, 137] and more recently in [89]. We have used some important convergence results for global errors of GLMs from the monograph [94] in our stiff analysis.

In the following we will study a number of topics: We start by reviewing GLMs in section 5.2. In section 5.3 we introduce the formulation for linearly implicit general linear methods. Section 5.4 gives the order conditions for these new methods. Sections 5.5 and 5.6 are dedicated to linear and stiff stability analysis for linearly implicit GLMs. In section 5.7 we show a number of new and existing families that fit in our framework including methods based on type 2 and type 4 DIMSIMS, BDF-W, and parallel methods. Section 5.8 presents the numerical experiments.

5.2 Traditional General Linear Methods

A GLM with s internal and r external stages advances the numerical solution of eq. (5.1) over the time interval $[t_{n-1}, t_n]$ with $t_n = t_{n-1} + h$ as follows:

$$Y_i = h \sum_{j=1}^{s} a_{i,j}\, f(Y_j) + \sum_{j=1}^{r} u_{i,j}\, \mathbb{y}_j^{[n-1]}, \qquad i = 1, \ldots, s, \tag{5.2a}$$

$$\mathbb{y}_i^{[n]} = h \sum_{j=1}^{s} b_{i,j}\, f(Y_j) + \sum_{j=1}^{r} v_{i,j}\, \mathbb{y}_j^{[n-1]}, \qquad i = 1, \ldots, r. \tag{5.2b}$$

Provided that the external stages from the previous step have the Taylor series expansion

$$\mathrm{y}_i^{[n-1]} = \sum_{k=0}^{p} w_{i,k}\, h^k\, y^{(k)}(t_{n-1}) + \mathcal{O}\big(h^{p+1}\big), \tag{5.3}$$

a method is said to have stage order q if

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

and order p if

$$\mathrm{y}_i^{[n]} = \sum_{k=0}^{p} w_{i,k}\, h^k\, y^{(k)}(t_n) + \mathcal{O}\big(h^{p+1}\big).$$

For brevity, the coefficients are represented in matrix form

$$\mathbf{A} \coloneqq [a_{i,j}] \in \mathbb{R}^{s\times s}, \quad \mathbf{U} \coloneqq [u_{i,j}] \in \mathbb{R}^{s\times r}, \quad \mathbf{B} \coloneqq [b_{i,j}] \in \mathbb{R}^{r\times s}, \tag{5.4}$$

$$\mathbf{V} \coloneqq [v_{i,j}] \in \mathbb{R}^{r\times r}, \quad \mathbf{W} \coloneqq [w_{i,j}] = [\mathbf{w}_0 \ \cdots \ \mathbf{w}_p] \in \mathbb{R}^{r\times(p+1)}, \tag{5.5}$$

and the GLM eq. (5.2) can be represented by the following Butcher tableau:

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

Similar to Runge–Kutta schemes, two broad categories of GLMs are of interest. Explicit GLMs computes internal stages using information from incoming external stages and previously computed internal stages. The computational cost is therefore limited to evaluating right hand side functions. Implicit methods on the other hand may compute coupled stages that require nonlinear system solves. methods with lower triangular $\mathbf{A}$ compute stages that are only implicit in the stage being computed. These methods also benefit from good linear stability properties. If $\mathbf{A}$ is strictly lower triangular the GLM computes internal stages using only previous stages and is therefore explicit. The GLM framework is extensive and well-established. Other structures of coefficient matrices are possible and lead to varying levels of implicit and explicitness. We refer readers interested in theoretical foundation of these methods to the literature [18, 22, 59]. In the following we reiterate an important theorem for the order conditions of GLMs.

Definition 5.1 (GLM preconsistency)**.** The GLM eq. (5.2) is said to be preconsistent if the following conditions hold:

$$\mathbf{U}\,\mathbf{w}_0 = \mathbf{1}_s, \quad \mathbf{V}\,\mathbf{w}_0 = \mathbf{w}_0. \tag{5.6}$$

Theorem 5.2 (GLM order conditions)**.** *Consider a preconsistent GLM with external stages satisfying eq. (5.3). All of the following are equivalent:*

1. *The GLM eq. (5.2) has order p and stage order $q \in \{p-1, p\}$ for all sufficiently smooth f.*

2. *The method coefficients satisfy*

$$\begin{aligned} \frac{\mathbf{c}^{\mathsf{X}k}}{k!} - \frac{\mathbf{A}\,\mathbf{c}^{\mathsf{X}(k-1)}}{(k-1)!} - \mathbf{U}\,\mathbf{w}_k &= \mathbf{0}_s, \qquad k = 1, \dots, q, \\ \sum_{\ell=0}^{k} \frac{\mathbf{w}_{k-\ell}}{\ell!} - \frac{\mathbf{B}\,\mathbf{c}^{\mathsf{X}(k-1)}}{(k-1)!} - \mathbf{V}\,\mathbf{w}_k &= \mathbf{0}_s, \qquad k = 1, \dots, p, \end{aligned}$$

   where we have used the notation $\mathbf{c}^{\mathsf{X}k}$ to denote component-wise k-th power of the vector $\mathbf{c}$.

3. *The GLM eq. (5.2) has order p and stage order $q \in \{p-1, p\}$ for all scalar, linear problems*

$$y' = \lambda\, y, \qquad y(0) = 1, \tag{5.7}$$

   where $\lambda \in \mathbb{R}$.

Proof. The equivalence of statements one and two is discussed in [59]. Clearly statement one implies statement three as it uses the special case $f(y) = \lambda\, y$. To complete the proof, we prove the converse.

First, assume a GLM has order p and stage order $q \in \{p-1, p\}$ for eq. (5.7). Therefore, the internal stages satisfy

$$\begin{aligned} Y &= h\,\mathbf{A}\,f(Y) + \mathbf{U}\,\mathbb{y}^{[n-1]}, \\ e^{\mathbf{c}\,z} &= z\,\mathbf{A}\,e^{\mathbf{c}\,z} + \mathbf{U}\sum_{k=0}^{p} \mathbf{w}_k\,h^k\,y^{(k)}(0) + \mathcal{O}\big(h^{q+1}\big), \\ e^{\mathbf{c}\,z} &= z\,\mathbf{A}\,e^{\mathbf{c}\,z} + \mathbf{U}\,\mathbf{w}(z) + \mathcal{O}\big(h^{p+1}\big), \end{aligned} \tag{5.8}$$

where $z = h\,\lambda$, $\mathbf{w}(z) = \sum_{k=0}^{p} \mathbf{w}_k\,z^k$, and the exponential of vectors are performed component-wise. Similarly, the external stages satisfy

$$\begin{aligned} \mathbb{y}^{[n]} &= h\,\mathbf{B}\,f(Y) + \mathbf{V}\,\mathbb{y}^{[n-1]}, \\ e^{\mathbf{c}\,z}\,\mathbf{w}(z) &= z\,\mathbf{B}\,e^{\mathbf{c}\,z} + \mathbf{V}\,\mathbf{w}(z) + \mathcal{O}\big(h^{p+1}\big). \end{aligned} \tag{5.9}$$

Together eqs. (5.8) and (5.9) are exactly the GLM order conditions given in [59, Theorems 2.4.1 and 2.4.2]. Thus, statement one is proved, and the proof is complete. □

5.3 Formulation of linearly implicit GLMs

In this section we introduce the new class of linearly-implicit general linear methods as the extension of Rosenbrock and linearly-implicit Runge-Kutta method to multi-step and multi-stage integration.

Definition 5.3. linearly implicit GLMs

One step of a linearly implicit GLM applied to eq. (5.1) with a (possibly) different approximate Jacobian matrix $\mathbf{L}_i \approx \frac{\partial f}{\partial y}$ at time $t_i = t_{n-1} + c_i h$ for $i = 1, \dots, s$ reads as

$$\begin{aligned} K_i = h\, f\left(\sum_{j=1}^{i-1} a_{i,j}\, K_j + \sum_{j=1}^{r} u_{i,j}\, \mathrm{y}_j^{[n-1]}\right) \\ + h\, \mathbf{L}_i \sum_{j=1}^{i} \gamma_{i,j}\, K_j + h\, \mathbf{L}_i \sum_{j=1}^{r} \psi_{i,j}\, \mathrm{y}_j^{[n-1]}, \end{aligned} \qquad i = 1, \dots, s, \tag{5.10a}$$

$$\mathrm{y}_i^{[n]} = \sum_{j=1}^{s} b_{i,j}\, K_j + \sum_{j=1}^{r} v_{i,j}\, \mathrm{y}_j^{[n-1]}, \qquad i = 1, \dots, r. \tag{5.10b}$$

The formulation of linearly implicit GLMs for non-autonomous systems will be discussed in section 5.4.3 after order conditions are introduced.

Equation (5.10) can be represented more compactly in matrix form as

$$K = h\, F\big(\mathbf{A} \circledast K + \mathbf{U} \circledast \mathrm{y}^{[n-1]}\big) + h\, \mathbf{L}\, \mathbf{\Gamma} \circledast K + h\, \mathbf{L}\, \mathbf{\Psi} \circledast \mathrm{y}^{[n-1]}, \tag{5.11a}$$

$$\mathrm{y}^{[n]} = \mathbf{B} \circledast K + \mathbf{V} \circledast \mathrm{y}^{[n-1]}, \tag{5.11b}$$

where $\mathbf{L} = \mathrm{blkdiag}\,(\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_s)$ and we have used the $\circledast$ notation to indicate Kronecker product with the identity matrix such that $\mathbf{A} \circledast K := (\mathbf{A} \otimes \mathbf{I}_{d\times d})\, K$, and

$$K = \begin{bmatrix} K_1 \\ \vdots \\ K_s \end{bmatrix}, \quad \mathrm{y}^{[n]} = \begin{bmatrix} \mathrm{y}_1^{[n]} \\ \vdots \\ \mathrm{y}_s^{[n]} \end{bmatrix}, \quad F(Y) = \begin{bmatrix} f(Y_1) \\ \vdots \\ f(Y_s) \end{bmatrix}. \tag{5.12}$$

We represent a linearly implicit GLM using the Butcher tableau

$$\begin{array}{c|cc|cc} \mathbf{c} & \mathbf{A} & \mathbf{\Gamma} & \mathbf{U} & \mathbf{\Psi} \\ \hline & \mathbf{B} & & \mathbf{V} & \end{array}, \tag{5.13}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{s\times s}$ and $\mathbf{\Psi} \in \mathbb{R}^{s\times r}$.

5.3.1 Efficient implementation

A more efficient formulation of linearly implicit GLMs is possible, similar to that described in [51, Sec IV.7] for Rosenbrock methods such that it incurs fewer Jacobian-vector product computations per internal stage. We start by considering the slightly more general form of eq. (5.1):

$$\mathbf{M}\, y' = f(y), \tag{5.14}$$

where $\mathbf{M}$ is a mass matrix. The special case of a singular $\mathbf{M}$ is considered in section 5.6.

Assuming $\mathbf{\Gamma}$ is lower triangular and invertible, we can formulate the method in the new variable Z_i, where

$$K_i = \sum_{j=1}^{i} \left(\mathbf{\Gamma}^{-1}\right)_{i,j} Z_j - \sum_{j=1}^{r} \left(\mathbf{\Gamma}^{-1}\, \mathbf{\Psi}\right)_{i,j} \mathit{y}_j^{[n-1]}. \tag{5.15}$$

After some algebraic manipulations, eq. (5.10) becomes

$$\begin{aligned}
\left(\frac{1}{\gamma_{i,i}}\mathbf{M} - h\,\mathbf{L}_i\right) Z_i = h\, f\left(\sum_{j=1}^{i-1} \left(\mathbf{A}\,\mathbf{\Gamma}^{-1}\right)_{i,j} Z_j + \sum_{j=1}^{r} \left(\mathbf{U} - \mathbf{A}\,\mathbf{\Gamma}^{-1}\,\mathbf{\Psi}\right)_{i,j} \mathit{y}_j^{[n-1]}\right) \\
+ \mathbf{M}\left(-\sum_{j=1}^{i-1} \left(\mathbf{\Gamma}^{-1}\right)_{i,j} Z_j + \sum_{j=1}^{r} \left(\mathbf{\Gamma}^{-1}\,\mathbf{\Psi}\right)_{i,j} \mathit{y}_j^{[n-1]}\right), \\
\mathit{y}_i^{[n]} = \sum_{j=1}^{s} \left(\mathbf{B}\,\mathbf{\Gamma}^{-1}\right)_{i,j} Z_j + \sum_{j=1}^{r} \left(\mathbf{V} - \mathbf{B}\,\mathbf{\Gamma}^{-1}\,\mathbf{\Psi}\right)_{i,j} \mathit{y}_j^{[n-1]}.
\end{aligned}$$

Note that the approximate Jacobians only appear on the left-hand side (in the linear system matrix), and this formulation avoids the additional matrix vector products present in the right-hand side of eq. (5.10a).

5.4 Order conditions for linearly implicit GLMs

In this section we derive the necessary and sufficient order conditions for desired internal and external stage orders and show how the conditions can ensure accuracy of the method with arbitrary Jacobians. We also explore how these conditions are related to IMEX GLM pairs.

5.4.1 Preconsistency conditions

We apply the method in eq. (5.11) to the test problem

$$y' = 0, \qquad y(t_0) = \kappa, \tag{5.16}$$

where $\kappa \in \mathbb{R}$ and note that the constant solution is $y(t) = \kappa$. We would like to have all approximations of $y(t)$ and $y'(t)$ to be $\mathcal{O}(h)$ accurate. We therefore require

$$\begin{aligned} \mathbf{A}\, K + \mathbf{U}\, \mathbb{y}^{[n-1]} &= \mathbf{1}_s\, \kappa + \mathcal{O}(h), \\ \mathbf{B}\, K + \mathbf{V}\, \mathbb{y}^{[n-1]} &= \mathbf{w}_0\, \kappa + \mathcal{O}(h), \\ h\, \mathbf{L}_i\, \mathbf{\Psi}\, \mathbb{y}^{[n-1]} &= \mathbf{0}_s + \mathcal{O}\big(h^2\big), \end{aligned}$$

which leads to the following preconsistency conditions:

$$\mathbf{U}\, \mathbf{w}_0 = \mathbf{1}_s, \tag{5.17a}$$
$$\mathbf{\Psi}\, \mathbf{w}_0 = \mathbf{0}_s, \tag{5.17b}$$
$$\mathbf{V}\, \mathbf{w}_0 = \mathbf{w}_0. \tag{5.17c}$$

5.4.2 Order conditions

Definition 5.4 (Linearly implicit GLM internal stage order)**.** A linearly implicit GLM with incoming external stages satisfying eq. (5.3) has order q if the following approximation holds:

$$K_i = h\, y'(t_{n-1} + c_i\, h) + \mathcal{O}\big(h^{q+2}\big). \tag{5.18}$$

Definition 5.5 (Linearly implicit GLM external stage order)**.** A linearly implicit GLM with incoming external stages satisfying eq. (5.3) has external stage order p if

$$\mathbb{y}_i^{[n]} = \sum_{k=0}^{p} w_{i,j}\, h^k\, y^{(k)}(t_n) + \mathcal{O}\big(h^{p+1}\big). \tag{5.19}$$

Definition 5.6 (Method order)**.** We say that a linearly implicit GLM is of order (q, p) if it has internal stage order q and external stage order p.

Theorem 5.7 (Order conditions)**.** *A linearly implicit GLM that satisfies the preconsistency conditions eq. (5.17) has order (q, p) with $q \in \{p, p-1\}$ for all $\mathbf{L}_i$ and sufficiently smooth f if and only if its coefficients satisfy the following order conditions:*

$$\frac{\mathbf{c}^{\fallingdotp k}}{k!} - \frac{\mathbf{A}\, \mathbf{c}^{\fallingdotp (k-1)}}{(k-1)!} - \mathbf{U}\, \mathbf{w}_k = \mathbf{0}_s, \qquad k = 1, \dots, q, \tag{5.20a}$$
$$\sum_{\ell=0}^{k} \frac{\mathbf{w}_{k-\ell}}{\ell!} - \frac{\mathbf{B}\, \mathbf{c}^{\fallingdotp (k-1)}}{(k-1)!} - \mathbf{V}\, \mathbf{w}_k = \mathbf{0}_r, \qquad k = 1, \dots, p, \tag{5.20b}$$
$$\frac{\mathbf{\Gamma}\, \mathbf{c}^{\fallingdotp k-1}}{(k-1)!} + \mathbf{\Psi}\, \mathbf{w}_k = \mathbf{0}_s, \qquad k = 1, \dots, q. \tag{5.20c}$$

Remark 5.8. The conditions in proposition 5.7 allow for different Jacobian approximations in different stages. This is particularly useful in the case of parallel methods as well as when stages are solved using iterative solvers.

Remark 5.9 (Implicit and explicit components)**.** Note that eqs. (5.20a) and (5.20b) are also the order (q,p) conditions for the explicit traditional GLM with coefficients $(\mathbf{A},\mathbf{U},\mathbf{B},\mathbf{V})$. Next, we introduce the following notation:

$$\widehat{\mathbf{A}} \coloneqq \mathbf{\Gamma} + \mathbf{A} \quad \text{and} \quad \widehat{\mathbf{U}} \coloneqq \mathbf{\Psi} + \mathbf{U}.$$

An equivalent formulation of eq. (5.20c) can be given as follows: Subtracting eq. (5.20c) from eq. (5.20a) leads to

$$\frac{\mathbf{c}^{\curlywedge k}}{k!} - \frac{\widehat{\mathbf{A}}\,\mathbf{c}^{\curlywedge (k-1)}}{(k-1)!} - \widehat{\mathbf{U}}\,\mathbf{w}_k = \mathbf{0}_s, \qquad k = 1,\dots,q, \tag{5.21}$$

and vice-versa, subtracting eq. (5.20a) from eq. (5.21) restores eq. (5.20c). Therefore the order conditions (5.20) can be written in an equivalent from by replacing eq. (5.20c) with eq. (5.21).

We note that eq. (5.21) and eq. (5.20b) define order (q,p) conditions for a traditional implicit GLM with coefficients $(\widehat{\mathbf{A}},\widehat{\mathbf{U}},\mathbf{B},\mathbf{V})$, having the same weights $\mathbf{W}$ as the explicit method.

Remark 5.10 (Abscissa vector)**.** From eq. (5.20a) it follows that, for methods of stage order $q \geq 1$:

$$\mathbf{c} = \mathbf{A}\,\mathbf{1}_s + \mathbf{U}\,\mathbf{w}_1. \tag{5.22}$$

Remark 5.11 (IMEX pair)**.** From eq. (5.21) the abscissa of the implicit method is

$$\widehat{\mathbf{c}} = \widehat{\mathbf{A}}\,\mathbf{1}_s + \widehat{\mathbf{U}}\,\mathbf{w}_1 = \mathbf{c} + \mathbf{\Gamma}\,\mathbf{1}_s + \mathbf{\Psi}\,\mathbf{w}_1 = \mathbf{c}.$$

As a consequence of (5.20c), the explicit method $(\mathbf{A},\mathbf{U},\mathbf{B},\mathbf{V})$ and the implicit method $(\widehat{\mathbf{A}},\widehat{\mathbf{U}},\mathbf{B},\mathbf{V})$ also share the same abscissa vector. Therefore, from any linearly implicit GLM one can obtain an IMEX-GLM of order (q,p) [132]. We note that the reverse is not true since in general IMEX-GLM pairs do not share $\mathbf{W}$ and $\mathbf{B}$ coefficients.

Proof. First assume a linearly implicit GLM has order (q,p) with $q \in \{p, p-1\}$ for all $\mathbf{L}_i$ and sufficiently smooth f. It must have this order for the particular case $\mathbf{L}_i = \mathbf{0}_{d\times d}$ in which the method degenerates to a traditional, explicit GLM with coefficients $(\mathbf{A},\mathbf{B},\mathbf{U},\mathbf{V})$. The traditional GLM order conditions of proposition 5.2 imply eqs. (5.20a) and (5.20b).

Consider also the particular case $f(y) = \mathbf{L}\,y$ and $\mathbf{L}_i = \mathbf{L}$ as in eq. (5.7). For linear problems, a linearly implicit GLM degenerates to the implicit GLM with coefficients $(\widehat{\mathbf{A}},\widehat{\mathbf{U}},\mathbf{B},\mathbf{V})$. By proposition 5.2, a GLM of order (q,p) for linear problems must have coefficients that satisfy eq. (5.20b) and eq. (5.21). Based on the discussion in proposition 5.9, this implies the final order condition given in eq. (5.20c).

Now assume a linearly implicit GLM has coefficients satisfying eq. (5.20). Substituting the exact solution into the right-hand side of eq. (5.11a) and using Taylor expansions of $y(t_{n-1})$

and $y(t_{n-1} + \mathbf{c}\,h)$ we get

$$\begin{aligned}
&h\,F\big(h\,\mathbf{A} \boxtimes y'(t_{n-1} + \mathbf{c}\,h) + \mathbf{U} \boxtimes \mathbb{y}^{[n-1]}\big) + h\,\mathbf{L}\,\mathbf{\Gamma} \boxtimes y'(t_{n-1} + \mathbf{c}\,h) + h\,\mathbf{L}\,\mathbf{\Psi} \boxtimes \mathbb{y}^{[n-1]} \\
&= h\,F\left((\mathbf{U}\,\mathbf{w}_0) \otimes y(t_{n-1}) + \sum_{k=1}^{p} h^k \left(\frac{\mathbf{A}\,\mathbf{c}^{\mathbb{X}(k-1)}}{(k-1)!} + \mathbf{U}\,\mathbf{w}_k\right) \otimes y^{(k)}(t_{n-1}) + \mathcal{O}\big(h^{p+1}\big)\right) \\
&\quad + h\,\mathbf{L}\left((\mathbf{\Psi}\,\mathbf{w}_0) \otimes y(t_{n-1}) + \sum_{k=1}^{p} h^k \left(\frac{\mathbf{\Gamma}\,\mathbf{c}^{\mathbb{X}(k-1)}}{(k-1)!} + \mathbf{\Psi}\,\mathbf{w}_k\right) \otimes y^{(k)}(t_{n-1}) + \mathcal{O}\big(h^{p+1}\big)\right) \\
&= h\,F\big(y(t_{n-1} + \mathbf{c}\,h) + \mathcal{O}\big(h^{q+1}\big)\big) \\
&\quad + h^{p+1}\left(\mathbf{L}\left(\frac{\mathbf{\Gamma}\,\mathbf{c}^{\mathbb{X}(p-1)}}{(p-1)!} + \mathbf{\Psi}\,\mathbf{w}_p\right)\right) \otimes y^{(p)}(t_{n-1}) + \mathcal{O}\big(h^{p+2}\big) \\
&= h\,y'(t_{n-1} + \mathbf{c}\,h) + \mathcal{O}\big(h^{q+2}\big).
\end{aligned} \tag{5.23}$$

Next, we subtract eq. (5.23) from eq. (5.11a) to approximate the error of the internal stages:

$$\begin{aligned}
K - h\,y'(t_{n-1} + \mathbf{c}\,h) = {}& h\,F\big(\mathbf{A} \boxtimes K + \mathbf{U} \boxtimes \mathbb{y}^{[n-1]}\big) \\
& - h\,F\big(h\mathbf{A} \boxtimes y'(t_{n-1} + \mathbf{c}\,h) + \mathbf{U} \boxtimes \mathbb{y}^{[n-1]}\big) \\
& + h\,\mathbf{L}\,(K - h\,y'(t_{n-1} + \mathbf{c}\,h)) + \mathcal{O}\big(h^{q+2}\big).
\end{aligned} \tag{5.24}$$

Applying the 2-norm $\|\cdot\| = \|\cdot\|_2$ yields

$$\begin{aligned}
\|K - h\,y'(t_{n-1} + \mathbf{c}\,h)\| \leq {}& h\,C\,\|\mathbf{A}\|\,\|K - h\,y'(t_{n-1} + \mathbf{c}\,h)\| \\
& + h\,\|\mathbf{L}\|\,\|K - h\,y'(t_{n-1} + \mathbf{c}\,h)\| + \mathcal{O}\big(h^{q+2}\big),
\end{aligned}$$

where C is the Lipschitz constant of F. Thus,

$$\|K - h\,y'(t_{n-1} + \mathbf{c}\,h)\| \leq (1 - h\,C\,\|\mathbf{A}\| - h\,\|\mathbf{L}\|)^{-1} \cdot \mathcal{O}\big(h^{q+2}\big),$$

and for sufficiently small h

$$\|K - h\,y'(t_{n-1} + \mathbf{c}\,h)\| = \mathcal{O}\big(h^{q+2}\big),$$

which proves the internal stage order of the method. A result from eq. (5.24) we will use later is that:

$$\begin{aligned}
&F\big(\mathbf{A} \boxtimes K + \mathbf{U} \boxtimes \mathbb{y}^{[n-1]}\big) - F(h\mathbf{A} \boxtimes y'(t_{n-1} + \mathbf{c}\,h) + (\mathbf{UW}) \boxtimes \eta(t_{n-1})) \\
&\quad = \mathcal{O}\big(h^{p+1}\big) + \mathcal{O}\big(h^{q+1}\big).
\end{aligned} \tag{5.25}$$

Next, we will show that the external stage order is p. Once again, using the order conditions

eq. (5.20) and the Taylor expansion of $y(t_{n-1} + \mathbf{c}\,h)$ we have:

$$\begin{aligned}
\mathbb{y}^{[n]} &= \mathbf{B} \circledast K + \mathbf{V} \circledast \mathbb{y}^{[n-1]} \\
&= h\mathbf{B} \circledast y'(t_{n-1} + \mathbf{c}\,h) + \mathbf{V} \circledast \mathbb{y}^{[n-1]} + \mathcal{O}\big(h^{q+2}\big) \\
&= (\mathbf{V}\,\mathbf{w}_0) \otimes y(t_{n-1}) + \sum_{k=1}^{p} h^k \left(\frac{\mathbf{B}\,\mathbf{c}^{\chi(k-1)}}{(k-1)!} + \mathbf{V}\,\mathbf{w}_k \right) \otimes y^{(k)}(t_{n-1}) \\
&\quad + \mathcal{O}\big(h^{p+1}\big) + \mathcal{O}\big(h^{q+2}\big) \\
&= \mathbf{w}_0 \otimes y(t_{n-1}) + \sum_{k=1}^{p} \left(\sum_{\ell=0}^{k} h^k \frac{\mathbf{w}_{k-\ell}}{\ell!} \right) \otimes y^{(k)}(t_{n-1}) + \mathcal{O}\big(h^{\min(p+1,q+2)}\big) \\
&= \sum_{k=0}^{p} h^k \mathbf{w}_k \otimes y^{(k)}(t_n) + \mathcal{O}\big(h^{p+1}\big).
\end{aligned}$$

Note that $\min(p+1, q+2) = p+1$. □

Corollary 5.12. *Suppose a linearly implicit GLM has stage order $q = p-1$, but the underlying implicit method $(\widehat{\mathbf{A}}, \widehat{\mathbf{U}}, \mathbf{B}, \mathbf{V})$ has stage order $q = p$. If and $\mathbf{L}_i = \frac{\partial f}{\partial y}\Big|_{t_{n-1}} + \mathcal{O}(h)$ for $i = 1, \dots s$, then $K_i = h\,y'(t_{n-1} + \mathbf{c}\,h) + \mathcal{O}(h^{p+2})$. That is, the stage order is one order higher than proposition 5.7 predicts.*

Proof. If the assumptions of proposition 5.12 hold and $\mathbf{L}_i = \frac{\partial f}{\partial y}\Big|_{t_{n-1}} + \mathcal{O}(h)$ the local truncation error in eq. (5.23) can be sharpened. We have that

$$\begin{aligned}
& h\,F\big(h\,\mathbf{A} \circledast y'(t_{n-1} + \mathbf{c}\,h) + \mathbf{U} \circledast \mathbb{y}^{[n-1]}\big) + h\,\mathbf{L}\,\mathbf{\Gamma} \circledast y'(t_{n-1} + \mathbf{c}\,h) + h\,\mathbf{L}\,\mathbf{\Psi} \circledast \mathbb{y}^{[n-1]} \\
&= h\,F\left(y(t_{n-1} + \mathbf{c}\,h) + h^p \left(\frac{\mathbf{A}\,\mathbf{c}^{\chi(p-1)}}{(p-1)!} + \mathbf{U}\,\mathbf{w}_p \right) \otimes y^{(p)}(t_{n-1}) + \mathcal{O}\big(h^{p+1}\big) \right) \\
&\quad + h^{p+1}\,\mathbf{L} \left(\frac{\mathbf{\Gamma}\,\mathbf{c}^{\chi(p-1)}}{(p-1)!} + \mathbf{\Psi}\,\mathbf{w}_p \right) \otimes y^{(p)}(t_{n-1}) + \mathcal{O}\big(h^{p+2}\big) \qquad (5.26) \\
&= h\,y'(t_{n-1} + \mathbf{c}\,h) + h^{p+1} \left. \frac{\partial F}{\partial y} \right|_{t_{n-1} + \mathbf{c}\,h} \left(\frac{\widehat{\mathbf{A}}\,\mathbf{c}^{\chi(p-1)}}{(p-1)!} + \widehat{\mathbf{U}}\,\mathbf{w}_p \right) \otimes y^{(p)}(t_{n-1}) \\
&\quad + \mathcal{O}\big(h^{p+2}\big),
\end{aligned}$$

where $\frac{\partial F}{\partial y} = \mathbf{I}_{s\times s} \otimes \frac{\partial f}{\partial y}$ Proceeding exactly as in the proof of proposition 5.7 gives the desired result of

$$\|K - h\,y'(t_{n-1} + \mathbf{c}\,h)\| = \mathcal{O}\big(h^{p+2}\big).$$

□

5.4.3 Non-autonomous formulation

So far we have considered the ode right-hand-side function eq. (5.1) to be autonomous. In this section we will show that the non-autonomous case does not affect the formulation of the method. Applying the method on the non-autonomous system

$$\begin{bmatrix} y \\ t \end{bmatrix}' = \begin{bmatrix} f(t,y) \\ 1 \end{bmatrix}, \tag{5.27}$$

and the approximate Jacobian eq. (5.27)

$$\begin{bmatrix} \mathbf{L}_i & \frac{\partial f}{\partial t}\big|_{t_{n-1}+c_i h} \\ 0 & 0 \end{bmatrix}, \tag{5.28}$$

we get

$$\begin{bmatrix} K_i \\ \kappa_i \end{bmatrix} = h \begin{bmatrix} f(\tau_i, Y_i) \\ 1 \end{bmatrix} + h \begin{bmatrix} \mathbf{L}_i & g_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{s} \gamma_{i,j}\, K_j \\ \sum_{i=1}^{s} \gamma_{i,j}\, \kappa_j \end{bmatrix} + h \begin{bmatrix} \mathbf{L}_i & g_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{r} \psi_{i,j}\, \mathbb{y}_j^{[n-1]} \\ \sum_{i=1}^{r} \psi_{i,j}\, \zeta_j^{[n-1]} \end{bmatrix},$$
$$\begin{bmatrix} \mathbb{y}_i^{[n]} \\ \zeta_i^{[n]} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{s} b_{i,j}\, K_j \\ \sum_{i=1}^{s} b_{i,j}\, \kappa_j \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^{r} v_{i,j}\, \mathbb{y}_j^{[n-1]} \\ \sum_{i=1}^{r} v_{i,j}\, \zeta_j^{[n-1]} \end{bmatrix}.$$

Assuming an exact starting procedure for the time variable, $\kappa_i = h$ and $\zeta_i^{[n]} = t_n\, w_{i,0} + h\, w_{i,1}$. From eq. (5.11a), the time argument for $f(\tau_i, Y_i)$ satisfies

$$\tau_i = \sum_{j=1}^{s} a_{i,j}\, \kappa_j + \sum_{j=1}^{r} u_{i,j}\, \zeta_j^{[n-1]} = t_{n-1} + c_i\, h, \tag{5.29}$$

by the preconsistency condition eq. (5.17a) and stage order one condition eq. (5.20a). Furthermore,

$$\begin{aligned} K_i &= h\, f(t_{n-1} + c_i\, h, Y_i) + h\, \mathbf{L}_i \sum_{j=1}^{i} \gamma_{i,j}\, K_j + h\, \mathbf{L}_i \sum_{j=1}^{r} \psi_{i,j}\, \mathbb{y}_j^{[n-1]} \\ &\quad + g_i \left(\sum_{j=1}^{i} \gamma_{i,j}\, \kappa_j + \psi_{i,j}\, \zeta_j^{[n-1]} \right) \\ &= h\, f(t_{n-1} + c_i\, h, Y_i) + h\, \mathbf{L}_i \sum_{j=1}^{i} \gamma_{i,j}\, K_j + h\, \mathbf{L}_i \sum_{j=1}^{r} \psi_{i,j}\, \mathbb{y}_j^{[n-1]} \end{aligned} \tag{5.30}$$

by the preconsistency condition eq. (5.17c) and stage order one condition eq. (5.20c). Note, the only difference between the autonomous form eq. (5.10) and the non-autonomous form eq. (5.30) is the time argument in the right hand side function f. Unlike Rosenbrock methods, the non-autonomous formulation of linearly implicit GLMs does not include the terms involving time-derivative of the right-hand-side.

5.5 Stability analysis for linearly implicit GLMs

We will now discuss the stability properties of linearly implicit GLMs when applied to linear, Prothero-Robinson, index-1 DAEs and singularly perturbed systems.

5.5.1 Linear stability

Applying eq. (5.11) to the scalar Dahlquist test problem

$$y' = \lambda\, y, \tag{5.31}$$

and using the Jacobian approximations $\mathbf{L}_i = \widehat{\lambda}$ reveals that the stability matrix of the linearly implicit GLM is

$$\begin{aligned} \mathrm{y}^{[n]} &= \mathbf{M}(z, \widehat{z})\, \mathrm{y}^{[n-1]}, \\ \mathbf{M}(z, \widehat{z}) &= \mathbf{V} + \mathbf{B}\left(\mathbf{I}_{s\times s} - z\,\mathbf{A} - \widehat{z}\,\mathbf{\Gamma}\right)^{-1}\left(z\,\mathbf{U} + \widehat{z}\,\mathbf{\Psi}\right), \end{aligned}$$

where $z = h\,\lambda$ and $\widehat{z} = h\,\widehat{\lambda}$.

In the case the exact Jacobian is used, $\widehat{z} = z$ and, as expected, the stability function of the underlying implicit method is recovered:

$$\widehat{\mathbf{M}}(z) = \mathbf{M}(z, z) = \mathbf{V} + z\,\mathbf{B}\left(\mathbf{I}_{s\times s} - z\,\widehat{\mathbf{A}}\right)^{-1}\widehat{\mathbf{U}}.$$

In the case where the arbitrary Jacobian is chosen as $\mathbf{L}_i = 0$ the stability matrix of linearly implicit GLM is the same as that of the explicit GLM:

$$\widetilde{\mathbf{M}}(z) = \mathbf{M}(z, 0) = \mathbf{V} + z\,\mathbf{B}\left(\mathbf{I}_{s\times s} - z\,\mathbf{A}\right)^{-1}\mathbf{U}.$$

We define regions of stability for the underlying implicit and explicit methods as $\mathbb{C}$

$$\widehat{S} = \{z \in \mathbb{C}^- : \sup_n \left\|\widehat{\mathbf{M}}(z)\right\|^n < \infty\}, \tag{5.32a}$$

$$\widetilde{S} = \{z \in \mathbb{C}^- : \sup_n \left\|\widetilde{\mathbf{M}}(z)\right\|^n < \infty\}. \tag{5.32b}$$

Remark 5.13. We can extend this result to the linear, non-autonomous, non-homogeneous test problem

$$\mathbf{M}(t)\, y' = \lambda(t)\, y + \phi(t). \tag{5.33}$$

Similarly, the linearly implicit GLM eq. (5.10) applied to eq. (5.33) using the exact Jacobians $\mathbf{L}_i = \lambda(t_{n-1} + \mathbf{c}\,h)$, computes stages as:

$$\begin{aligned} \mathbf{M}(t)\, K &= h\,\lambda(t_{n-1} + \mathbf{c}\,h)\left(\widehat{\mathbf{A}}\,K + \widehat{\mathbf{U}}\,\xi^{[n-1]}\right) + \phi(t_{n-1} + \mathbf{c}\,h), \\ \mathrm{y}^{[n]} &= \mathbf{B}\,K + \mathbf{V}\,\mathrm{y}^{[n-1]}, \end{aligned} \tag{5.34}$$

where eq. (5.34) is exactly the stages of the underlying implicit method applied to eq. (5.33). As a result the stability properties of linearly implicit GLM for this problem is the same as its underlying implicit GLM.

The linear stability analysis in this section gives us a picture of how the accuracy of the approximate Jacobian $\mathbf{L}$ and it's closeness to the exact Jacobian $\mathbf{J}$ of the system affects the stability of the method. We note that in the best case scenario the stability of the linearly-implicit GLM is as good as an implicit method, when exact Jacobian is used. On the other hand when the Jacobian does not contribute to the linear system solutions ($\mathbf{L} = \mathbf{0}$) the linear stability reduces to that of an explicit method.

5.6 Stiff accuracy of linearly implicit GLMs

5.6.1 Convergence of linearly implicit GLMs applied to the Prothero Robinson problem

The Prothero–Robinson problem [82] is a linear scalar problem of the form

$$y' = \mu \left(y - \phi \left(t \right) \right) + \phi' \left(t \right). \tag{5.35}$$

The stages for the method in eq. (5.11) applied to eq. (5.35) are

$$\begin{aligned} K &= z \left(\mathbf{A}\, K + \mathbf{U} \mathbb{y}^{[n-1]} - \phi \left(t_{n-1} + \mathbf{c}\, h \right) \right) \\ &\quad + h\, \phi' \left(t_{n-1} + \mathbf{c}\, h \right) + z\, \mathbf{\Psi}\, \mathbb{y}^{[n-1]} + z\, \mathbf{\Gamma}\, K, \\ &= z \left(\widehat{\mathbf{A}}\, K + \widehat{\mathbf{U}} \mathbb{y}^{[n-1]} - \phi \left(t_{n-1} + \mathbf{c}\, h \right) \right) + h\, \phi' \left(t_{n-1} + \mathbf{c}\, h \right), \qquad (5.36a) \\ \mathbb{y}^{[n]} &= \mathbf{B}\, K + \mathbf{V}\, \mathbb{y}^{[n-1]}, \qquad (5.36b) \end{aligned}$$

where $z = h\mu$. Due to the linearity of the problem, the stages in eq. (5.36) are exactly the same as stages of the underlying implicit method applied to the Prothero-Robinson problem. Convergence results therefore follow from the traditional GLMs [12, Section 4] and [132, Theroem 3]. Assuming $\mathbf{L}_i = \mu$ and denoting the global external stage errors by

$$e_{\xi}^{[n-1]} := \mathbb{y}^{[n-1]} - \mathbf{W}\, \eta \left(t_{n-1} \right),$$

we present the following corollaries:

Corollary 5.14 (Non-stiff PR-problem)**.** *For $z \in \widehat{S}$, $z = h\mu$ and $-\infty < Re(\mu) < 0$ the linearly implicit GLM applied to the PR-problem eq. (5.35) is convergent with the external stage order p such that if $e_{\xi}^{[0]} = \mathcal{O}(h^{p+1})$,*

$$e_{\xi}^{[n]} = \mathcal{O}(h^{p}), \quad as \quad h \to 0.$$

This result is due to the fact that the non-stiff PR problem is a linear problem for which the stages of the linearly-implicit and the underlying implicit methods are the same.

Corollary 5.15 (Stiff PR-problem)**.** *For $z \in \widehat{S}$, $z = h\mu$ and $\mu \to -\infty$, if $\widehat{\mathbf{A}}$ is invertible the linearly implicit GLM applied to the PR-problem eq. (5.35) is convergent with order* $\min(q, p)$ *such that if* $e_\xi^{[0]} = \mathcal{O}(h^{p+1})$

$$e_\xi^{[n]} = \mathcal{O}\big(h^{\min(q,p)}\big), \quad as \quad h \to 0.$$

This results follows from [12, Theorem 4.2] for implicit GLMs. As $z \to -\infty$ and only equipped with the uniform boundedness of $\left\|\widehat{\mathbf{M}}(\infty)^n\right\|$, the error recurrence compounds the error over n steps leading to the above estimates.

Corollary 5.16 (Stiffly accurate linearly implicit GLM)**.** *Under the assumptions of proposition 5.15, sharper estimates of the convergence rate are available when the method additionally satisfies $\rho(\widehat{\mathbf{M}}(\infty)) < 1$ such that if $e_\xi^{[0]} = \mathcal{O}(h^{p+1})$*

$$e_\xi^{[n]} = \mathcal{O}\big(h^{\min(q+1,p)}\big), \quad as \quad h \to 0.$$

This results follows from [12, Theorem 4.2] for implicit GLMs.

5.6.2 Convergence of linearly implicit GLMs applied to singularly perturbed problems

In this section we study the convergence of linearly implicit GLMs on the singular perturbation problem

$$\begin{bmatrix} y' \\ z' \end{bmatrix} = \begin{bmatrix} f(y,z) \\ \varepsilon^{-1} g(y,z) \end{bmatrix}, \tag{5.37}$$

with $\varepsilon \ll 1$, f and g sufficiently smooth and $g_z(y,z)$ invertible in the neighbourhood of the solution with a bounded logarithmic norm $\mu(g_z(y,z)) \leq -1$ [100, Equation 23]. To apply the linearly implicit GLM (5.11) to (5.37) we use the following notation: Y and Z refer to the function arguments, K and S denote the internal stage values, and $\mathbb{y}^{[n-1]}$ and $\mathbb{z}^{[n-1]}$ represent the incoming external stages at step n. The numerical method reads as

$$Y := \mathbf{A} \circledast K + \mathbf{U} \circledast \mathbb{y}^{[n-1]}, \tag{5.38a}$$

$$Z := \mathbf{A} \circledast S + \mathbf{U} \circledast \mathbb{z}^{[n-1]}, \tag{5.38b}$$

$$\begin{aligned}\begin{bmatrix} K \\ \varepsilon\, S \end{bmatrix} &= h \begin{bmatrix} F\left(Y, Z\right) \\ G\left(Y, Z\right) \end{bmatrix} + h \begin{bmatrix} F_y \left(\mathbf{\Gamma} \circledast K\right) + F_z \left(\mathbf{\Gamma} \circledast S\right) \\ G_y \left(\mathbf{\Gamma} \circledast K\right) + G_z \left(\mathbf{\Gamma} \circledast S\right) \end{bmatrix} \\ &\quad + h \begin{bmatrix} F_y \left(\mathbf{\Psi} \circledast \mathbb{y}^{[n]}\right) + F_z \left(\mathbf{\Psi} \circledast \mathbb{y}^{[n-1]}\right) \\ G_y \left(\mathbf{\Psi} \circledast \mathbb{z}^{[n]}\right) + G_z \left(\mathbf{\Psi} \circledast \mathbb{z}^{[n-1]}\right) \end{bmatrix}, \end{aligned} \tag{5.38c}$$

$$\begin{bmatrix} \mathbb{y}^{[n]} \\ \mathbb{z}^{[n]} \end{bmatrix} = \begin{bmatrix} \mathbf{B} \circledast K + \mathbf{V} \circledast \mathbb{y}^{[n-1]} \\ \mathbf{B} \circledast S + \mathbf{V} \circledast \mathbb{z}^{[n-1]} \end{bmatrix}, \tag{5.38d}$$

where we use $F_y = \mathbf{I}_{s\times s} \otimes f_y(y_{n-1}, z_{n-1})$ to denote the sub-Jacobian evaluation and similar notation for F_z, G_y, and G_z in relation to eq. (5.12) is adopted. We first consider convergence of linearly implicit GLMs for index-1 DAEs. Later, we will extend the result to singularly perturbed problems.

Index-1 DAEs

In the limit $\varepsilon \to 0$, eq. (5.38) turns into the index-1 DAE:

$$\begin{aligned} y' &= f\left(y, z\right), \\ 0 &= g\left(y, z\right). \end{aligned}$$

We will consider the numerical values of the stages of a linearly implicit GLM satisfying order (q, p) conditions applied to this problem and compare numerical stages to the same equations when exact values are substituted. We will then derive an error recurrence that will determine the convergence rates for the differential and algebraic variables. Let us start with eq. (5.38) and take the limit $\varepsilon \to 0$:

$$\begin{aligned} K &= h\, F\left(Y, Z\right) + h\, \left(F_y \left(\mathbf{\Gamma} \circledast K\right) + F_z \left(\mathbf{\Gamma} \circledast S\right)\right) \\ &\quad + h\, \left(F_y \left(\mathbf{\Psi} \circledast \mathbb{y}^{[n-1]}\right) + F_z \left(\mathbf{\Psi} \circledast \mathbb{z}^{[n-1]}\right)\right), \end{aligned} \tag{5.39a}$$

$$\begin{aligned} 0 &= G\left(Y, Z\right) + G_y \left(\mathbf{\Gamma} \circledast K\right) + G_z \left(\mathbf{\Gamma} \circledast S\right) \\ &\quad + G_y \left(\mathbf{\Psi} \circledast \mathbb{y}^{[n-1]}\right) + G_z \left(\mathbf{\Psi} \circledast \mathbb{z}^{[n-1]}\right). \end{aligned} \tag{5.39b}$$

Define $\widetilde{Y}$ and $\widetilde{Z}$ as function arguments when exact values are inserted into eqs. (5.38a) and (5.38b). Considering the internal stage order conditions for the underlying explicit method we have that:

$$\begin{aligned} \widetilde{Y} &= h\mathbf{A} \circledast y'\left(t_{n-1} + \mathbf{c}\, h\right) + \left(\mathbf{UW}\right) \circledast \eta\left(t_{n-1}\right) \\ &= y(t_{n-1} + \mathbf{c}h) + \mathcal{O}\big(h^{q+1}\big), \end{aligned} \tag{5.40a}$$

$$\begin{aligned} \widetilde{Z} &= h\mathbf{A} \circledast z'\left(t_{n-1} + \mathbf{c}\, h\right) + \left(\mathbf{UW}\right) \circledast \zeta\left(t_{n-1}\right) \\ &= z(t_{n-1} + \mathbf{c}h) + \mathcal{O}\big(h^{q+1}\big), \end{aligned} \tag{5.40b}$$

where $\eta(t_{n-1})$ and $\zeta(t_{n-1})$ are Nordsieck vectors corresponding to the derivatives of $y(t)$ and $z(t)$ at the time t_{n-1} respectively. Furthermore, define the global errors as:

$$\begin{aligned}
\Delta K &:= K - hy'\left(t_{n-1} + \mathbf{c}\, h\right), \\
\Delta \mathbb{y}^{[n-1]} &:= \mathbb{y}^{[n-1]} - \mathbf{W} \circledast \eta\left(t_{n-1}\right), \\
\Delta Y := Y - \widetilde{Y} &= \mathbf{A} \circledast \left(K - hy'\left(t_{n-1} + \mathbf{c}\, h\right)\right) \\
&\quad + \mathbf{U} \circledast \mathbb{y}^{[n-1]} - (\mathbf{UW}) \circledast \eta\left(t_{n-1}\right) \\
&= \mathbf{A} \circledast \Delta K + \mathbf{U} \circledast \Delta \mathbb{y}^{[n-1]},
\end{aligned}$$

We assume similar notation for the errors in the algebraic variable $\Delta K, \Delta S$ and $\Delta \mathbb{z}^{[n-1]}$.

Now, we insert exact values in eq. (5.39a) and using the fact that the linearly implicit GLM satisfies the internal and external stage order conditions eq. (5.20) to write:

$$\begin{aligned}
hy'\left(t_{n-1} + \mathbf{c}\, h\right) = hF\left(\widetilde{Y}, \widetilde{Z}\right) &+ h^2 F_y\left(\mathbf{\Gamma} \circledast y'\left(t_{n-1} + \mathbf{c}\, h\right)\right) \\
&+ h^2 F_z\left(\mathbf{\Gamma} \circledast z'\left(t_{n-1} + \mathbf{c}\, h\right)\right) \\
&+ hF_y\left((\mathbf{\Psi W}) \circledast \eta(t_{n-1})\right) + hF_z\left((\mathbf{\Psi W}) \circledast \zeta(t_{n-1})\right) \\
&+ \mathcal{O}\big(h^{q+2}\big).
\end{aligned} \tag{5.42}$$

Subtracting eq. (5.42) from eq. (5.39a)

$$\begin{aligned}
\Delta K = hF\left(Y, Z\right) - hF\left(\widetilde{Y}, \widetilde{Z}\right) &\\
+ h\left(F_y\left(\mathbf{\Gamma} \circledast \Delta K\right) + F_z\left(\mathbf{\Gamma} \circledast \Delta S\right) + F_y\left(\mathbf{\Psi} \circledast \Delta \mathbb{y}^{[n-1]}\right) + F_z\left(\mathbf{\Psi} \circledast \Delta \mathbb{z}^{[n-1]}\right)\right) &\\
+ \mathcal{O}\big(h^{q+2}\big) + \mathcal{O}\big(h^{p+1}\big).&
\end{aligned} \tag{5.43}$$

From eq. (5.43) we have:

$$\begin{aligned}
\Delta K = h\left(F_y\left(\widehat{\mathbf{A}} \circledast \Delta K\right) + F_z\left(\widehat{\mathbf{A}} \circledast \Delta S\right)\right) &\\
+ h\left(F_y\left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{y}^{[n-1]}\right) + F_z\left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{z}^{[n-1]}\right)\right) &\\
+ \mathcal{O}\big(h^{\nu+1}\big),&
\end{aligned} \tag{5.44}$$

where $\nu = \min\left(q+1, p\right)$ and the order of $hF\left(Y, Z\right) - hF\left(\widetilde{Y}, \widetilde{Z}\right)$ is due to eq. (5.25). Repeating the same steps for the algebraic stages we get:

$$\begin{aligned}
0 &= G_y\left(\widehat{\mathbf{A}} \circledast \Delta K\right) + G_z\left(\widehat{\mathbf{A}} \circledast \Delta S\right) \\
&\quad + G_y\left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{y}^{[n-1]}\right) + G_z\left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{z}^{[n-1]}\right) \\
&\quad + \mathcal{O}(h^{\nu}), \\
\Delta S &= -\left(G_z \widehat{\mathbf{A}} \otimes \mathbf{I}_{d\times d}\right)^{-1}\left(\left(\widehat{\mathbf{A}} \circledast \Delta K\right) + G_y\left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{y}^{[n-1]}\right) + G_z\left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{z}^{[n-1]}\right)\right) \\
&\quad + \mathcal{O}(h^{\nu}),
\end{aligned} \tag{5.45}$$

Solving for $\widehat{\mathbf{A}} \circledast \Delta S + \widehat{\mathbf{U}} \circledast \Delta \mathbb{z}^{[n-1]}$ in eq. (5.45) and inserting into eq. (5.44) we get

$$\begin{aligned} \Delta K &= h\mathbf{J}^{\text{red}} \left(\widehat{\mathbf{A}} \circledast \Delta K \right) + h\mathbf{J}^{\text{red}} \left(\widehat{\mathbf{U}} \circledast \Delta \mathbb{y}^{[n-1]} \right) + \mathcal{O}\big(h^{\nu+1}\big), \\ \Delta \mathbb{y}^{[n]} &= \mathbf{B} \circledast \Delta K + \widehat{\mathbf{U}} \circledast \Delta \mathbb{y}^{[n-1]}. \end{aligned} \tag{5.46}$$

Equation (5.46) describes the global error for the underlying implicit method applied to the linear ODE $y' = \mathbf{J}^{\text{red}} y$ but with local truncation error of $\mathcal{O}(h^{\nu+1})$. Therefore, by classical convergence properties of GLMs leads to $\Delta \mathbb{y}^{[n]} = \mathcal{O}(h^{\nu})$.

Assuming G_z and $\widehat{\mathbf{A}}$ are invertible, we define some new notations that will become useful:

$$\begin{aligned} \mathbf{J}^{\text{red}} &= F_y - F_z\, G_z^{-1}\, G_y = \mathbf{I}_{s\times s} \otimes \left(f_y - f_z g_z^{-1} g_y \right) \\ &= \mathbf{I}_{s\times s} \otimes \mathcal{J}, \end{aligned}$$

noting that this the Jacobian of the index-reduced system obtained by differentiating the DAE. Rearranging eqs. (5.44) and (5.45) we have

$$\begin{bmatrix} \Delta K \\ \Delta S \end{bmatrix} = \mathbf{D}(h)^{-1} \left(\begin{bmatrix} hF_y & hF_z \\ G_y & G_z \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{U}} \circledast \Delta \mathbb{y}^{[n-1]} \\ \widehat{\mathbf{U}} \circledast \Delta \mathbb{z}^{[n-1]} \end{bmatrix} + \begin{bmatrix} \mathcal{O}(h^{\nu+1}) \\ \mathcal{O}(h^{\nu}) \end{bmatrix} \right), \tag{5.47}$$

where

$$\mathbf{D}(h) = \begin{bmatrix} \mathbf{I}_{sd\times sd} - h\widehat{\mathbf{A}} \otimes f_y & -h\widehat{\mathbf{A}} \otimes f_z \\ -\widehat{\mathbf{A}} \otimes g_y & -\widehat{\mathbf{A}} \otimes g_z \end{bmatrix},$$

with the Schur complement,

$$\begin{aligned} \mathbf{D}(h)^{-1} &= \begin{bmatrix} \mathbf{D}_{1,1}^{-1} & \mathbf{D}_{1,2}^{-1} \\ \mathbf{D}_{2,1}^{-1} & \mathbf{D}_{2,2}^{-1} \end{bmatrix}, \\ \mathbf{D}_{1,1}^{-1} &= \mathbf{I}_{sd\times sd} - h\widehat{\mathbf{A}} \otimes \mathcal{J}, \\ \mathbf{D}_{1,2}^{-1} &= -h \left(\mathbf{I}_{sd\times sd} - h\widehat{\mathbf{A}} \otimes \mathcal{J} \right)^{-1} \left(\mathbf{I}_{s\times s} \otimes \left(f_z g_z^{-1} \right) \right) \\ \mathbf{D}_{2,1}^{-1} &= -\left(\mathbf{I}_{s\times s} \otimes \left(g_z^{-1} g_y \right) \right) \left(\mathbf{I}_{sd\times sd} - h\widehat{\mathbf{A}} \otimes \mathcal{J} \right)^{-1}, \\ \mathbf{D}_{2,2}^{-1} &= -\widehat{\mathbf{A}}^{-1} \otimes g_z^{-1} + h \left(\mathbf{I}_{s\times s} \otimes \left(g_z^{-1} g_y \right) \right) \left(\mathbf{I}_{sd\times sd} - h\widehat{\mathbf{A}} \otimes \mathcal{J} \right)^{-1} \left(\mathbf{I}_{s\times s} \otimes \left(f_z g_z^{-1} \right) \right). \end{aligned}$$

Note that

$$\mathbf{D}(h)^{-1} = \begin{bmatrix} \mathcal{O}(1) & \mathcal{O}(h) \\ \mathcal{O}(1) & \mathcal{O}(1) \end{bmatrix}.$$

$$\begin{bmatrix} \Delta K \\ \Delta S \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{1,1}^{-1} & \mathbf{D}_{1,2}^{-1} \\ \mathbf{D}_{2,1}^{-1} & \mathbf{D}_{2,2}^{-1} \end{bmatrix} \begin{bmatrix} h \left(\widehat{\mathbf{U}} \otimes f_y \right) \Delta \mathbb{y}^{[n-1]} + h \left(\widehat{\mathbf{U}} \otimes f_z \right) \Delta \mathbb{z}^{[n-1]} \\ \left(\widehat{\mathbf{U}} \otimes g_y \right) \Delta \mathbb{y}^{[n-1]} + \left(\widehat{\mathbf{U}} \otimes g_z \right) \Delta \mathbb{z}^{[n-1]} \end{bmatrix} + \begin{bmatrix} \mathcal{O}(h^{\nu+1}) \\ \mathcal{O}(h^{\nu}) \end{bmatrix},$$

Inserting eq. (5.47) into eq. (5.38d) we get the error recurrence:

$$\begin{aligned}\begin{bmatrix}\Delta \mathbb{y}^{[n]}\\ \Delta \mathbb{z}^{[n]}\end{bmatrix} &= \begin{bmatrix}\mathbf{B} \circledast \Delta K + \mathbf{V} \circledast \Delta \mathbb{y}^{[n-1]}\\ \mathbf{B} \circledast \Delta S + \mathbf{V} \circledast \Delta \mathbb{z}^{[n-1]}\end{bmatrix}\\ &= \begin{bmatrix}\widehat{\mathbf{M}}(\mathcal{Z}) & \mathbf{0}\\ \mathcal{R}(\mathcal{Z}) & \widehat{\mathbf{M}}(\infty)\end{bmatrix}\begin{bmatrix}\Delta \mathbb{y}^{[n-1]}\\ \Delta \mathbb{z}^{[n-1]}\end{bmatrix} + \begin{bmatrix}\mathcal{O}(h^{\nu+1})\\ \mathcal{O}(h^{\nu})\end{bmatrix}.\end{aligned} \tag{5.48}$$

where

$$\begin{aligned}\mathcal{Z} &= h\mathbf{J}^{\mathrm{red}},\\ \widehat{\mathbf{M}}(\mathcal{Z}) &= \mathbf{V}\otimes \mathbf{I}_{d\times d} + h\mathbf{B} \circledast \left(\mathbf{I}_{sd\times sd} - \mathcal{Z}\left(\widehat{\mathbf{A}}\otimes \mathbf{I}_{d\times d}\right)\right)^{-1}\mathcal{Z}\widehat{\mathbf{U}}\\ &= \mathbf{V}\otimes \mathbf{I}_{d\times d} + h\mathbf{B} \circledast \left(\mathbf{I}_{sd\times sd} - h\widehat{\mathbf{A}}\otimes \mathcal{J}\right)^{-1}\left(\widehat{\mathbf{U}}\otimes \mathcal{J}\right).\end{aligned}$$

Convergence of the error recurrence in eq. (5.48) depends on the eigenvalues of the error amplification matrix which in this case has lower block-triangular structure. As a result convergence is determined by the eigenvalues of $\widehat{\mathbf{M}}(\mathcal{Z})$ including at infinity. A detailed discussion of various cases can be found in [100, Lemma 2]. Here, we use the following corollary.

Corollary 5.17. *Assume g_z is invertible in a neighborhood of the solution, and also assume we start from an algebraically consistent initial condition $g(y_0, z_0) = 0$. If the underlying implicit method satisfies $\rho(\widehat{\mathbf{M}}(\infty)) < 1$ and is power bounded elsewhere then the global error of linearly implicit GLM (5.11) applied to the index-1 DAE problem satisfies:*

$$\begin{aligned}\mathbb{y}^{[n]} - y(t_n) &= O(h^{\nu}), \qquad & K - hy'(t_{n-1} + \mathbf{c}\,h) &= O(h^{\nu+1}),\\ \mathbb{z}^{[n]} - z(t_n) &= O(h^{\nu}), \qquad & S - hz'(t_{n-1} + \mathbf{c}\,h) &= O(h^{\nu+1}).\end{aligned}$$

Proof. Simple application of [100, Lemma 2] to estimate the global errors leads to $\mathbb{y}^{[n]} - y(t_n) = O(h^{\nu})$ and $\mathbb{z}^{[n]} - z(t_n) = O(h^{\nu})$. For internal stages we use eq. (5.47) to get:

$$\begin{bmatrix}\Delta K\\ \Delta S\end{bmatrix} = \begin{bmatrix}\mathcal{O}(h) & \mathcal{O}(h)\\ \mathcal{O}(1) & \mathcal{O}(1)\end{bmatrix}\begin{bmatrix}\Delta \mathbb{y}^{[n-1]}\\ \Delta \mathbb{z}^{[n-1]}\end{bmatrix} + \begin{bmatrix}\mathcal{O}(h^{\nu+1})\\ \mathcal{O}(h^{\nu})\end{bmatrix}.$$

If global error estimates for the external stages are substituted in this equation we arrive at $\Delta K = \mathcal{O}(h^{\nu+1})$ and $\Delta S = \mathcal{O}(h^{\nu+1})$. □

$\varepsilon-$expansion of smooth solutions

We are now interested in smooth solutions of the singular perturbation problem eq. (5.37) that can be expanded according to

$$y(t) = \sum_{k\geq 0} y^k(t)\varepsilon^k, \qquad z(t) = \sum_{k\geq 0} z^k(t)\varepsilon^k, \tag{5.49}$$

where solution components $y^k(t)$ and $z^k(t)$ are not dependent on ε. We assume that the initial values $\mathbb{y}^{[0]}, \mathbb{z}^{[0]}$ are on the smooth solution manifold. Following te standard analysis we expand the function arguments and internal and external stages of the numerical solution in powers of ε:

$$Y = \sum_{i\geq 0} Y^i \varepsilon^i, \qquad Z = \sum_{i\geq 0} Z^i \varepsilon^i,$$
$$K = \sum_{i\geq 0} K^i \varepsilon^i, \qquad S = \sum_{i\geq 0} S^i \varepsilon^i,$$
$$\mathbb{y}^{[n]} = \sum_{i\geq 0} \mathbb{y}^{[n],i} \varepsilon^i, \qquad \mathbb{z}^{[n]} = \sum_{i\geq 0} \mathbb{z}^{[n],i} \varepsilon^i.$$

Inserting eq. (5.49) into eq. (5.37) and separating $\mathcal{O}(1)$ terms leads to the index-1 DAE discussed in section 5.6.2:

$$y^{0'} = f\left(y^0, z^0\right), \tag{5.50a}$$
$$0 = g\left(y^0, z^0\right), \tag{5.50b}$$

for which the convergence of linearly implicit GLMs is already established. If $\mathcal{O}(\varepsilon^1)$ terms are considered, in addition to eq. (5.50) we get the index-2 DAE:

$$y^{1'} = f_y\left(y^0, z^0\right) y^1 + f_z\left(y^0, z^0\right) z^1, \tag{5.51a}$$
$$z^{0'} = g_y\left(y^0, z^0\right) y^1 + g_z\left(y^0, z^0\right) z^1. \tag{5.51b}$$

Assuming g_z is invertible we can insert z^1 from eq. (5.51b) into eq. (5.51a) to get the reduced ODE:

$$z^1 = -\left(g_z^{-1} g_y\right) y^1 + g_z^{-1} z^{0'}, \tag{5.52a}$$
$$y^{1'} = \left(f_y - f_z g_z^{-1} g_y\right) y^1 + \left(f_z g_z^{-1}\right) z^{0'}. \tag{5.52b}$$

All Jacobians in eq. (5.52) are evaluated at $(y^0(t), z^0(t))$ and we have dropped the argument for brevity. Note that eq. (5.52b) is a linear ODE in $y^1(t)$ with a time-dependent term on the right-hand side function. In general the differential-algebraic equation for component ν of the SPP problem can be represented by [51, Section VI.3]

$$y^{\nu'} = f_y\left(y^0, z^0\right) y^\nu + f_z\left(y^0, z^0\right) z^\nu + \phi^\nu\left(y^0, z^0, \ldots, y^{\nu-1}, z^{\nu-1}\right), \tag{5.53a}$$
$$z^{\nu-1'} = g_y\left(y^0, z^0\right) y^\nu + g_z\left(y^0, z^0\right) z^\nu + \psi^\nu\left(y^0, z^0, \ldots, y^{\nu-1}, z^{\nu-1}\right). \tag{5.53b}$$

In [108] Schneider shows the convergence of SPP components for general linear methods. Here, we use the linearity of the systems in eq. (5.53) to show that the stages of linearly implicit GLM applied to these problems are within the local truncation errors of the stages of the underlying implicit method applied to the same system when the exact Jacobian is used. Convergence of the SPP problem, therefore, follows the same asymptotic as the implicit GLM.

Theorem 5.18 (Convergence of linearly implicit GLMs for the $\varepsilon-$expansion terms of SPP)**.** *For an internally consistent linearly implicit GLM of order (q,p), with the underlying implicit method satisfying $\rho(\mathbf{M}(\infty)) < 1$, $\mathbf{M}(z)$ power bounded elsewhere, and eigenvalues of the coefficient matrix $\widehat{\mathbf{A}}$ having positive real values, and further, assuming starting with consistent initial values*

$$\begin{aligned} \Delta \mathbb{y}^{\nu,[n]} &= \mathcal{O}\big(h^{q+2-\nu}\big), \quad \Delta Y^{\nu,[n]} = \mathcal{O}\big(h^{q+2-\nu}\big), \\ \Delta \mathbb{z}^{\nu,[n]} &= \mathcal{O}\big(h^{q+1-\nu}\big), \quad \Delta Z^{\nu,[n]} = \mathcal{O}\big(h^{q+1-\nu}\big), \end{aligned} \tag{5.54}$$

for $1 \le \nu \le q+1$.

Proof. We will present the proof for $\nu = 1$. For $\nu > 1$ an induction argument can be constructed. Starting at $\nu = 1$ the index-2 problem reads as:

$$y^{1'} = f_y\left(y^0, z^0\right) y^1 + f_z\left(y^0, z^0\right) z^1 = \mathcal{F}(y^1, z^1), \tag{5.55a}$$
$$z^{0'} = g_y\left(y^0, z^0\right) y^1 + g_z\left(y^0, z^0\right) z^1 = \mathcal{G}(y^1, z^1). \tag{5.55b}$$

We like to show that the difference between the linearly implicit GLM and the underlying implicit stages remain bounded. Consider the implicit method's stages:

$$\begin{aligned} \widehat{Y}^1 &= \widehat{\mathbf{A}} \otimes \widehat{K}_y^1 + \widehat{\mathbf{U}} \otimes \widehat{\mathbb{y}}^{1,[n-1]}, & \widehat{Z}^1 &= \widehat{\mathbf{A}} \otimes \widehat{K}_z^1 + \widehat{\mathbf{U}} \otimes \widehat{\mathbb{z}}^{1,[n-1]}, \\ \widehat{K}_y^1 &= h\mathcal{F}\left(\widehat{Y}^1, \widehat{Z}^1\right), & \widehat{K}_z^0 &= h\mathcal{G}\left(\widehat{Y}^1, \widehat{Z}^1\right). \end{aligned}$$

The stage values according to eq. (5.55) are

$$\begin{aligned} \widehat{K}^1 &= hF_y\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \widehat{K}^1 + \widehat{\mathbf{U}} \otimes \widehat{\mathbb{y}}^{1,[n-1]}\right) \\ &\quad + hF_z\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \widehat{S}^1 + \widehat{\mathbf{U}} \otimes \widehat{\mathbb{y}}^{1,[n-1]}\right), \\ \widehat{S}^0 &= hG_y\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \widehat{K}^1 + \widehat{\mathbf{U}} \otimes \widehat{\mathbb{y}}^{1,[n-1]}\right) \\ &\quad + hG_z\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \widehat{S}^1 + \widehat{\mathbf{U}} \otimes \widehat{\mathbb{z}}^{1,[n-1]}\right), \end{aligned}$$

where $\widehat{Y}^0$ and $\widehat{Z}^0$ are the index-1 solutions computed by the implicit method. Similarly, the

linearly implicit GLM stages read as

$$
\begin{aligned}
Y^1 &= \mathbf{A} \otimes K_y^1 + \mathbf{U} \otimes \mathbb{y}^{1,[n-1]},\\
Z^1 &= \mathbf{A} \otimes K_z^1 + \mathbf{U} \otimes \mathbb{z}^{1,[n-1]},\\
K_y^1 &= h\mathcal{F}\left(Y^1, Z^1\right) + \mathcal{F}_y\left(Y^1, Z^1\right)\left(\mathbf{\Gamma} \otimes K^1\right) + \mathcal{F}_z\left(Y^1, Z^1\right)\left(\mathbf{\Gamma} \otimes S^1\right)\\
&\quad + \mathcal{F}_y\left(Y^1, Z^1\right)\left(\mathbf{\Psi} \otimes \mathbb{y}^{1,[n-1]}\right) + \mathcal{F}_z\left(Y^1, Z^1\right)\left(\mathbf{\Psi} \otimes \mathbb{z}^{1,[n-1]}\right)\\
&= hF_y\left(Y^0, Z^0\right)\left(\widehat{\mathbf{A}} \otimes K^1 + \widehat{\mathbf{U}} \otimes \mathbb{y}^{1,[n-1]}\right)\\
&\quad + hF_z\left(Y^0, Z^0\right)\left(\widehat{\mathbf{A}} \otimes S^1 + \widehat{\mathbf{U}} \otimes \mathbb{z}^{1,[n-1]}\right),\\
S^0 &= hG_y\left(Y^0, Z^0\right)\left(\widehat{\mathbf{A}} \otimes K^1 + \widehat{\mathbf{U}} \otimes \mathbb{y}^{1,[n-1]}\right)\\
&\quad + hG_z\left(Y^0, Z^0\right)\left(\widehat{\mathbf{A}} \otimes S^1 + \widehat{\mathbf{U}} \otimes \mathbb{z}^{1,[n-1]}\right),
\end{aligned}
$$

Convergence of index-1 problems for linearly implicit GLMs is discussed in section 5.6.2. Since we arrive at the same rates as the underlying implicit method [108, Section 4], we can change the argument of the sub-Jacobians such that

$$
Y^0 = \widehat{Y}^0 + \mathcal{O}\big(h^{q+1}\big) \quad \textit{and} \quad Z^0 = \widehat{Z}^0 + \mathcal{O}\big(h^{q+1}\big).
$$

Defining $\delta K^1 := K^1 - \widehat{K}^1$ *and similar notation for the error between stages of the underlying implicit GLM and the linearly implicit GLM we have*

$$
\begin{aligned}
\delta K^1 &= hF_y\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \delta K^1 + \widehat{\mathbf{U}} \otimes \delta\mathbb{y}^{1,[n-1]}\right) \qquad (5.58a)\\
&\quad + hF_z\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \widehat{S}^1 + \widehat{\mathbf{U}} \otimes \delta\mathbb{y}^{1,[n-1]}\right) + \mathcal{O}\big(h^{q+2}\big),\\
\delta S^0 &= hG_y\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \delta K^1 + \widehat{\mathbf{U}} \otimes \delta\mathbb{y}^{1,[n-1]}\right) \qquad (5.58b)\\
&\quad + hG_z\left(\widehat{Y}^0, \widehat{Z}^0\right)\left(\widehat{\mathbf{A}} \otimes \delta S^1 + \widehat{\mathbf{U}} \otimes \delta\mathbb{z}^{1,[n-1]}\right) + \mathcal{O}\big(h^{q+2}\big),\\
\delta\mathbb{y}^1 &= h\widehat{\mathbf{A}} \otimes \delta K^1 + \widehat{\mathbf{U}} \otimes \delta\mathbb{y}^{1,[n-1]}, \qquad (5.58c)\\
\delta\mathbb{z}^1 &= h\widehat{\mathbf{A}} \otimes \delta S^1 + \widehat{\mathbf{U}} \otimes \delta\mathbb{z}^{1,[n-1]}. \qquad (5.58d)
\end{aligned}
$$

Note that in eq. (5.58) the error between the two methods has the same structure as the local truncation error of the implicit GLM [108, Therorem 2.1] and [51, Theorem 3.4]. Therefore under the same assumptions for convergence, we have the following result:

$$
\begin{aligned}
\Delta K^1 = \delta K^1 + \Delta\widehat{K}^1 = \mathcal{O}\big(h^{q+2}\big), \quad \Delta S^1 = \delta S^1 + \Delta\widehat{S}^1 = \mathcal{O}\big(h^{q+1}\big), \qquad (5.59a)\\
\Delta\mathbb{y}^1 = \delta\mathbb{y}^1 + \Delta\widehat{\mathbb{y}}^1 = \mathcal{O}\big(h^{q+1}\big), \quad \Delta\mathbb{z}^1 = \delta\mathbb{z}^1 + \Delta\widehat{\mathbb{z}}^1 = \mathcal{O}(h^q). \qquad (5.59b)
\end{aligned}
$$

Corollary 5.19 (Estimation of the remainder in the numerical solution)**.** *Under the same assumptions as in proposition 5.18, the global error of linearly implicit GLM applied to the SPP problem eq. (5.37) satisfies:*

$$\begin{aligned} y_n - y(t_n) &= \mathcal{O}(h^p + \varepsilon h^q), \\ z_n - z(t_n) &= \mathcal{O}(h^p + \varepsilon h^q), \end{aligned} \tag{5.60}$$

for some $h/\varepsilon \geq D, h \leq h_0$ with h_0 independent of ε.

Proof. This result directly follows from [108, Theorem 2.4], having shown the $\varepsilon-$expansion terms in the stages of linearly implicit GLM have the same error convergence rate as the underlying implicit method.

5.7 Special families of linearly implicit GLMs

In this section we study a number of existing linearly-implicit GLMs from the literature and show how they conform to the formulation in eq. (5.11). We also introduce new families of W-methods based on available implicit GLMs.

5.7.1 Two-step Runge–Kutta and Peer methods

A general formulation of Two-step Runge–Kutta methods and Peer methods [9, 60] may treat both families as special cases of GLMs. Let us consider a method that advance the solution y_{n+1} as:

$$\begin{aligned} Y^{[n]} &= \mathbf{u}^{[n-2]} \circledast y_{n-2} + \mathbf{u}^{[n-1]} \circledast y_{n-1} + \mathbf{D}^{[n-1]} \circledast Y^{[n-1]} \\ &\quad + h\,\mathbf{A}^{[n]} \circledast F\left(Y^{[n]}\right) + h\,\mathbf{A}^{[n-1]} \circledast F\left(Y^{[n-1]}\right), \end{aligned} \tag{5.61a}$$

$$\begin{aligned} y_n &= \vartheta^{[n-2]}\, y_{n-2} + \vartheta^{[n-1]}\, y_{n-1} + \boldsymbol{\theta}^{[n-1]\,T} \circledast Y^{[n-1]} \\ &\quad + h\,\mathbf{v}^{[n]\,T} \circledast F\left(Y^{[n]}\right) + h\,\mathbf{v}^{[n-1]\,T} \circledast F\left(Y^{[n-1]}\right). \end{aligned} \tag{5.61b}$$

The method (5.61) is a GLM with the tableau of coefficients:

$$\begin{array}{c|c} \mathbf{A} & \mathbf{U} \\ \hline \mathbf{B} & \mathbf{V} \end{array} = \begin{array}{c|cccc} \mathbf{A}^{[n]} & \mathbf{u}^{[n-2]} & \mathbf{u}^{[n-1]} & \mathbf{D}^{[n-1]} & \mathbf{A}^{[n-1]} \\ \hline \mathbf{0}_{s\times s} & \mathbf{0}_{s\times 1} & \mathbf{1}_s & \mathbf{0}_{s\times s} & \mathbf{0}_{s\times s} \\ \mathbf{v}^{[n]\,T} & \vartheta^{[n-2]} & \vartheta^{[n-1]} & \boldsymbol{\theta}^{[n-1]\,T} & \mathbf{v}^{[n-1]\,T} \\ \mathbf{A}^{[n]} & \mathbf{u}^{[n-2]} & \mathbf{u}^{[n-1]} & \mathbf{D}^{[n-1]} & \mathbf{A}^{[n-1]} \\ \mathbf{I}_{d\times d} & \mathbf{0}_{s\times 1} & \mathbf{0}_{s\times 1} & \mathbf{0}_{s\times s} & \mathbf{0}_{s\times s} \end{array}, \tag{5.62}$$

The external stages contain solution and function values:

$$\mathbb{y}^{[n-1]} = \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ Y^{[n-1]} \\ hF\left(Y^{[n-1]}\right) \end{bmatrix} \in \mathbb{R}^{(2s+2)d\times 1},$$

and using Taylor series expansion of the external stages we have the following for the columns of the $\mathbf{W}$ matrix:

$$\mathbf{w}_0 = \begin{bmatrix} 1 & 1 & \mathbf{1}_{s\times 1} & \mathbf{0}_{s\times 1} \end{bmatrix}^T, \tag{5.63a}$$

$$\mathbf{w}_k = \begin{bmatrix} \frac{(-1)^k}{k!} & 0 & \frac{(\mathbf{c}-\mathbf{1}_s)^k}{k!} & \frac{(\mathbf{c}-\mathbf{1}_s)^{k-1}}{(k-1)!} \end{bmatrix}^T, \qquad k = 1, \ldots, p. \tag{5.63b}$$

The pre-consistency conditions for the method in eq. (5.62) is

$$\left(\mathbf{u}^{[n-2]} + \mathbf{u}^{[n-1]} + \mathbf{D}^{[n-1]}\right)\mathbf{1}_s = \mathbf{1}_s, \tag{5.64a}$$

$$\vartheta^{[n-2]} + \vartheta^{[n-1]} + \boldsymbol{\theta}^{[n-1]\,T}\mathbf{1}_s = 1, \tag{5.64b}$$

Using (5.63) in the GLM order conditions we get the following internal stage order conditions:

$$\begin{aligned} &\frac{\mathbf{c}^{\intercal k}}{k!} - \frac{\mathbf{A}^{[n]}\mathbf{c}^{\intercal(k-1)}}{(k-1)!} - \frac{\mathbf{A}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\intercal(k-1)}}{(k-1)!} \\ &\quad - \frac{\mathbf{D}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\intercal k}}{k!} - \mathbf{u}^{[n-2]}\frac{(-1)^k}{k!} = 0, \qquad k = 1, \ldots, q, \end{aligned} \tag{5.65}$$

Similarly, using the structure of the tableau eq. (5.62) and eq. (5.63) in the GLM external stage order conditions proposition 5.2 we have the following identities:

$$\sum_{\ell=0}^{k} \frac{(-1)^{k-\ell}}{(k-\ell)!\,\ell!} = 0, \qquad k = 1, \ldots, p, \tag{5.66a}$$

$$\begin{aligned} &\frac{1}{k!} - \frac{\mathbf{v}^{[n]\,T}\mathbf{c}^{\intercal(k-1)}}{(k-1)!} - \frac{\mathbf{v}^{[n-1]\,T}(\mathbf{c}-\mathbf{1}_s)^{\intercal(k-1)}}{(k-1)!} \\ &\quad - \vartheta^{[n-2]}\frac{(-1)^k}{k!} - \frac{\boldsymbol{\theta}^{[n-1]\,T}(\mathbf{c}-\mathbf{1}_s)^{\intercal k}}{k!} = 0, \qquad k = 1, \ldots, p, \end{aligned} \tag{5.66b}$$

$$\begin{aligned} &\sum_{\ell=0}^{k} \frac{(\mathbf{c}-\mathbf{1}_s)^{\intercal(k-\ell)}}{(k-\ell)!\,\ell!} - \frac{\mathbf{A}^{[n]}\mathbf{c}^{\intercal(k-1)}}{(k-1)!} - \frac{\mathbf{A}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\intercal(k-1)}}{(k-1)!} \\ &\quad - \frac{\mathbf{D}^{[n-1]}\mathbf{c}^{\intercal k}}{k!} - \mathbf{u}^{[n-2]}\frac{(-1)^k}{k!} = 0, \qquad k = 1, \ldots, q, \end{aligned} \tag{5.66c}$$

$$\sum_{\ell=0}^{k-1} \frac{(\mathbf{c}-\mathbf{1}_s)^{\intercal(k-\ell-1)}}{(k-\ell-1)!\,\ell!} - \frac{\mathbf{c}^{\intercal(k-1)}}{(k-1)!} = 0, \qquad k = 1, \ldots, p. \tag{5.66d}$$

Equations (5.66a) and (5.66d) are trivially satisfied. Equation (5.66c) is equivalent to the stage order equation eq. (5.65) since

$$\sum_{\ell=0}^{k} \frac{(\mathbf{c}-\mathbf{1}_s)^{\curlyvee(k-\ell)}}{(k-\ell)!\,\ell!} = \frac{\mathbf{c}^{\curlyvee k}}{k!}, \quad k=1,\dots,q.$$

We have the following result:

Remark 5.20 (TSRK/Peer order conditions)**.** A two-step Runge-Kutta method or a GLM peer method represented by formulation eq. (5.61) is order (q,p) if the coefficients satisfy the preconsistency conditions eq. (5.64a) and

$$\begin{aligned} &\frac{\mathbf{c}^{\curlyvee k}}{k!} - \frac{\mathbf{A}^{[n]}\mathbf{c}^{\curlyvee(k-1)}}{(k-1)!} - \frac{\mathbf{A}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\curlyvee(k-1)}}{(k-1)!} \\ &\quad - \frac{\mathbf{D}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\curlyvee k}}{k!} - \mathbf{u}^{[n-2]}\frac{(-1)^k}{k!} = 0, \qquad k=1,\dots,q, \end{aligned} \tag{5.67a}$$

$$\begin{aligned} &\frac{1}{k!} - \frac{\mathbf{v}^{[n]\,T}\mathbf{c}^{\curlyvee(k-1)}}{(k-1)!} - \frac{\mathbf{v}^{[n-1]\,T}(\mathbf{c}-\mathbf{1}_s)^{\curlyvee(k-1)}}{(k-1)!} \\ &\quad - \vartheta^{[n-2]}\frac{(-1)^k}{k!} - \frac{\boldsymbol{\theta}^{[n-1]\,T}(\mathbf{c}-\mathbf{1}_s)^{\curlyvee k}}{k!} = 0, \qquad k=1,\dots,p. \end{aligned} \tag{5.67b}$$

Linearly implicit TSRK/Peer methods

Once the TSRK/Peer methods are considered as GLMs, extension to linearly-implicit version is a straight-forward process. We consider the linearly-implicit method

$$\begin{aligned} Y^{[n]} &= \mathbf{u}^{[n-2]} \circledast y_{n-2} + \mathbf{u}^{[n-1]} \circledast y_{n-1} + \mathbf{A}^{[n]} \circledast K^{[n]} + \mathbf{A}^{[n-1]} \circledast K^{[n-1]}, \\ K^{[n]} &= hF\left(Y^{[n]}\right) + h\mathbf{L}\left(\boldsymbol{\Gamma}^{[n]} \circledast K^{[n]} + \boldsymbol{\Gamma}^{[n-1]} \circledast K^{[n-1]}\right) \\ &\quad + h\mathbf{L}\left(\widetilde{\mathbf{u}}^{[n-2]} \circledast y_{n-2} + \widetilde{\mathbf{u}}^{[n-1]} \circledast y_{n-1}\right), \\ y_n &= \vartheta^{[n-2]}\, y_{n-2} + \vartheta^{[n-1]}\, y_{n-1} \\ &\quad + \mathbf{v}^{[n-1]\,T} \circledast K^{[n-1]} + \mathbf{v}^{[n]\,T} \circledast K^{[n]}. \end{aligned} \tag{5.68}$$

The order conditions are:

$$\frac{\mathbf{c}^{\curlywedge k}}{k!} - \frac{\mathbf{A}^{[n]}\mathbf{c}^{\curlywedge(k-1)}}{(k-1)!} - \frac{\mathbf{A}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\curlywedge(k-1)}}{(k-1)!} - \mathbf{u}^{[n-2]}\frac{(-1)^k}{k!} = 0, \tag{5.69a}$$
$$k = 1, \ldots, q,$$

$$\frac{1}{k!} - \frac{\mathbf{v}^{[n]\,T}\mathbf{c}^{\curlywedge(k-1)}}{(k-1)!} - \frac{\mathbf{v}^{[n-1]\,T}(\mathbf{c}-\mathbf{1}_s)^{\curlywedge(k-1)}}{(k-1)!} - \vartheta^{[n-2]}\frac{(-1)^k}{k!} = 0, \tag{5.69b}$$
$$k = 1, \ldots, p,$$

$$\frac{\mathbf{\Gamma}^{[n]}\mathbf{c}^{\curlywedge(k-1)}}{(k-1)!} + \frac{\mathbf{\Gamma}^{[n-1]}(\mathbf{c}-\mathbf{1}_s)^{\curlywedge(k-1)}}{(k-1)!} + \widetilde{\mathbf{u}}^{[n-2]}\frac{(-1)^k}{k!} = 0, \tag{5.69c}$$
$$k = 1, \ldots, q.$$

Example

The sequential peer two-step method `PEER4A` described in [80] is a special case of eq. (5.68) with order (2,3) conditions. Indeed in [80, Example 4], the "Nordsieck form" of the peer method is the underlying implicit and the "predictor" coefficients give the underlying explicit method.

5.7.2 Parallel Methods

The family of parallel linearly implicit GLMs allows independent and parallel computation of internal stages. In the case of linearly implicit GLMs choosing $\mathbf{A} = \mathbf{0}$ and $\mathbf{\Gamma} = \lambda\,\mathbf{I}_{s\times s}$ leads to a family of parallel methods . Similarly, the two-step linearly implicit Runge–Kutta methods become parallel when $\mathbf{A}^{[n]} = 0$ and $\mathbf{\Gamma}^{[n]} = \lambda\,\mathbf{I}_{s\times s}$.

5.7.3 Parallel Ensemble Methods

$$\mathbf{B} = \mathbf{C}_s\,\mathbf{F}_s\,(\mathbf{I}_{s\times s} - \lambda\,\mathbf{K}_s)\,\mathbf{C}_s^{-1}, \quad \mathbf{V} = \mathbf{I}_{s\times s}, \quad \mathbf{W} = \mathbf{C}_{s+1} - \lambda\,\mathbf{C}_{s+1}\,K_{s+1},$$

where the Toeplitz matrices

$$\mathbf{K}_n = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad \mathbf{F}_n = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{6} & \cdots & \frac{1}{n!} \\ & 1 & \frac{1}{2} & \cdots & \frac{1}{(n-1)!} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \frac{1}{2} \\ & & & & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}$$

and scaled Vandermone matrix

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

are used.

$$\mathbf{\Psi} = -\lambda\,\mathbf{C}_s\,\mathbf{K}_s\,(\mathbf{I}_{s\times s} - \lambda\,\mathbf{K}_s)^{-1}\,\mathbf{C}_s^{-1}$$

The stability essentially matches that of the one stage Rosenbrock W-method:

$$y_{n+1} = y_n + h\,(\mathbf{I}_{d\times d} - h\,\lambda\,\mathbf{L}_1)\,f(y_n).$$

5.7.4 Backward Differentiation Formulae

Backward differentiation formula (BDF) methods are a popular family of implicit linear multistep methods. The general form for a k-step BDF method is

$$y_n = \beta_0 h f(y_n) + \sum_{i=1}^{k} \alpha_i y_{n-i}. \tag{5.70}$$

These can be cast into the framework of GLMs and have a tableau of the form

$$\begin{array}{c|c|ccccc} 1 & \beta_0 & \alpha_1 & \dots & \alpha_{k-1} & \alpha_k \\ \hline & \beta_0 & \alpha_1 & \dots & \alpha_{k-1} & \alpha_k \\ & 0 & 1 & \dots & 0 & 0 \\ & \vdots & \vdots & \ddots & \vdots & \vdots \\ & 0 & 0 & \dots & 1 & 0 \end{array}.$$

Now we use this GLM to construct the linearly implicit GLM

$$K_1 = h\,f\left(\sum_{i=1}^{k} \widehat{\alpha}_i\,y_{n-i}\right) + \beta_0\,h\,\mathbf{L}\,K_1 + h\,\mathbf{L}\sum_{i=1}^{k}\psi_i\,y_{n-i}, \tag{5.71a}$$

$$y_n = \beta_0\,K_1 + \sum_{i=1}^{k}\alpha_i\,y_{n-i}, \tag{5.71b}$$

where $\alpha_i = \widehat{\alpha}_i + \psi_i$ to preserve eq. (5.70) as the underlying implicit method. The tableau for eq. (5.71a) is

$$\begin{array}{c|cc|ccccccc} 1 & 0 & \beta_0 & \widehat{\alpha}_1 & \dots & \widehat{\alpha}_{k-1} & \widehat{\alpha}_k & \psi_1 & \dots & \psi_k \\ \hline & \beta_0 & & \alpha_1 & \dots & \alpha_{k-1} & \alpha_k & & & \\ & 0 & & 1 & \dots & 0 & 0 & & & \\ & \vdots & & \vdots & \ddots & \vdots & \vdots & & & \\ & 0 & & 0 & \dots & 1 & 0 & & & \end{array} \tag{5.72}$$

We use this structure to derive order conditions for linearly implicit BDF methods with $p = q + 1 = k$. Solving eq. (5.20) leads to the method coefficients

$$\beta_0 = \left(\sum_{j=0}^{k} \frac{1}{j} \right)^{-1}, \quad \widehat{\alpha}_i = (-1)^{i+1} \binom{k}{i}, \quad \psi_i = \widehat{\alpha}_i \left(\frac{\beta_0}{i} - 1 \right), \quad i = 1, \ldots, k. \tag{5.73}$$

By applying the transformation eq. (5.15) and some algebraic simplifications, we arrive at the following compact form of linearly-implicit BDF methods:

$$y_n = (\mathbf{I}_{s\times s} - \beta_0\, h\, \mathbf{L})^{-1} \left(\beta_0\, h\, f\Big(\sum_{i=1}^{k} \widehat{\alpha}_i\, y_{n-i} \Big) + \sum_{i=1}^{k} \psi_i\, y_{n-i} \right) + \sum_{i=1}^{k} \widehat{\alpha}_i\, y_{n-i}. \tag{5.74}$$

5.8 Numerical experiments

In this section we provide numerical experiments using various linearly implicit GLMs derived from the families introduced in section 5.7 applied to a number of test problems. Test problems are chosen to highlight features of this class of linearly implicit methods. We present convergence results on a DAE problem, a PDE with time-dependent boundary conditions, Euler equations of gas dynamics with approximate Jacobian and the Van der Pol oscillator for a stiff system.

5.8.1 The transistor-amplifier test problem

The transistor-amplifier test problem is an index-1 DAE describing various voltages across a two-transistor amplifier circuit with a singular mass matrix. The derivation of the equations is documented in [115] and the Matlab implementation of this problem can be found in `ODE Test Problems` suit [90]. We perform convergence experiments for a range of time-steps using the exact Jacbobian over a timespan $t = [0, 0.2]$ and record the l_2 error between the final solution and a reference solution computed with tight tolerances. Figure 5.1 shows normalized error versus number of time steps for linearly implicit GLMsup to order 5. We observe nominal order of convergence for the methods.

5.8.2 Convergence study with the Hundsdorfer problem problem

In the next set of experiments we use the Hundsdorfer PDE with time-dependent boundary conditions [56, Sec. 6.3]. The PDE is described as

$$\begin{aligned} u_t + u_x &= -k_1 u + k_2 v, \\ v_t &= k_1 u - k_2 v + 1, \end{aligned} \tag{5.75}$$

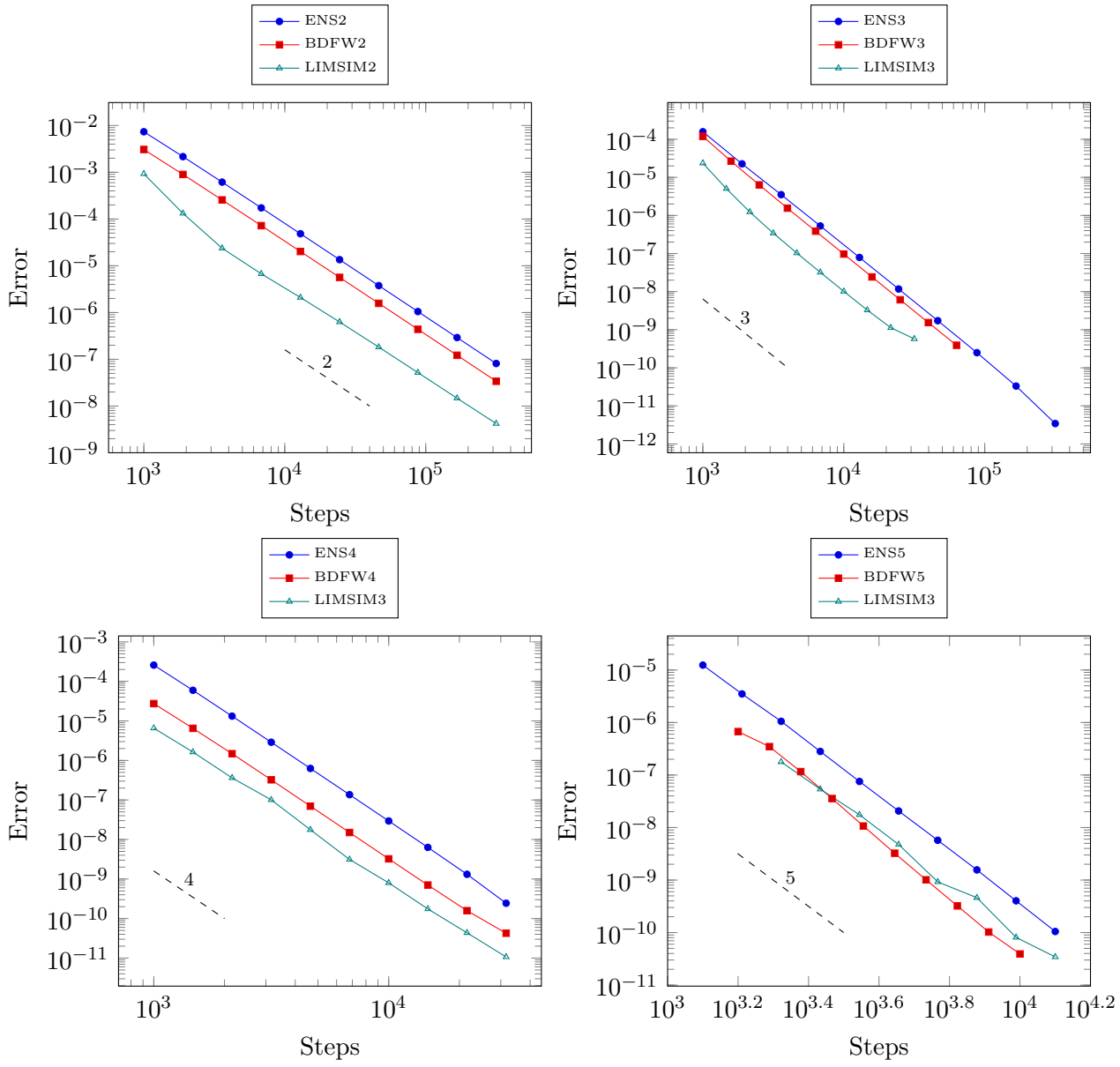
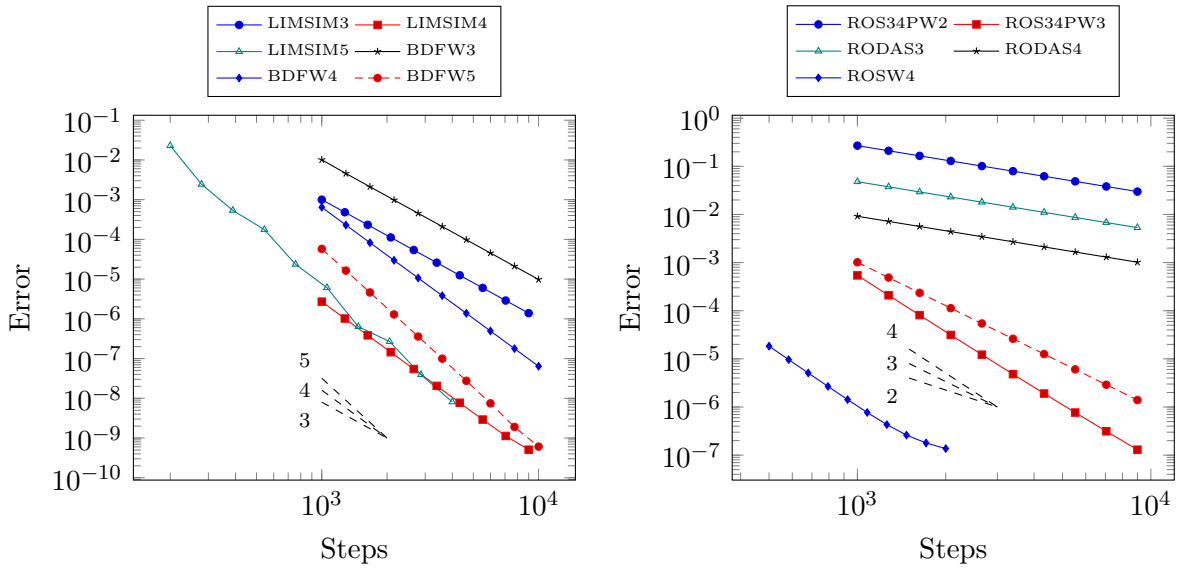


Figure 5.1: Convergence study with the Transistor-Amplifier test problem



(a) linearly implicit GLMs (b) Rosernbrock and ROW methods

FIGURE 5.2: Convergence plots for the Hundsdorfer problem

where $x, t \in [0, 1]$ and the parameters are chosen as $k_1 = 10^6$ and $k_2 = 2k_1$. Fourth-order central finite differences are used for spatial discretization and the time-dependent Dirichlet boundary condition is $u(0, t) = 1 - \sin(12t)^4$. These choices minimize the errors due to spatial-discretization and allow better estimates of the time-discretization error. This linear advection-reaction problem is known to cause order reduction in Runge-Kutta methods due to the influences of the time-dependent boundary conditions [57, Chapter 2].

We compare the convergence rates of linearly implicit GLMs against a number of Rosenbrock and Rosenbrock-W methods from the literature: RODAS3 and RODAS4 are Rosenbrock methods of orders 3 and 4 from [98]. ROS34PW1b and ROS34PW2 are third order ROW methods, and ROS34PW4 is a fourth order ROW method from [87]. ROSW4 is a fourth order ROW method from [84, 6S4O(C)-W]. Readers interested in verifying the implementation and nominal orders of these methods may consult the Brusselator test problem results in the supplementary materials section in section B.5.

In all our experiments , the exact Jacobian is used, but the time derivative of the right hand side is ignored as suggested in [87, Section 1]. We note that among Rosenbrock and ROW methods tested on this problem ROS34PW1b, ROS34PW3, and ROSW4 retain their nominal convergence rates as shown in fig. 5.2a while other linearly-implicit Runge–Kutta methods encounter order reduction. On the other hand, as fig. 5.2b indicates, the linearly implicit GLMs perform close to their theoretical order of convergence. The only exception here is the LIMSIM5 method which due to the limited precision in the starting procedure shows fluctuations in its order.

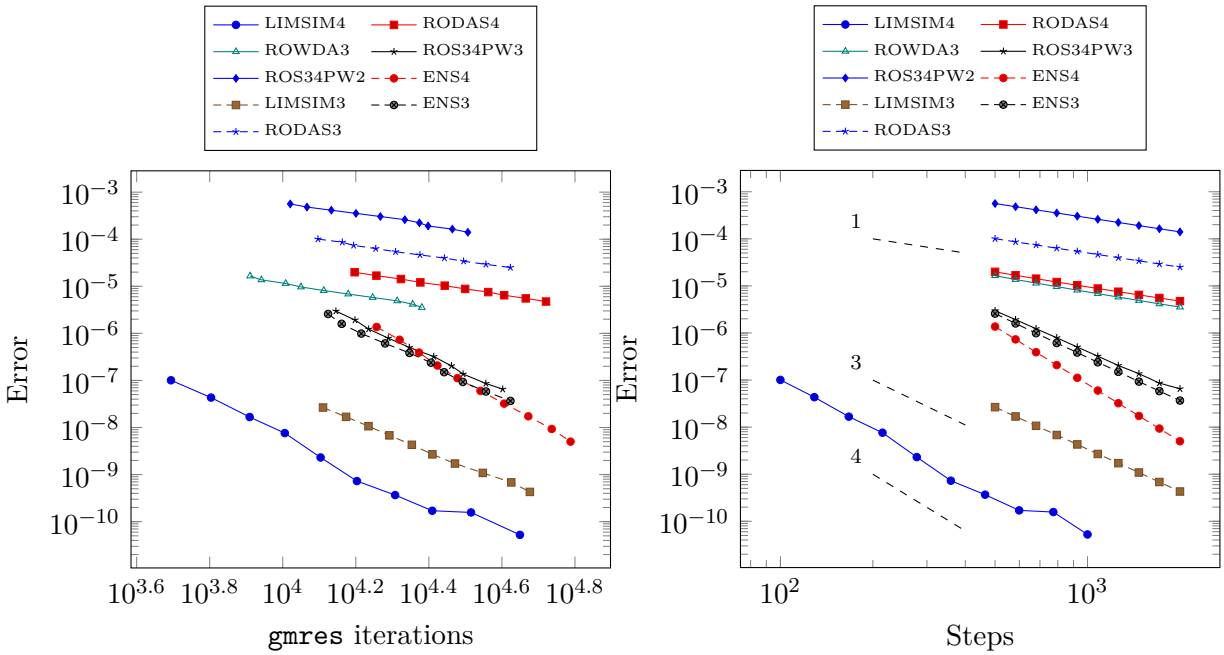


FIGURE 5.3: Performance evaluations for the Isentropic vortex test problem using approximate Jacobian-vector products

5.8.3 Euler equations with approximate Jacobian

In this set of experiments we use the 2D Euler equations for compressible gas dynamics summarized in eq. (5.76) and discretized using Discrete Galerkin (DG) finite element method with degree 5 nodal polynomial basis. Here, we are interested in studying the computational efficiency of our methods compared to Rosenbrock and ROW methods. We have used the `Matlab` implementation of the Isentropic Vortex test provided in [54, Section 6.6]. The system is integrated over timespan $t = [0, 10]$ with parameter $\gamma = 1.4$ and $E = \frac{p}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2)$.

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0, \\
&\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2 + p}{\partial x} + \frac{\partial \rho u v}{\partial y} = 0, \\
&\frac{\partial \rho v}{\partial t} + \frac{\partial \rho u v}{\partial x} + \frac{\partial \rho v^2 + p}{\partial y} = 0, \\
&\frac{\partial E}{\partial t} + \frac{\partial u(E + p)}{\partial x} + \frac{\partial v(E + p)}{\partial y} = 0.
\end{aligned} \tag{5.76}$$

In many practical applications the exact Jacobian of a complex discretization can not be

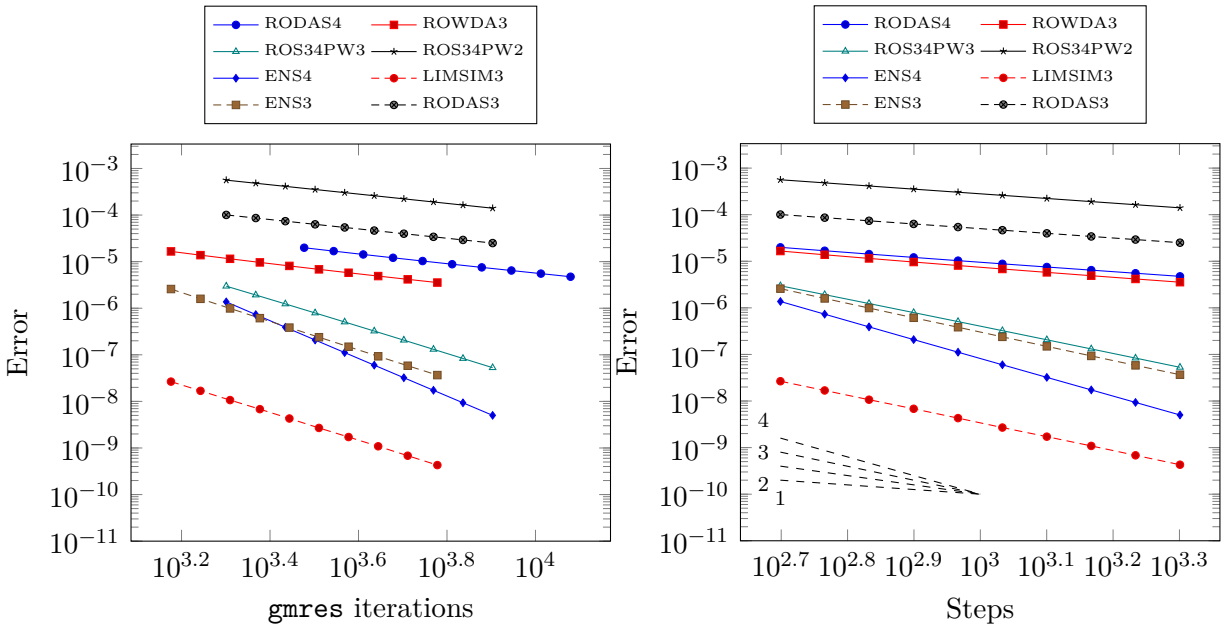
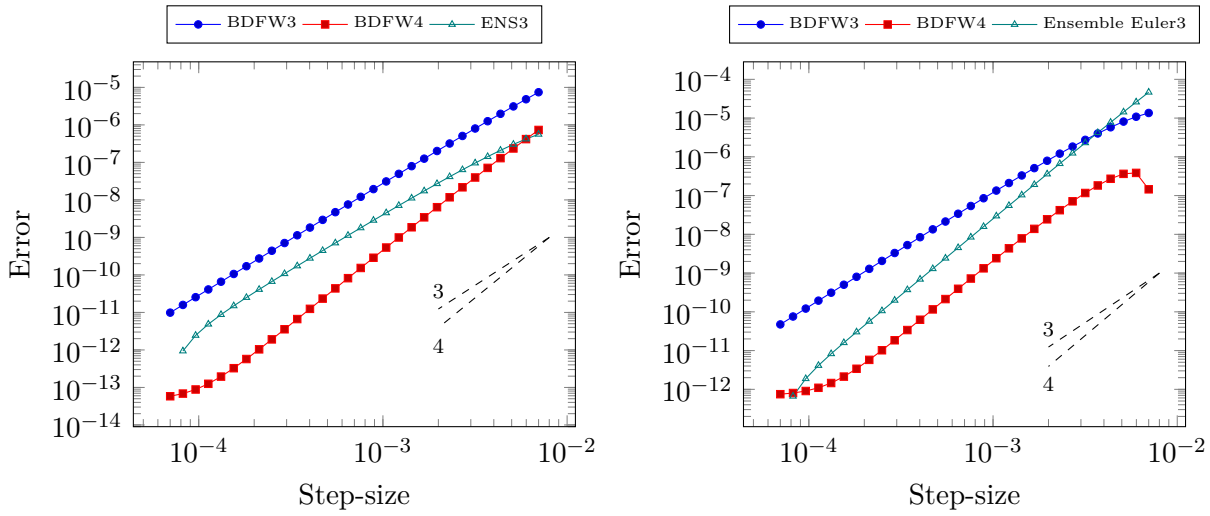


FIGURE 5.4: Performance evaluations for the Isentropic vortex test problem using direct solvers

computed or factorized for use in linear system solves. We used the `numjac` function to create an inexact sparse Jacobian at the beginning of the integration and employ it in our computations. Furthermore, the stage equations are solved with the iterative `gmres` solver. As a result , the final error is affected both by the local truncation error of the method as well as the convergence of the solver (i.e. the iterative solver may not converge within specified tolerance for some methods). We used `Tol=1E-6` for the `gmres` solver. The l_2 norm errors are computed at the final time for the density (ρ) component of the system.

Figure 5.3a shows total number of `gmres` iterations versus final error for a number of linearly-implicit methods. We observe the performance advantage of linearly implicit GLMs with the LIMSIM4 method being the most accurate for a fixed computational cost. A number of reasons can account for this observation. As the convergence diagram fig. 5.3b indicates, linearly implicit GLMs keep their accuracy where other methods show order reduction, reducing the computational efficiency of the method. We also note that the number of internal stages increases dramatically with the order in Runge–Kutta methods, whereas for linearly implicit GLMs this number increases linearly.

Figure 5.4 shows the same experiment, however, linear systems are solved using direct methods. Here, we report the number of right-hand-side function evaluations versus the final error. Again, fig. 5.4a shows many of the linearly implicit GLMs outperform linearly-implicit Runge-Kutta ones.



(a) $y-$component (b) $z-$component

FIGURE 5.5: Convergence plots for linearly implicit GLMs applied to the Van der Pol oscillator

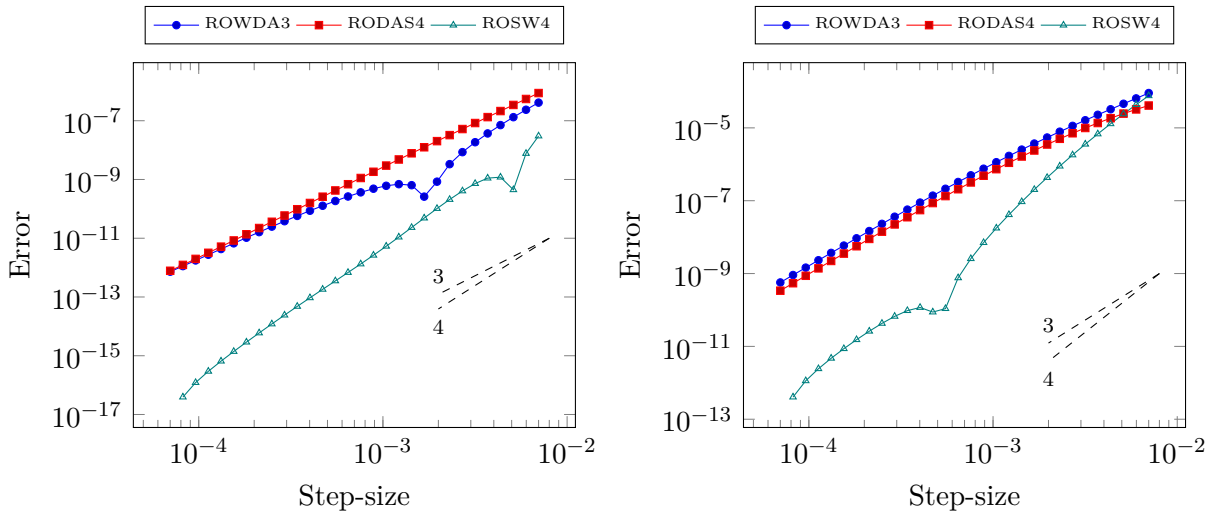
5.8.4 Van der Pol oscillator

The Van der Pol problem is commonly used in the literature [45, 100, 108] for studying the stiff behavior of time-stepping methods. Following the analysis in section 5.6.2 we test the convergence of our methods on the van de Pol eq. (5.77):

$$\begin{aligned} y' &= z, \\ \varepsilon z' &= (1 - y^2)z - y. \end{aligned} \tag{5.77}$$

For the complete problem definition and a highly accurate initial condition refer to [45, Section 5]. Here, we provide convergence rates of the $y-$ and $z-$ components of the system as we integrate eq. (5.77) with $\varepsilon = 10^{-3}$ over timespan $t = [0, 0.4]$.

Figure 5.5 indicates full convergence of both stiff and non-stiff variables when integrated using linearly implicit GLMs. For the $z-$variable in fig. 5.5b, since the Jacobian is $\mathcal{O}(h)$ accurate, we recover from the internal stage order q to the method order p. Figure 5.6 shows the results for linearly-implicit Runge–Kutta methods. We see the "Hump" phenomenon, a local increase of the order due to cancellation of the leading error term, as explained in [51, p. 113]. Therefore, we will consider the slope of the convergence plots away from these irregularities. Although the non-stiff component shows full order in fig. 5.6a, ROWDA3 and RODAS4 show order reduction for the stiff component when the step sizes are large in fig. 5.6b.



(a) $y-$component (b) $z-$component

FIGURE 5.6: Convergence plots for linearly implicit GLMs applied to the Van der Pol oscillator

Chapter 6

Conclusions

This manuscript started with a study of the performance of several types of high-order matrix-free time stepping methods when applied to solve unsteady flow problems in chapter 2. The methods considered are explicit Runge-Kutta, diagonally implicit Runge-Kutta and Rosenbrock schemes paired with iterative linear algebra solvers, and Rosenbrock-Krylov schemes. All implementations of implicit methods are matrix-free where the necessary Jacobian-vector products are approximated by finite differences; this is a typical setting for solving large-scale applications.

We observed that favorable properties of explicit methods include their easier implementation overhead and the low computational cost per step. As expected, the numerical experiments show that the overall performance of explicit methods deteriorates quickly in the presence of mild stiffness, and for such problems they are not competitive with implicit methods.

Traditional implicit methods such as SDIRK and Rosenbrock can take large steps on stiff problems, as expected. However, their overall performance depends on how well the underlying linear systems are solved at each step. We observed that inaccurate linear solutions lead to loss of convergence, and that in absence of well-tuned linear algebra preconditioners the computational costs of matrix-free implicit methods are quite large. Based on these results we identified the need for methods that can provide a better balance between implicit integration and computational cost. We also considered addressing the order reduction phenomenon as a goal of this thesis.

Chapter 3 was dedicated to developing multirate methods suitable for multiphysics problems that operate on inherently diverse time-scales. This chapter devises a coherent design strategy for high order MrGARK, and constructs several particular schemes of explicit-explicit, explicit-implicit and implicit-explicit types up to order four. Adopting a two-way split system for simplicity, we identify coupling structures for the fast and slow components that strike a balance between stability and computational efficiency of MrGARK methods. Furthermore, in the process of designing schemes a variety of optimizations such as local truncation error minimization, FSAL, and stiff accuracy of base and overall implicit methods are explored. When possible, the methods are endowed with telescopic properties to facilitate their application to multi-scale, multi-domain problems. A novel concept of H-M adaptivity, based on multiple error estimators and the property of natural adaptivity, is presented and tested.

The numerical experiments in this chapter demonstrate several applications of these methods

with finite element and finite difference discretized PDEs, where different time-scales in system variables (component partitioning), or in various physical processes of the problem (additive partitioning) are treated with different time steps.

In chapter 4, General Linear methods becomes the focus of our attention due to their significant advantages over Runge–Kutta methods specially against order reduction. This chapter introduces the new family of ADI-GLM schemes that perform alternating directions implicit integration. Each stage of a ADI-GLM scheme is implicit in a single directional component of the PDE system, and is explicitly coupled to the other components. This ensures low computational cost. The ADI character of the method stems from the fact that consecutive stages are implicit in different partitions, thereby "alternating directions." Order conditions and stability of these methods are investigated theoretically. The ADI-GLM structure allows for high stage order approximations, and this property alleviates the order reduction observed with other families of schemes.

Using the new ADI-GLM theory we construct practical linearly implicit GLMs of orders two, three, and four. Our design emphasizes stability when applied to parabolic systems where each component has a Jacobian with real negative eigenvalues. The stability analysis and plots show the new schemes are well-suited for these problems. Numerical experiments show that the new methods retain their high order of accuracy when applied to parabolic equations with time-dependent Dirichlet boundary conditions where other ADI methods suffer from order reduction. We show that the proposed framework is extensive and flexible, both encompassing existing linearly implicit methods as well as examples of new ones easily built from diagonally implicit methods.

Finally, chapter 5 presented the new class of linearly implicit GLMs for linearly implicit integration of systems with high orders both in external and internal stages. The order conditions show a close relation between coefficients of linearly implicit GLMs and IMEX-GLM pairs that can be leveraged to design methods with preferable stability and error properties. We study the convergence of these methods for Prothero-Robinson problem, index-1 DAEs and singularly perturbed problems. We also provide closed-from solutions for a number of families of methods that can be used to create schemes of any desired order. Numerical experiments highlight a number of advantages for linearly implicit GLMs when competing methods encounter order reduction under different scenarios: approximate Jacobians, stiffness, or asymptotic residuals. We also provide evidence of the computational efficiency of linearly implicit GLMs compared to Runge–Kutta ones specially at high orders.

Bibliography

[1] Georgios Akrivis and Michel Crouzeix. Linearly implicit methods for nonlinear parabolic equations. *Math. Comp*, 73:613–635, 2004.

[2] R. Alexander. Diagonally implicit Runge–Kutta methods for stiff ODE's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[3] 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.

[4] I. Alonso-Mallo and B. Cano. Spectral/rosenbrock discretizations without order reduction for linear parabolic problems. *Applied Numerical Mathematics*, 41(2):247 – 268, 2002. ISSN 0168-9274. doi: 10.1016/S0168-9274(01)00101-5. URL http://www.sciencedirect.com/science/article/pii/S0168927401001015.

[5] J.F. Andrus. Numerical solution for ordinary differential equations separated into subsystems. *SIAM Journal on Numerical Analysis*, 16(4):605–611, 1979.

[6] J.F. Andrus. Stability of a multirate method for numerical integration of ODEs. *Computers Math. Applic*, 25:3–14, 1993.

[7] A. Bartel and M. Günther. A multirate W-method for electrical networks in state-space formulation. *Journal of Computational Applied Mathematics*, 147(2):411–425, 2002. ISSN 0377-0427. doi: 10.1016/S0377-0427(02)00476-4.

[8] Francesco Bassi, L Botti, Alessandro Colombo, A Ghidoni, and F Massa. Linearly implicit Rosenbrock-type Runge–Kutta schemes applied to the discontinuous Galerkin solution of compressible and incompressible unsteady flows. *Computers & Fluids*, 118: 305–320, 2015.

[9] Steffen Beck, Rüdiger Weiner, Helmut Podhaisky, and Bernhard A. Schmitt. Implicit peer methods for large stiff ode systems. *Journal of Applied Mathematics and Computing*, 38(1):389–406, Feb 2012. ISSN 1865-2085. doi: 10.1007/s12190-011-0485-0.

[10] Swagata Bhaumik, Soumyo Sengupta, and Aditi Sengupta. Wave properties of fourth-order fully implicit Runge–Kutta time integration schemes. *Computers & Fluids*, 81: 110–121, 2013.

[11] Michał Braś, Angelamaria Cardone, Zdzisław Jackiewicz, and Bruno Welfert. Order reduction phenomenon for general linear methods. *Applied Numerical Mathematics*, 119:94 – 114, 2017. ISSN 0168-9274. doi: 10.1016/j.apnum.2017.04.001.

[12] Michał Braś, Angelamaria Cardone, Zdzisław Jackiewicz, and Bruno Welfert. Order reduction phenomenon for general linear methods. *Applied Numerical Mathematics*, 119:94–114, Sep 2017. ISSN 0168-9274. doi: 10.1016/j.apnum.2017.04.001.

[13] S. Bremicker-Trübelhorn and S. Ortleb. On multirate GARK schemes with adaptive micro-step sizes for fluid-structure interaction: order conditions and preservation of the geometric conservation law. *Aerospace*, 4(8), 2017. doi: 10.3390/aerospace4010008.

[14] B. Bujanda and J.C. Jorge. Stability results for fractional-step discretizations of time dependent coefficient evolutionary problems. *Applied Numerical Mathematics*, 38:69–86, 2001.

[15] B. Bujanda and J.C. Jorge. Fractional-step Runge–Kutta methods for time dependent coefficient parabolic problems. *Applied Numerical Mathematics*, 45:99–122, 2003.

[16] J.C. Butcher. Diagonally-implicit multi-stage integration methods. *Appl. Numer. Math.*, 11(5):347–363, 1993. ISSN 0168-9274. doi: http://dx.doi.org/10.1016/0168-9274(93)90059-Z.

[17] J.C. Butcher. A history of Runge–Kutta methods. *Applied Numerical Mathematics*, 20(247–260), 1996.

[18] J.C. Butcher. General linear methods for stiff differential equations. *BIT*, 41(2): 240–264, 2001. doi: 10.1023/A:1021986222073.

[19] J.C. Butcher and Z. Jackiewicz. Diagonally implicit general linear methods for ordinary differential equations. *BIT*, 33(3):452–472, 1993. doi: 10.1007/BF01990528.

[20] J.C. Butcher and Z. Jackiewicz. Construction of diagonally implicit general linear methods of type 1 and 2 for ordinary differential equations. *Applied Numerical Mathematics*, 21(4):385–415, 1996. doi: 10.1016/S0168-9274(96)00043-8.

[21] JC Butcher and Zdzislaw Jackiewicz. Diagonally implicit general linear methods for ordinary differential equations. *BIT Numerical Mathematics*, 33(3):452–472, 1993.

[22] J.C. Butcher and W.M. Wright. The construction of practical general linear methods. *BIT*, 43(4):695–721, 2003. doi: 10.1023/B:BITN.0000009952.71388.23.

[23] G Califano, G Izzo, and Z Jackiewicz. Starting procedures for general linear methods. *Applied Numerical Mathematics*, 120:165–175, 2017.

[24] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang. Extrapolation-based implicit-explicit general linear methods. *Numerical Algorithms*, 65(3):377–399, 2014. doi: 10.1007/s11075-013-9759-y. URL http://dx.doi.org/10.1007/s11075-013-9759-y.

[25] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang. Construction of highly-stable implicit-explicit general linear methods. In M. de Leon, W. Feng, Z. Feng, J.L. Gomez, X. Lu, J.M. Martell, J. Parcet, D. Peralta-Salas, and W. Ruan, editors, *AIMS proceedings*, number 0133-0189_2015_special_185 in Dynamical Systems, Differential Equations, and Applications, pages 185–194, Madrid, Spain, 2015. doi: 10.3934/proc.2015.0185.

[26] E.M. Constantinescu and A. Sandu. Multirate timestepping methods for hyperbolic conservation laws. *Journal on Scientific Computing*, 33(3):239–278, 2007. doi: 10.1007/s10915-007-9151-y. URL http://dx.doi.org/10.1007/s10915-007-9151-y.

[27] E.M. Constantinescu and A. Sandu. On extrapolated multirate methods. In Hans-Georg Bock, Frank Hoog, Avner Friedman, Arvind Gupta, Helmut Neunzert, William R. Pulleyblank, Torgeir Rusten, Fadil Santosa, Anna-Karin Tornberg, Vincenzo Capasso, Robert Mattheij, Helmut Neunzert, Otmar Scherzer, Alistair D. Fitt, John Norbury, Hilary Ockendon, and Eddie Wilson, editors, *Progress in Industrial Mathematics at ECMI 2008*, volume 15 of *Mathematics in Industry*, pages 341–347. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-12110-4_52. URL http://dx.doi.org/10.1007/978-3-642-12110-4_52.

[28] E.M. Constantinescu and A. 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. URL http://dx.doi.org/10.1007/s10915-012-9662-z.

[29] Andrea Crivellini and Francesco Bassi. An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. *Computers & Fluids*, 50(1):81–93, 2011.

[30] Joseph M Derlaga, Tyrone S Phillips, Christopher J Roy, and V Tech. SENSEI computational fluid dynamics code: a case study in modern fortran software development. In *21st AIAA Computational Fluid Dynamics Conference*, page 2450, 2013.

[31] J. R. Dormand, M. A. Lockyer, N. E. McGorrigan, and P. J. Prince. Global error estimation with Runge-Kutta triples. *Computers & Mathematics with Applications*, 18 (9):835–846, 1989.

[32] J. Douglas. On the numerical integration of $u_{x,x} + u_{y,y} = u_t$ by implicit methods. *SIAM*, 3:42–65, 1955.

[33] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82:421–439, 1956.

[34] Howard C Elman, David J Silvester, and Andrew J Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation, 2014.

[35] B. Engquist and R. Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Mathematics of Computation*, 74:1707–1742, 2005.

[36] C. Engstler and C. Lubich. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58(2):173–185, Jun 1997. ISSN 1436-5057. doi: 10.1007/BF02684438. URL https://doi.org/10.1007/BF02684438.

[37] Wayne H Enright, Desmond J Higham, Brynjulf Owren, and Philip W Sharp. A survey of the explicit Runge-Kutta method, 1994.

[38] Erwin Fehlberg. Low–order classical Runge–Kutta formulas with stepsize control and their application to some heat transfer problems. *NASA Technical Report, NASA-TR-R-315*, 1969.

[39] C.W. Gear and D.R. Wells. Multirate linear multistep methods. *BIT*, 24:484–502, 1984.

[40] R. Glandon, P. Tranquilli, and A. Sandu. Biorthogonal Rosenbrock-Krylov time discretization methods. *Applied Numerical Mathematics*, 150:233–251, 2020. doi: 10.1016/j.apnum.2019.09.003. URL https://doi.org/10.1016/j.apnum.2019.09.003.

[41] S.R. Glandon, M. Narayanamurthi, and A. Sandu. Linearly implicit multistep methods for time integration. *SIAM Journal of Scientific Computing*, Accepted(M133748), 2021. URL https://arxiv.org/abs/2011.10685.

[42] Severiano González-Pinto, D Hernández-Abreu, and Severiano Pérez-Rodríguez. Rosenbrock-type methods with inexact AMF for the time integration of advection–diffusion–reaction PDEs. *Journal of Computational and Applied Mathematics*, 262:304–321, 2014. ISSN 0377-0427. doi: 10.1016/j.cam.2013.10.050.

[43] Severiano González-Pinto, D Hernández-Abreu, and S Pérez-Rodríguez. AMF–Runge–Kutta formulas and error estimates for the time integration of advection diffusion reaction PDEs. *Journal of Computational and Applied Mathematics*, 289:3–21, 2015. ISSN 0377-0427. doi: 10.1016/j.cam.2015.03.048.

[44] Severiano González-Pinto, E Hairer, D Hernández-Abreu, and S Pérez-Rodríguez. AMF–type W–methods for parabolic problems with mixed derivatives. *SIAM Journal on Scientific Computing*, 40(5):A2905–A2929, 2018. doi: 10.1137/17M1163050.

[45] S. González-Pinto and D. Hernández-Abreu. Global error estimates for a uniparametric family of stiffly accurate runge-kutta collocation methods on singularly perturbed problems. *BIT Numerical Mathematics*, 51(1):155–175, Mar 2011. ISSN 1572-9125. doi: 10.1007/s10543-010-0304-2.

[46] M. Günther and M. Hoschek. ROW methods adapted to electric circuit simulation packages. In *ICCAM '96: Proceedings of the Seventh International Congress on Computational and Applied Mathematics*, pages 159–170. Elsevier Science Publishers B. V., 1997. doi: 10.1016/S0377-0427(97)00043-5.

[47] M. Günther and A. Sandu. Multirate generalized additive Runge-Kutta methods. *Numerische Mathematik*, 133(3):497–524, 2016. doi: 10.1007/s00211-015-0756-z. URL http://dx.doi.org/10.1007/s00211-015-0756-z.

[48] M. Günther, A. Kværnø, and P. Rentrop. Multirate partitioned Runge-Kutta methods. *BIT Numerical Mathematics*, 41(3):504–514, Jun 2001. ISSN 1572-9125. doi: 10.1023/A:1021967112503.

[49] M. Günther, C. Hachtel, and A. Sandu. Multirate GARK schemes for multiphysics problems. In *10th International Conference on Scientific Computing in Electrical Engineering*, 2014. doi: 10.1007/978-3-319-30399-4_12. URL http://dx.doi.org/10.1007/978-3-319-30399-4_12.

[50] Michael Günther and Peter Rentrop. *Partitioning and multirate strategies in latent electric circuits*, pages 33–60. Birkhäuser Basel, Basel, 1994. ISBN 978-3-0348-8528-7. doi: 10.1007/978-3-0348-8528-7_3.

[51] E. Hairer and G. 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.

[52] E. Hairer, S.P. Norsett, and G. Wanner. *Solving ordinary differential equations I: Non-stiff problems*. Number 8 in Springer Series in Computational Mathematics. Springer-Verlag Berlin Heidelberg, 1993.

[53] Wolfram Heineken and Gerald Warnecke. Partitioning methods for reaction–diffusion problems. *Applied Numerical Mathematics*, 56(7):981–1000, 2006.

[54] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer-Verlag, New York, NY, USA, 2008. ISBN 978-0-387-72065-4. doi: 10.1007/978-0-387-72067-8.

[55] Rong F. Huang and Chih L. Lin. Vortex shedding and shear-layer instability of wing at low-Reynolds numbers. *AIAA Journal*, 33(8):1398–1403, 1995.

[56] 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, Aug 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2007.03.003.

[57] Willem Hundsdorfer and Jan G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, Berlin, Germany, 2003. ISBN 978-3-540-03440-7. doi: 10.1007/978-3-662-09017-6.

[58] G. Izzo and Z. Jackiewicz. Transformed implicit-explicit DIMSIMs with strong stability preserving explicit part. *Numerical Algorithms*, 81(4):1343–1359, Aug 2019. ISSN 1572-9265. doi: 10.1007/s11075-018-0647-3.

[59] Z. Jackiewicz. *General Linear Methods for Ordinary Differential Equations*. Wiley, Hoboken, New Jersey, 2009. ISBN 978-0-470-40855.

[60] Z. Jackiewicz and S. Tracogna. A general class of two-step runge–kutta methods for ordinary differential equations. *SIAM Journal on Numerical Analysis*, 32(5):1390–1427, Oct 1995. ISSN 0036-1429. doi: 10.1137/0732064.

[61] Zdzislaw Jackiewicz. *General linear methods for ordinary differential equations*. John Wiley & Sons, 2009.

[62] Philip CE Jorgenson and Rodrick C Chima. Explicit Runge-Kutta method for unsteady rotor/stator interaction. *AIAA journal*, 27(6):743–749, 1989.

[63] T. Kato and T. Kataoka. Circuit analysis by a new multirate method. *Electrical Engineering in Japan*, 126(4):55–62, 1999.

[64] 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.

[65] Christopher A. Kennedy, Mark H. Carpenter, and R. Michael Lewis. Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations. *Applied Numerical Mathematics*, 35(3):177–219, November 2000. ISSN 0168-9274. doi: 10.1016/S0168-9274(99)00141-5. URL http://www.sciencedirect.com/science/article/pii/S0168927499001415.

[66] Christopher A. Kennedy, Mark H. Carpenter, and R. Michael Lewis. Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations. *Applied Numerical Mathematics*, 35(3):177–219, November 2000. ISSN 0168-9274. doi: 10.1016/S0168-9274(99)00141-5. URL http://www.sciencedirect.com/science/article/pii/S0168927499001415.

[67] Christiaan M Klaij, Jaap JW van der Vegt, and Harmen van der Ven. Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations. *Journal of Computational Physics*, 217(2):589–611, 2006.

[68] D.A. Knoll and D.E. Keyes. Jacobian–free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193:357 – 397, 2004. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2003.08.010. URL http://www.sciencedirect.com/science/article/pii/S0021999103004340.

[69] A. Kværnø. Stability of multirate Runge-Kutta schemes. *International Journal of Differential Equations and Applications*, 1(1):97–105, 2000.

[70] A. Kværnø and P. Rentrop. Low order multirate Runge-Kutta methods in electric circuit simulation, 1999. URL citeseer.ist.psu.edu/629589.html.

[71] J. Lang and J. Verwer. ROS3P—An Accurate Third-Order Rosenbrock Solver Designed for Parabolic Problems. *BIT Numerical Mathematics*, 41(4):731–738, Sep 2001. ISSN 1572-9125. doi: 10.1023/A:1021900219772.

[72] Kyoung J Lee, WD McCormick, Qi Ouyang, and Harry L Swinney. Pattern formation by interacting chemical fronts. *Science*, 261(5118):192–194, 1993.

[73] A. Logg. Multi-adaptive Galerkin methods for ODEs I. *SIAM Journal on Scientific Computing*, 24:1879–1902, 2003.

[74] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, pages 41–58, 1996.

[75] C. Lubich and A. Ostermann. Linearly implicit time discretization of nonlinear parabolic equations. *IMA Journal on Numerical Mathematics*, 15:555–583, 1995.

[76] Syvert P. Nørsett and Arne Wolfbrandt. Order conditions for rosenbrock type methods. *Numerische Mathematik*, 32(1):1–15, Mar 1979. ISSN 0945-3245. doi: 10.1007/BF01397646.

[77] A Ostermann and M Roche. Runge–Kutta methods for partial differential equations and fractional orders of convergence. *Mathematics of computation*, 59(200):403–420, 1992.

[78] A. Ostermann and M. Roche. Rosenbrock methods for partial differential equations and fractional orders of convergence. *SIAM Journal on Numerical Analysis*, 30(4): 1084–109, 1993.

[79] D.W. Peaceman and H.H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal of Society for Indistrial and Applied Mathematics*, 3: 28–42, 1955.

[80] H. Podhaisky, R. Weiner, and B. A. Schmitt. Linearly-implicit two-step methods and their implementation in nordsieck form. *Applied Numerical Mathematics*, 56(3): 374–387, Mar 2006. ISSN 0168-9274. doi: 10.1016/j.apnum.2005.04.024.

[81] L. Portero, J.C. Jorge, and B. Bujanda. Avoiding order reduction of fractional step Runge–Kutta discretizations for linear time dependent coefficient parabolic problems. *Applied Numerical Mathematics*, 48(3):409 – 424, 2004. ISSN 0168-9274. doi: 10.1016/j.apnum.2003.11.006.

[82] 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. ISSN 00255718.

[83] Ning Qin, David K. Ludlow, and Scott T. Shaw. A matrix–free preconditioned Newton/GMRES method for unsteady Navier–Stokes solutions. *International Journal for Numerical Methods in Fluids*, 33(2):223–248, May 2000. ISSN 1097-0363. doi: 10.1002/(SICI)1097-0363(20000530)33:2<223::AID-FLD10>3.0.CO;2-V. URL http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-0363(20000530)33:2<223::AID-FLD10>3.0.CO;2-V/abstract.

[84] Arunasalam Rahunanthan and Dan Stanescu. High-order w-methods. *Journal of Computational and Applied Mathematics*, 233(8):1798–1811, Feb 2010. ISSN 03770427. doi: 10.1016/j.cam.2009.09.017.

[85] Manoj K Rajpoot, Tapan K Sengupta, and Pravir K Dutt. Optimal time advancing dispersion relation preserving schemes. *Journal of Computational Physics*, 229(10): 3623–3651, 2010.

[86] Anthony Ralston. Runge–Kutta methods with minimum error bounds. *Mathematics of Computation*, 16(80):431–437, 1962.

[87] Joachim Rang and L Angermann. New Rosenbrock W-methods of order 3 for partial differential algebraic equations of index 1. *BIT Numerical Mathematics*, 45(4):761–787, 2005.

[88] J.R. Rice. Split Runge-Kutta methods for simultaneous equations. *Journal of Research of the National Institute of Standards and Technology*, 60(B), 1960.

[89] S. Roberts, A. Sarshar, and A. Sandu. Parallel implicit-explicit general linear methods. *Communications on Applied Mathematics and Computation*, in print, 2020. doi: 10.1007/s42967-020-00083-5. URL https://doi.org/10.1007/s42967-020-00083-5.

[90] Steven Roberts, Andrey A Popov, and Adrian Sandu. ODE test problems: a Matlab suite of initial value problems. Technical Report CSL-TR-19-1, Computational Science Laboratory, Virginia Tech, 2019. URL https://github.com/ComputationalScienceLaboratory/ODE-Test-Problems.

[91] H. H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *The Computer Journal*, 5(4):329–330, 1963. doi: 10.1093/comjnl/5.4.329. URL http://comjnl.oxfordjournals.org/content/5/4/329.abstract.

[92] Yousef Saad. *Iterative methods for sparse linear systems*. Siam, 2003.

[93] A. Sandu. Rosenbrock methods with an explicit first stage. *International Journal of Computer Mathematics*, 93(6):995–1010, 2016. doi: 10.1080/00207160.2015.1012837. URL http://dx.doi.org/10.1080/00207160.2015.1012837.

[94] A. Sandu. Convergence results for implicit-explicit general linear methods. *Applied Numerical Mathematics*, 156:242–264, 2020. doi: 10.1016/j.apnum.2020.04.005. URL https://doi.org/10.1016/j.apnum.2020.04.005.

[95] A. Sandu and E.M. Constantinescu. Multirate time discretizations for hyperbolic partial differential equations. In *International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2009)*, volume 1168-1 of *American Institute of Physics (AIP) Conference Proceedings*, pages 1411–1414, 2009. doi: 10.1063/1.3241354. URL http://dx.doi.org/10.1063/1.3241354.

[96] A. Sandu and E.M. Constantinescu. Multirate Adams methods for hyperbolic equations. *Journal of Scientific Computing*, 38(2):229–249, 2009. doi: doi:10.1007/s10915-008-9235-3. URL http://dx.doi.org/10.1007/s10915-008-9235-3.

[97] A. Sandu and M. Günther. A generalized-structure approach to additive Runge-Kutta methods. *SIAM Journal on Numerical Analysis*, 53(1):17–42, 2015. doi: 10.1137/130943224. URL http://dx.doi.org/10.1137/130943224.

[98] A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee, G.R. Carmichael, and F.A. Potra. Benchmarking stiff ODE solvers for atmospheric chemistry problems. II: Rosenbrock solvers. *Atmospheric Environment*, 31:3459–3472, 1997. doi: 10.1016/s1352-2310(97)83212-8. URL http://dx.doi.org/10.1016/s1352-2310(97)83212-8.

[99] A. Sandu, M. Guenther, and S.B. Roberts. Linearly implicit GARK schemes. *Applied Numerical Mathematics*, 161:286–310, 2021. doi: 10.1016/j.apnum.2020.11.014. URL https://doi.org/10.1016/j.apnum.2020.11.014.

[100] Adrian Sandu. Convergence results for implicit–explicit general linear methods, Oct 2020. ISSN 0168-9274. [Online; accessed 4. Dec. 2020].

[101] Adrian Sandu, John Linford, Vishwaz Rao, and Tony D'Augustine. Matlab integration package for stiff ODEs, 2011. URL http://people.cs.vt.edu/~asandu/Software/MatlODE/matlode.html.

[102] Arash Sarshar, Paul Tranquilli, Brent Pickering, Andrew McCall, Christopher J. Roy, and Adrian Sandu. A numerical investigation of matrix-free implicit time-stepping methods for large {CFD} simulations. *Computers & Fluids*, 159:53–63, 2017. ISSN 0045-7930. doi: 10.1016/j.compfluid.2017.09.014. URL http://www.sciencedirect.com/science/article/pii/S0045793017303493.

[103] Arash Sarshar, Steven Roberts, and Adrian Sandu. ADI-GLM coefficients". Mendeley Data, 2019.

[104] Bernhard A. Schmitt and Rüdiger Weiner. Parallel Two-Step W-Methods with Peer Variables. *SIAM Journal on Numerical Analysis*, Jul 2006.

URL https://epubs.siam.org/doi/abs/10.1137/S0036142902411057?casa_token=XpxlKOmeRj4AAAAA:xXnP1_iQXqSq2I1xCsoeSL-gGr-HQK2Pk_uNDQ2jUIcW4gIkdEKxQ3KZArLpmfemRZxxgzdJQXA.

[105] Bernhard A Schmitt, Rüdiger Weiner, and Helmut Podhaisky. Multi-implicit peer two-step w-methods for parallel time integration. *BIT Numerical Mathematics*, 45(1): 197–217, 2005.

[106] Moritz Schneider, Jens Lang, and Willem Hundsdorfer. Extrapolation–based super-convergent implicit–explicit Peer methods with A–stable implicit part. *Journal of Computational Physics*, 367:121 – 133, 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.04.006.

[107] Moritz Schneider, Jens Lang, and Rüdiger Weiner. Super-convergent implicit-explicit peer methods with variable step sizes. *arXiv preprint arXiv:1902.01161*, 2019.

[108] S. Schneider. Convergence results for general linear methods on singular perturbation problems. *BIT Numerical Mathematics*, 33:670–686, 1993.

[109] Felix Schwitzer. W-methods for semilinear parabolic equations. *Applied Numerical Mathematics*, 18(1–3):351 – 366, 1995. ISSN 0168-9274. doi: 10.1016/0168-9274(95)00062-Y. URL http://www.sciencedirect.com/science/article/pii/016892749500062Y.

[110] Tapan Sengupta and Tapan Kumar Sengupta. *High accuracy computing methods: fluid flows and wave phenomena*. Cambridge University Press, 2013.

[111] Tapan K Sengupta and Anurag Dipankar. A comparative study of time advancement methods for solving Navier–Stokes equations. *Journal of Scientific Computing*, 21(2): 225–250, 2004.

[112] Tapan K Sengupta, Anurag Dipankar, and Pierre Sagaut. Error dynamics: beyond von Neumann analysis. *Journal of Computational Physics*, 226(2):1211–1218, 2007.

[113] Tapan K Sengupta, Manoj K Rajpoot, and Yogesh G Bhumkar. Space-time discretizing optimal DRP schemes for flow and wave propagation problems. *Computers & Fluids*, 47(1):144–154, 2011.

[114] T.K Sengupta. A critical assessment of simulations for transitional and turbulent flows. In *Proc. of IUTAM Symp. Advances in Computation, Modeling and Control of Transitional and Turbulent Flows*, pages 491–532. World Scientific Publishing Company, Singapore, 2015.

[115] Karline Soetaert, Jeff Cash, and Francesca Mazzia. *Solving Differential Equations in R | Karline Soetaert | Springer*. Springer-Verlag Berlin Heidelberg, 2012. ISBN 978-3-642-28069-6. doi: 10.1007/978-3-642-28070-2.

[116] Mark Sofroniou and Giulia Spaletta. Construction of explicit Runge–Kutta pairs with stiffness detection. *Mathematical and Computer Modelling*, 40(11-12):1157–1169, 2004.

[117] Trond Steihaug and Arne Wolfbrandt. An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Mathematics of Computation*, 33(146):521–521, may 1979. doi: 10.1090/s0025-5718-1979-0521273-8. URL http://dx.doi.org/10.1090/s0025-5718-1979-0521273-8.

[118] G. W. Stewart. A Krylov–Schur Algorithm for Large Eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, Jul 2006. URL https://epubs-siam-org.ezproxy.lib.vt.edu/doi/abs/10.1137/S0895479800371529?casa_token=eRJz_2fyx2IAAAAA%3A47US_8KaJZHyHV6tLKt_uvkklo9XI3OyEEurX_jzyHTFQMZ1KuvdUXIGHidrvrzPWhiNfNsAYc0&.

[119] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5:506–517, 1968.

[120] Maurizio Tavelli and Michael Dumbser. A staggered space–time discontinuous Galerkin method for the three-dimensional incompressible Navier–Stokes equations on unstructured tetrahedral meshes. *Journal of Computational Physics*, 319:294–323, 2016.

[121] Maurizio Tavelli and Michael Dumbser. A pressure-based semi-implicit space–time discontinuous Galerkin method on staggered unstructured meshes for the solution of the compressible Navier–Stokes equations at all Mach numbers. *Journal of Computational Physics*, 341:341–376, 2017.

[122] P. Tranquilli and A. Sandu. Rosenbrock-Krylov methods for large systems of differential equations. *SIAM Journal on Scientific Computing*, 36(3):A1313–A1338, 2014. doi: 10.1137/130923336. URL http://dx.doi.org/10.1137/130923336.

[123] P. Tranquilli, R. Glandon, A. Sarshar, and A. Sandu. Analytical Jacobian-vector products for matrix-free methods. *Journal of Computational and Applied Mathematics*, 310:213–223, 2017. doi: 10.1016/j.cam.2016.05.002. URL http://www.sciencedirect.com/science/article/pii/S0377042716302199.

[124] P. Tranquilli, R. Glandon, and A. Sandu. Subspace adaptivity in Rosenbrock-Krylov methods for the time integration of initial value problems. *Journal of Computational and Applied Mathematics*, 385:113188, 2021. doi: 10.1016/j.cam.2020.113188. URL https://doi.org/10.1016/j.cam.2020.113188.

[125] Paul Tranquilli, S. Ross Glandon, Arash Sarshar, and Adrian Sandu. Analytical jacobian-vector products for the matrix-free time integration of partial differential equations. *Journal of Computational and Applied Mathematics*, pages –, 2016. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2016.05.002. URL http://www.sciencedirect.com/science/article/pii/S0377042716302199.

[126] PJ Van der Houwen. Explicit Runge–Kutta formulas with increased stability boundaries. *Numerische Mathematik*, 20(2):149–164, 1972.

[127] Bram van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.

[128] R. Weiner, B. A. Schmitt, and H. Podhaisky. Parallel 'Peer' two-step W-methods and their application to MOL-systems. *Applied Numerical Mathematics*, 48(3):425–439, Mar 2004. ISSN 0168-9274. doi: 10.1016/j.apnum.2003.10.005.

[129] N.N. Yanenko. *The Method of Fractional-Steps*. Springer, Berlin Heidelberg NewYork, 1971.

[130] H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters*, 150: 262–268, 1990.

[131] H. Zhang and A. 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, April 2012. doi: 10.1016/j.procs.2012.04.112. URL http://dx.doi.org/10.1016/j.procs.2012.04.112.

[132] H. Zhang, A. Sandu, and S. 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.

[133] H. Zhang, A. Sandu, and P. Tranquilli. Application of approximate matrix factorization to high-order linearly-implicit Runge-Kutta methods. *Journal of Computational and Applied Mathematics*, 286:196–210, 2015. doi: 10.1016/j.cam.2015.03.005. URL http://dx.doi.org/10.1016/j.cam.2015.03.005.

[134] H. Zhang, A. Sandu, and S. 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.

[135] Hong Zhang, Adrian Sandu, and Sebastien Blaise. High order implicit-explicit general linear methods with optimized stability regions. *SIAM Journal on Scientific Computing*, 38(3):A1430–A1453, 2016.

[136] E. Zharovsky and A. Sandu. IMEX two-step Runge-Kutta methods. Technical Report CS TR-12-08, Computational Science Laboratory, Virginia Tech, February 2012.

[137] E. Zharovsky, A. Sandu, and H. Zhang. A class of IMEX two-step Runge-Kutta methods. *SIAM Journal on Numerical Analysis*, 53(1):321–341, 2015. doi: 10.1137/130937883. URL http://dx.doi.org/10.1137/130937883.

Appendices

Appendix A

Order reduction with matrix-free methods

A.1 Lorenz-96 problem

In this section we confirm the orders of convergence for the time-stepping methods used in chapter 2 by applying them to Lorenz–96 test problem [74]. Proposed by Edward Lorenz, this model is often used to model chaotic behavior of atmospheric systems:

$$\frac{\partial u_i}{\partial t} = (u_{i+1} - u_{i-2})\, u_{i-1} - u_i + F \qquad \text{for} \quad i = 1, \cdots, 40.$$

The Jacobian of this system is a banded matrix that can be implemented in sparse format and used in the implicit methods of Section 2.2. Setting the external force factor $F = 8$ and using a range of fixed time steps to propagate the model forward, we observe nearly full order for all of the methods as reported in Table A.1. One source of local error causing order reduction can be traced back to the truncation errors made by replacing the Jacobian-vector product in equation (2.11) with a first-order finite difference approximation. Poor convergence of the Newton's iteration as well as the Krylov-based solver for the linear system in the implicit methods are other sources of error contributing to inexact stage vectors and ultimately to order reduction. It is notable, however, that in all the convergence tests the Rosenbrock-Krylov method retains its full order of convergence. Unlike fully and linearly implicit methods, the system solved in equation (2.10c) is formed using the Arnoldi iteration with an inexact Jacobian from the beginning and is solved using a direct method and therefore the issues arising from the convergence of the iterative solvers is avoided altogether. Interested readers may consult [122] for a detailed analysis of this phenomenon.

Table A.1: Orders of convergence for methods applied to the Lorenz–96 problem.

Method	Numerical order	Theoretical order
ERK4	4.00	4
ERK5 (DOPRI5)	5.58	5
ERK5 (DOPRI853)	6.14	8
SDIRK	3.89	4
ROS4	3.91	4
ROW	2.94	3
ROK	3.85	4
ODE45	5.47	5

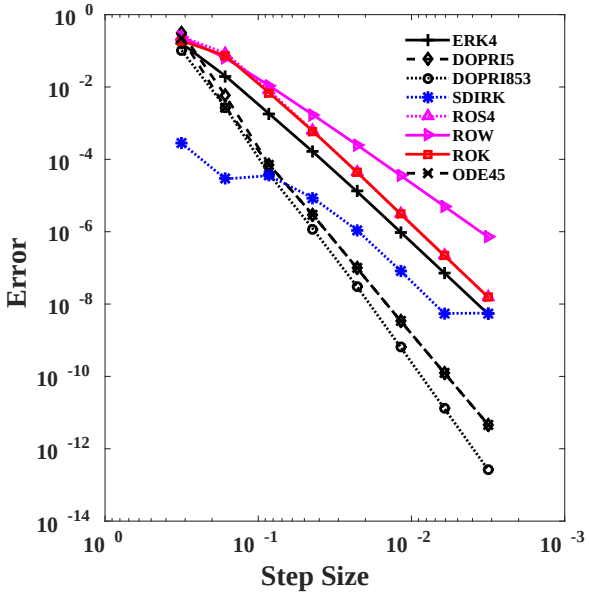


Figure A.1: Convergence diagram for the Lorenz–96 test problem.

Appendix B

Decoupled Multirate GARK methods

B.1 Decoupled multirate GARK schemes

We use the naming convention *FASTf-SLOWs p(p̂)[type]*, where p is the method order, $\widehat{p}$ is the embedded order, f is the number of stages in the fast base method, and s is the number of stages in the slow base method. Each component method is either explicit or implicit: FAST, SLOW $\in$ {EX,IM}. We distinguish between methods of type A (optimized for accuracy and for better step size control) and methods of type S (optimized for simplicity and for stability), therefore *type* $\in$ {A,S}.

B.1.1 MrGARK EX2-EX2 2(1)[A]

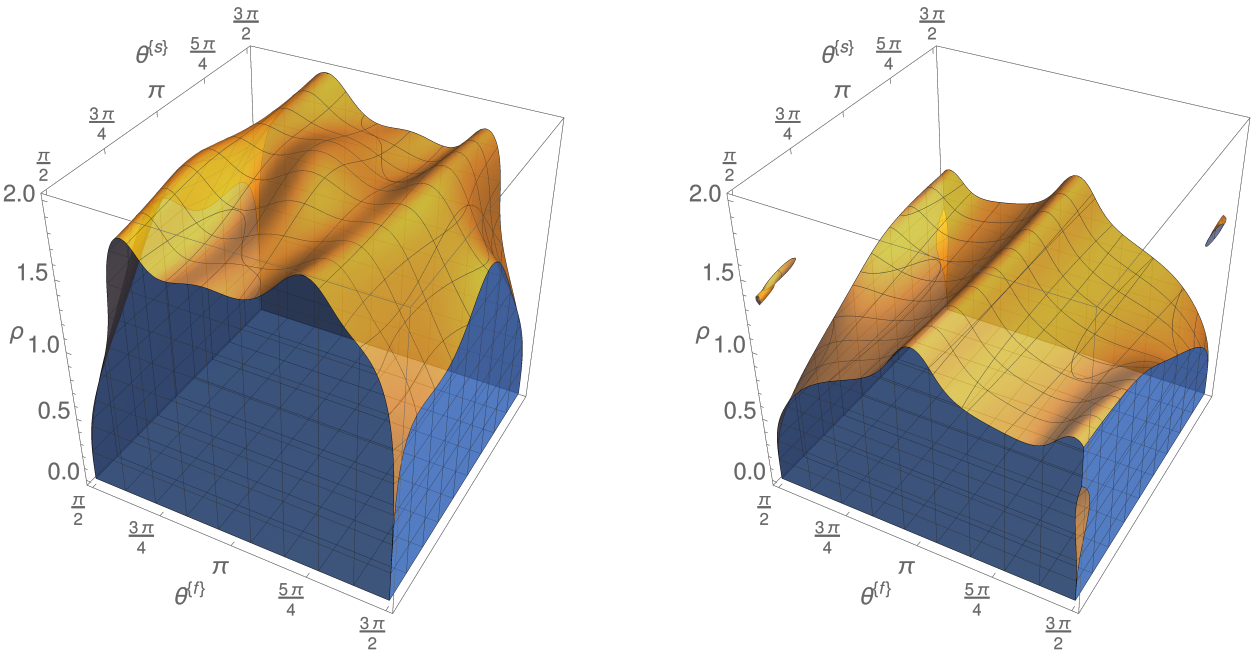
This explicit method uses a base method from [86] and has telescopic eq. (3.6) and naturally adaptive (proposition 3.7) properties.

$$A^{\{\mathfrak{f},\mathfrak{f}\}} = \begin{bmatrix} 0 & 0 \\ \frac{2}{3} & 0 \end{bmatrix}, \qquad A^{\{\mathfrak{s},\mathfrak{s}\}} = \begin{bmatrix} 0 & 0 \\ \frac{2}{3} & 0 \end{bmatrix}, \qquad A^{\{\mathfrak{f},\mathfrak{s},1\}} = \begin{bmatrix} 0 & 0 \\ \frac{2}{3M} & 0 \end{bmatrix},$$

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \frac{3M^3-11M^2+20\lambda M-20M-20\lambda+20}{20(M-1)M} & -\frac{M(3M-11)}{20(M-1)} \\ \frac{-3M^3-9M^2+60\lambda M-20M-60\lambda+20}{60(M-1)M} & \frac{M(M+3)}{20(M-1)} \end{bmatrix}, \qquad \lambda = 2,\dots,M,$$

$$A^{\{\mathfrak{s},\mathfrak{f},1\}} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{3}(M-2)M & \frac{M^2}{3} \end{bmatrix}, \quad A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \lambda = 2,\dots,M,$$

$$b^{\{\mathfrak{f}\}} = b^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \end{bmatrix}^T, \qquad \widehat{b}^{\{\mathfrak{f}\}} = \widehat{b}^{\{\mathfrak{s}\}} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T.$$



(a) $M = 2$. (b) $M = 4$.

Figure B.1: EX2-EX2 2(1)[A] stability regions.

B.1.2 MrGARK EX2-EX2 2(1)[S]

This explicit method uses a two stage base method and has telescopic eq. (3.6) and naturally adaptive (proposition 3.7) properties. The scheme computes the first slow stage, followed by L_2 fast steps, then the second slow stage, followed by $M - L_2$ fast steps. For example, we can take $L_2 = floor(c_2 M)$.

$$A^{\{\mathfrak{f},\mathfrak{f}\}} = \begin{bmatrix} 0 & 0 \\ c_2 & 0 \end{bmatrix}, \qquad A^{\{\mathfrak{s},\mathfrak{s}\}} = \begin{bmatrix} 0 & 0 \\ c_2 & 0 \end{bmatrix},$$

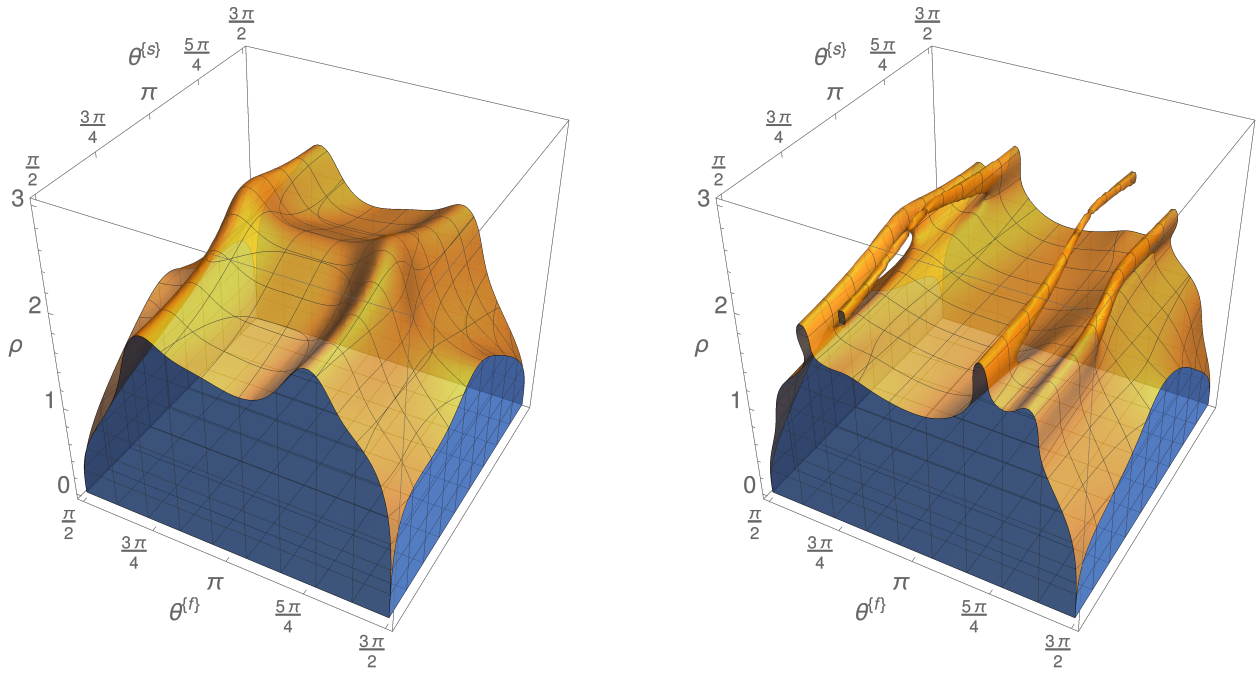
$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \frac{\lambda-1}{M} & 0 \\ \frac{\lambda+c_2-1}{M} & 0 \end{bmatrix}, \qquad \lambda = 1,\ldots,L_2,$$

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \frac{(\lambda-1)(2c_2-1)}{2Mc_2} & \frac{\lambda-1}{2Mc_2} \\ \frac{M}{3(L_2-M)} + \frac{-\lambda+2c_2(2\lambda+c_2-2)+1}{2Mc_2} & \frac{M}{3M-3L_2} + \frac{\lambda-1}{2Mc_2} + \frac{1-\lambda}{M} \end{bmatrix}, \quad \lambda = L_2+1,\ldots,M,$$

$$A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ \frac{M(-2M+6c_2+3L_2-3)}{6L_2} & \frac{M(2M-3L_2+3)}{6L_2} \end{bmatrix}, \qquad \lambda = 1,\ldots,L_2,$$

$$A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \lambda = L_2+1,\ldots,M,$$

$$b^{\{\mathfrak{f}\}} = b^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{2c_2-1}{2c_2} & \frac{1}{2c_2} \end{bmatrix}^T, \qquad \widehat{b}^{\{\mathfrak{f}\}} = \widehat{b}^{\{\mathfrak{s}\}} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T.$$



(a) $M = 2$. (b) $M = 4$.

Figure B.2: EX2-EX2 2(1)[S] stability regions with $c_2 = \frac{2}{3}$.

B.1.3 MrGARK EX2-IM2 2(1)[A]

This explicit-implicit method uses the fast method from [86] and slow method from [2]. The multirate scheme is stiffly accurate (3.21) in the slow partition and the coupling error is independent of M.

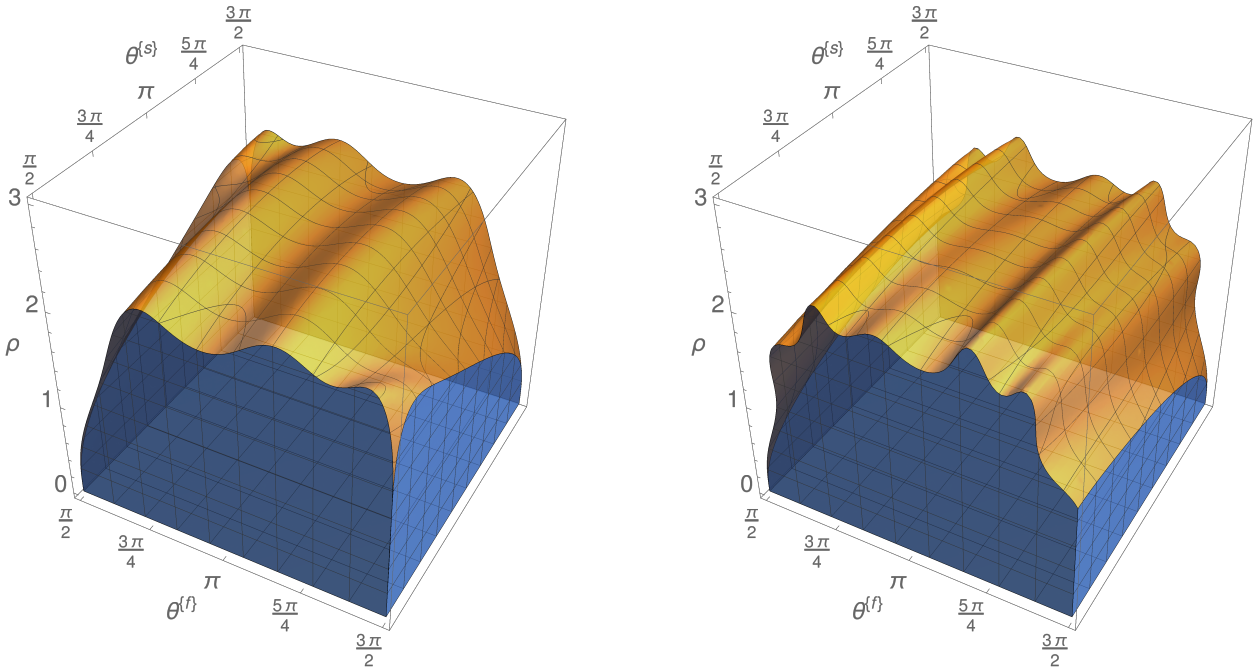
$$A^{\{\mathfrak{f},\mathfrak{f}\}} = \begin{bmatrix} 0 & 0 \\ \frac{2}{3} & 0 \end{bmatrix}, \qquad A^{\{\mathfrak{s},\mathfrak{s}\}} = \begin{bmatrix} 1-\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{bmatrix}, \quad A^{\{\mathfrak{s},\mathfrak{f},1\}} = \begin{bmatrix} M-\frac{M}{\sqrt{2}} & 0 \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix},$$

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \frac{\lambda-1}{M} & 0 \\ \frac{3\lambda-1}{3M} & 0 \end{bmatrix}, \quad \lambda = 1,\ldots,M,$$

$$A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}, \qquad \lambda = 2,\ldots,M,$$

$$b^{\{\mathfrak{f}\}} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \end{bmatrix}^T, \qquad b^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{bmatrix}^T,$$

$$\widehat{b}^{\{\mathfrak{f}\}} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \qquad \widehat{b}^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{3}{5} & \frac{2}{5} \end{bmatrix}^T.$$



(a) $M = 2$. (b) $M = 4$.

FIGURE B.3: EX2-IM2 2(1)[A] stability regions.

B.1.4 MrGARK IM2-EX2 2(1)[A]

This implicit-explicit method uses the fast method from [2] and the slow method from [86]. The multirate scheme is stiffly accurate (3.21) in the fast partition.

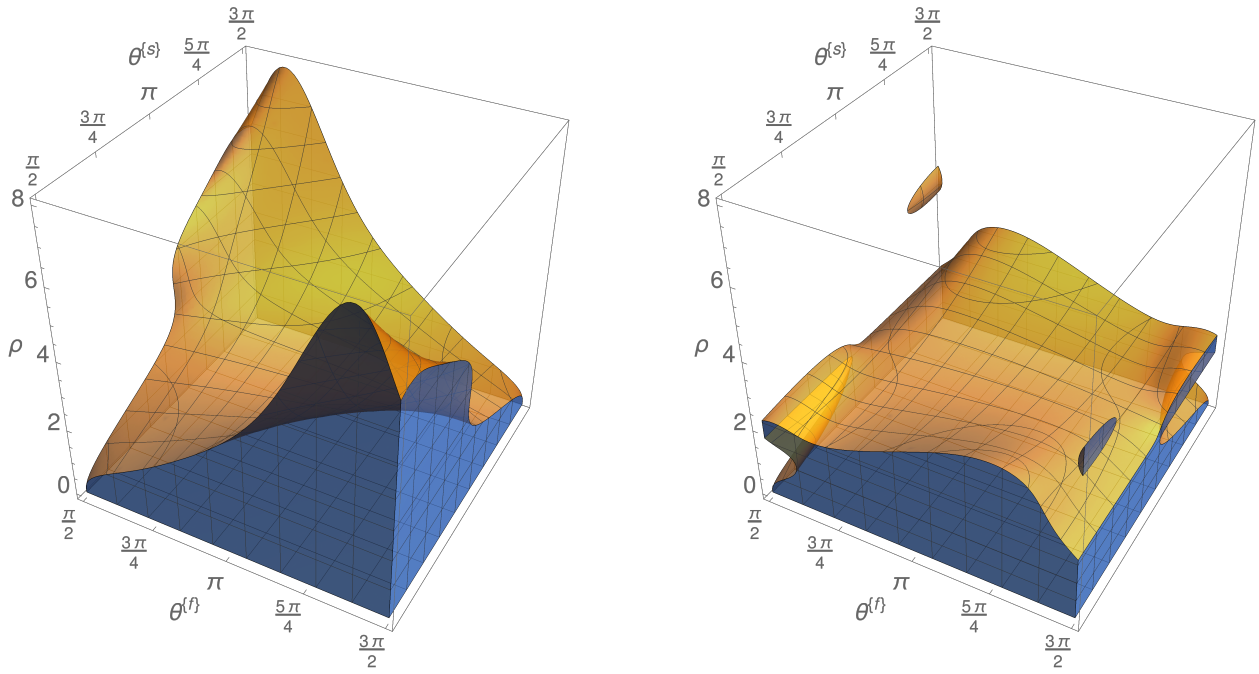
$$A^{\{\mathfrak{f},\mathfrak{f}\}} = \begin{bmatrix} 1-\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{bmatrix}, \quad A^{\{\mathfrak{s},\mathfrak{s}\}} = \begin{bmatrix} 0 & 0 \\ \frac{2}{3} & 0 \end{bmatrix}, \quad A^{\{\mathfrak{f},\mathfrak{s},M\}} = \begin{bmatrix} \frac{2M-\sqrt{2}}{2M} & 0 \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix},$$

$$A^{\{\mathfrak{f},\mathfrak{s},\lambda\}} = \begin{bmatrix} \frac{2\lambda-\sqrt{2}}{2M} & 0 \\ \frac{\lambda}{M} & 0 \end{bmatrix}, \qquad \lambda = 1, \ldots, M-1,$$

$$A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ \frac{2}{3} & 0 \end{bmatrix}, \qquad \lambda = 1, \ldots, M,$$

$$b^{\{\mathfrak{f}\}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{bmatrix}^T, \qquad b^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \end{bmatrix}^T,$$

$$\widehat{b}^{\{\mathfrak{f}\}} = \begin{bmatrix} \frac{3}{5} & \frac{2}{5} \end{bmatrix}^T. \qquad \widehat{b}^{\{\mathfrak{s}\}} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T,$$



(a) $M = 2$. (b) $M = 4$.

FIGURE B.4: IM2-EX2 2(1)[A] stability regions.

B.1.5 MrGARK EX3-EX3 3(2)[A]

This explicit method uses a base method from [86] and has telescopic eq. (3.6) properties. The coupling error is independent of M.

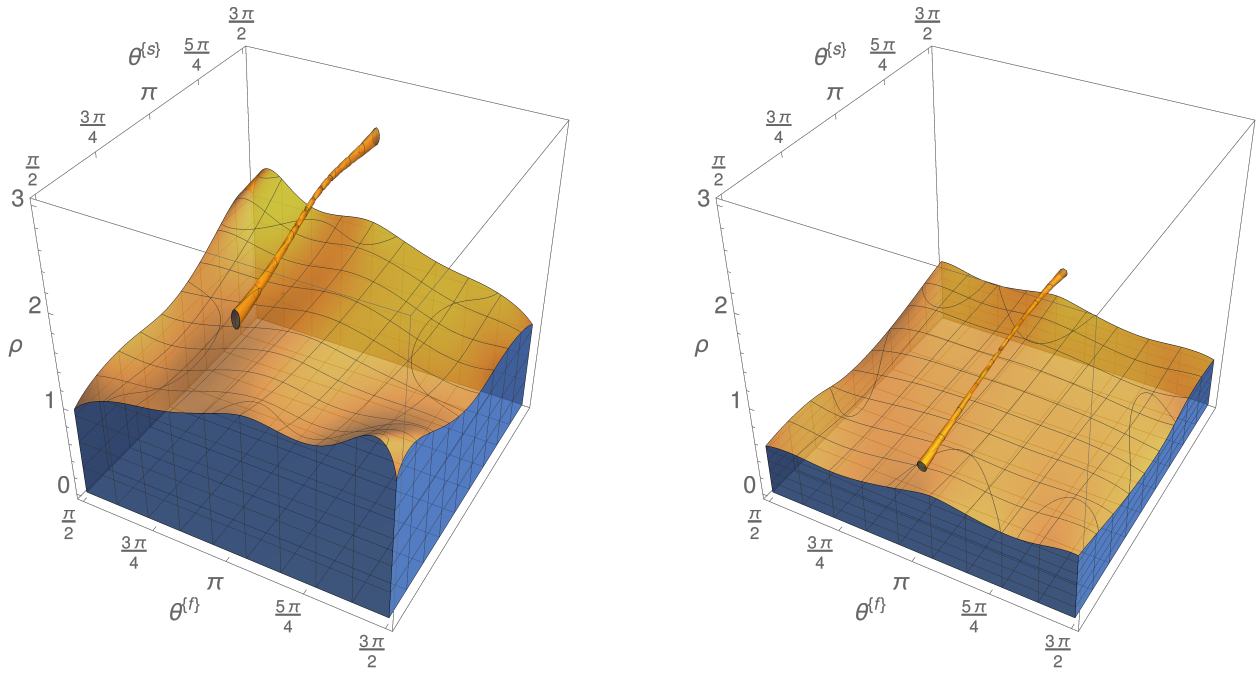
$$A^{\{f,f\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{4} & 0 \end{bmatrix}, \qquad A^{\{s,s\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{4} & 0 \end{bmatrix}, \qquad A^{\{f,s,1\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2M} & 0 & 0 \\ 0 & \frac{3}{4M} & 0 \end{bmatrix},$$

$$A^{\{f,s,\lambda\}} = \begin{bmatrix} \frac{3M^3-8M^2+6\lambda M-6\lambda+6}{6(M-1)M} & \frac{-3M^2+8M-6}{6(M-1)} & 0 \\ \frac{-2M^2+6\lambda M-3M-6\lambda+3}{6(M-1)M} & \frac{M}{3(M-1)} & 0 \\ \frac{-3M^3+2M^2+12\lambda M-9M-12\lambda+12}{12(M-1)M} & \frac{3M^3-2M^2+6M-9}{12(M-1)M} & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,$$

$$A^{\{s,f,1\}} = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{66}M(16M-33) & \frac{8M^2}{33} & 0 \\ \frac{1}{264}\left(11M^4-22M^3+26M^2+11M+44\right) & \frac{1}{88}\left(-11M^4+22M^3-16M^2-11M+22\right) & \frac{1}{12}\left(M^4-2M^3+M^2+M+4\right) \end{bmatrix},$$

$$A^{\{s,f,\lambda\}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-M^4+2M^3+2M^2+3M-4}{24(M-1)} & \frac{1}{8}\left(M^3-M^2-M+2\right) & \frac{-M^4+2M^3-M^2+3M-4}{12(M-1)} \end{bmatrix}, \quad \lambda = 2, \ldots, M,$$

$$b^{\{f\}} = b^{\{s\}} = \begin{bmatrix} \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{bmatrix}^T, \quad \widehat{b}^{\{f\}} = \widehat{b}^{\{s\}} = \begin{bmatrix} \frac{1}{40} & \frac{37}{40} & \frac{1}{20} \end{bmatrix}^T.$$



(a) $M = 2$. (b) $M = 4$.

Figure B.5: EX3-EX3 3(2)[A] stability regions.

B.1.6 MrGARK EX4-EX4 3(2)[A]

This explicit method is telescopic eq. (3.6) and naturally adaptive (proposition 3.7).

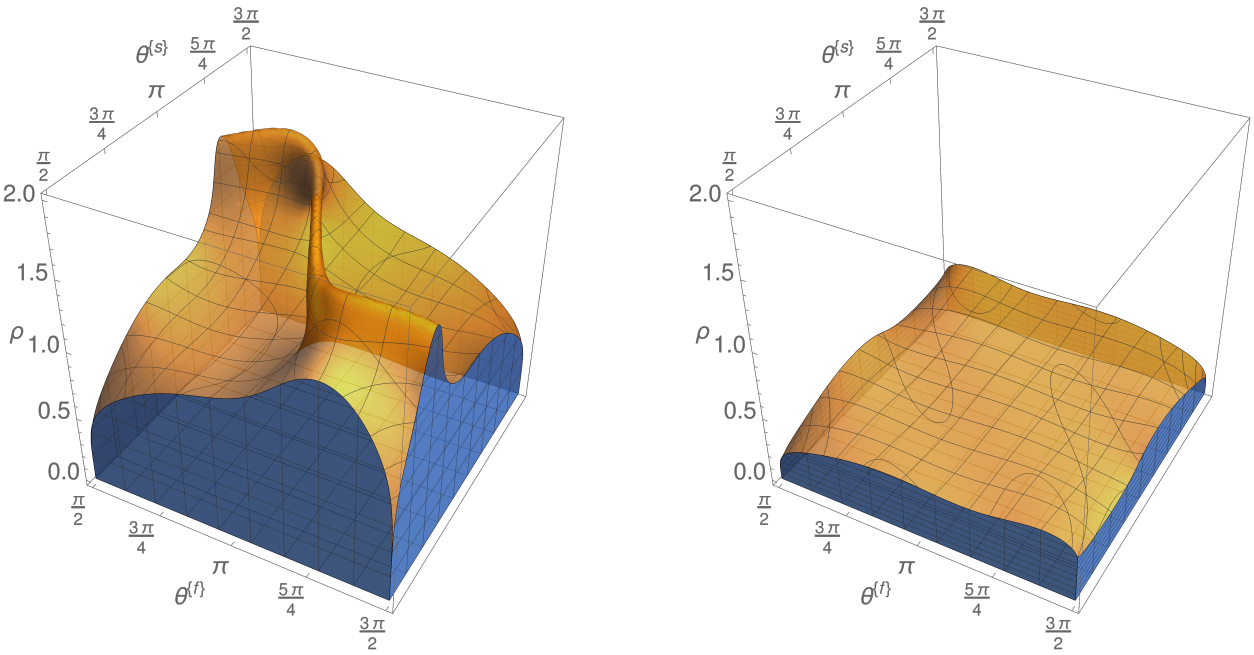
$$A^{\{\mathfrak{f},\mathfrak{f}\}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{5}{9} & 0 & 0 \\ \frac{833}{7680} & \frac{833}{9216} & \frac{3213}{5120} & 0 \end{bmatrix}, \quad A^{\{\mathfrak{s},\mathfrak{s}\}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{5}{9} & 0 & 0 \\ \frac{833}{7680} & \frac{833}{9216} & \frac{3213}{5120} & 0 \end{bmatrix}, \quad A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \lambda = 2, \dots, M,$$

$$A^{\{\mathfrak{f},\mathfrak{s},1\}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{3M} & 0 & 0 & 0 \\ \frac{5(518M^3-2140M^2+2399M-777)}{2331M(3M-4)} & -\frac{5(518M^3-2140M^2+1622M+259)}{2331M(3M-4)} & 0 & 0 \\ \frac{17(141932M^3-445231M^2+481160M-178710)}{852480M(3M-4)} & -\frac{17(94535M^3-228442M^2+142736M-5180)}{340992M(3M-4)} & \frac{3213M}{5120} & 0 \end{bmatrix},$$

$$A^{\{\mathfrak{s},\mathfrak{f},\lambda\}} = \begin{bmatrix} \frac{\lambda-1}{M} & 0 & 0 & 0 \\ \frac{3\lambda-2}{3M} & 0 & 0 & 0 \\ \frac{-5965M^3+6993\lambda M^2+12092M^2-16317\lambda M-858M+9324\lambda-5439}{2331M(3M^2-7M+4)} & \frac{5(1193M^3-3040M^2+1622M+259)}{2331M(3M^2-7M+4)} & 0 & 0 \\ \frac{-867119M^3+511488\lambda M^2+1937719M^2-1193472\lambda M-1006056M+681984\lambda-74370}{170496M(3M^2-7M+4)} & \frac{17(51007M^3-119207M^2+71368M-2590)}{170496M(3M^2-7M+4)} & 0 & 0 \end{bmatrix}, \quad \lambda = 2, \dots, M,$$

$$A^{\{\mathfrak{s},\mathfrak{f},1\}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{361M-102M^2}{1083} & \frac{34M^2}{361} & 0 & 0 \\ 0 & -\frac{5M(981M-1805)}{6498} & \frac{5M(327M-361)}{2166} & 0 \\ \frac{M(1480461M^2-3944118M+3007130)}{2772480} & -\frac{119M(3249M^2-20358M+18050)}{3326976} & -\frac{119M(66063M^2-78954M-18050)}{5544960} & (M-1)M^2 \end{bmatrix},$$

$$b^{\{\mathfrak{f}\}} = b^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{101}{714} & \frac{1}{3} & \frac{1}{6} & \frac{128}{357} \end{bmatrix}^T, \quad \widehat{b}^{\{\mathfrak{f}\}} = \widehat{b}^{\{\mathfrak{s}\}} = \begin{bmatrix} \frac{7}{40} & -\frac{425}{8784} & \frac{100037}{131760} & \frac{188}{1647} \end{bmatrix}^T.$$



(a) $M = 2$. (b) $M = 4$.

FIGURE B.6: EX4-EX4 3(2)[A] stability regions.

B.1.7 MrGARK EX3-EX3 3(2)[S]

This explicit method uses a three stage base method and has telescopic eq. (3.6) property. Once again, we can take $L_2 = floor(c_2 M)$.

$$A^{\{\mathrm{f},\mathrm{f}\}} = A^{\{\mathrm{s},\mathrm{s}\}} = \begin{bmatrix} 0 & 0 & 0 \\ c_2 & 0 & 0 \\ \frac{(3c_2^2-3c_2+1)}{c_2(3c_2-2)} & \frac{(c_2-1)}{c_2(3c_2-2)} & 0 \end{bmatrix},$$

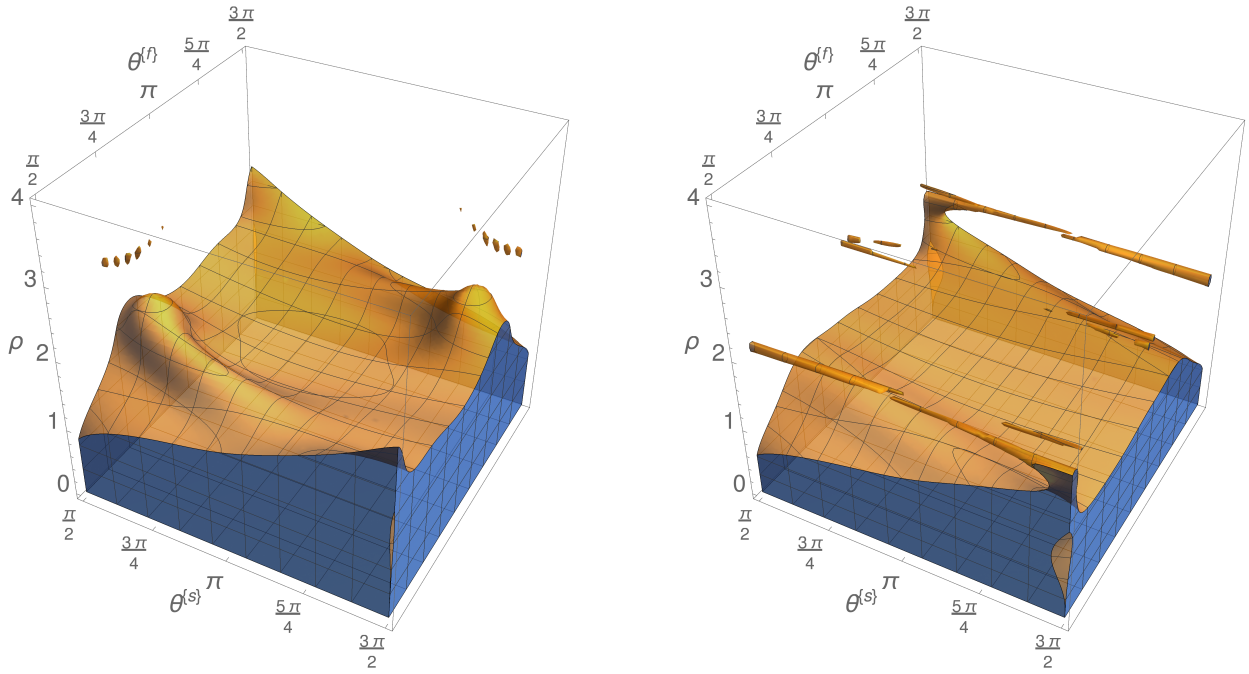
$$A^{\{\mathrm{f},\mathrm{s},\lambda\}} = \begin{bmatrix} \frac{\lambda-1}{M} & 0 & 0 \\ \frac{c_2+\lambda-1}{M} & 0 & 0 \\ \frac{\lambda}{M} & 0 & 0 \end{bmatrix}, \qquad \lambda = 1,\dots,L_2,$$

$$A^{\{\mathrm{f},\mathrm{s},\lambda\}} = \begin{bmatrix} \frac{2\lambda-1}{12c_2(L_2-M)} + \frac{1-2\lambda}{12c_2(L_2+M)} + \frac{2\lambda-1}{2M} & \frac{2\lambda-1}{12c_2(L_2+M)} + \frac{1-2\lambda}{12c_2(L_2-M)} - \frac{1}{2M} & 0 \\ \frac{2\lambda-1}{12c_2(L_2-M)} + \frac{1-2\lambda}{12c_2(L_2+M)} + \frac{2\lambda-1}{2M} & \frac{2\lambda-1}{12c_2(L_2+M)} + \frac{1-2\lambda}{12c_2(L_2-M)} + \frac{2c_2-1}{2M} & 0 \\ \frac{2\lambda-1}{12c_2(L_2-M)} + \frac{1-2\lambda}{12c_2(L_2+M)} + \frac{2\lambda-1}{2M} & \frac{2\lambda-1}{12c_2(L_2+M)} + \frac{1-2\lambda}{12c_2(L_2-M)} + \frac{1}{2M} & 0 \end{bmatrix}, \qquad \lambda = L_2+1,\dots,M,$$

$$A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 \\ c_2\left(\frac{2\lambda(c_2(4L_2-3)-3L_2+3)}{(c_2-1)(3c_2^2+4c_2+1)(L_2+1)} + \frac{M}{L_2}\right) & -\frac{c_2\lambda(c_2(4L_2-3)-3L_2+3)}{(c_2-1)(3c_2^2+4c_2+1)(L_2+1)} & -\frac{c_2\lambda(c_2(4L_2-3)-3L_2+3)}{(c_2-1)(3c_2^2+4c_2+1)(L_2+1)} \\ \frac{2c_2\lambda(c_2(4L_2-3)-3L_2+3)}{(c_2-1)(3c_2^2+4c_2+1)(L_2+1)} & -\frac{c_2\lambda(c_2(4L_2-3)-3L_2+3)}{(c_2-1)(3c_2^2+4c_2+1)(L_2+1)} & -\frac{c_2\lambda(c_2(4L_2-3)-3L_2+3)}{(c_2-1)(3c_2^2+4c_2+1)(L_2+1)} \end{bmatrix}, \qquad \lambda = 1,\dots,L_2,$$

$$A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\lambda}{3c_2-2} + \frac{c_2(3L_2-4)-3L_2+3}{6c_2-4} + \frac{M}{M-L_2} & \frac{\lambda}{2-3c_2} & \frac{c_2(4-3L_2)+3(L_2-1)}{6c_2-4} \end{bmatrix}, \qquad \lambda = L_2+1,\dots,M,$$

$$b^{\{\mathrm{f}\}} = b^{\{\mathrm{s}\}} = \begin{bmatrix} \frac{3c_2-1}{6c_2} & -\frac{1}{6(c_2-1)c_2} & \frac{3c_2-2}{6(c_2-1)} \end{bmatrix}^T, \quad \widehat{b}^{\{\mathrm{f}\}} = \widehat{b}^{\{\mathrm{s}\}} = \begin{bmatrix} \hat{b}_2\,(c_2-1) + \frac{1}{2} & \hat{b}_2 & \frac{1-2\hat{b}_2c_2}{2} \end{bmatrix}^T.$$



(a) $M = 2$. (b) $M = 4$.

FIGURE B.7: Stability plots for MrGARK EX3-EX3 3(2)[S] method with $c_2 = \frac{1}{2}$.

B.1.8 MrGARK EX3-IM3 3(2)[A]

This explcit method uses a fast method from [86] and a slow method from [2] and is stiffly accurate (3.21) in the slow partition.

$$A^{\{\mathrm{f},\mathrm{f}\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{4} & 0 \end{bmatrix}, \quad A^{\{\mathrm{s},\mathrm{s}\}} = \begin{bmatrix} \gamma & 0 & 0 \\ -\frac{2(3\gamma^3-9\gamma^2+6\gamma-1)}{3(2\gamma^2-4\gamma+1)} & \gamma & 0 \\ \frac{4\gamma-1}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & -\frac{3(2\gamma^2-4\gamma+1)^2}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & \gamma \end{bmatrix},$$

$$A^{\{\mathrm{f},\mathrm{s},\lambda\}} = \begin{bmatrix} \frac{\lambda-1}{M} & 0 & 0 \\ \frac{2\lambda-1}{2M} & 0 & 0 \\ \frac{-60\lambda\gamma^3+42\gamma^3+18M\gamma^2+72\lambda\gamma^2-72\gamma^2-36M\gamma+42\lambda\gamma+3\gamma+9M-16\lambda+4}{16M(3\gamma^3-9\gamma^2+6\gamma-1)} & -\frac{9(2\gamma^2-4\gamma+1)(M+3\gamma-6\gamma\lambda)}{16M(3\gamma^3-9\gamma^2+6\gamma-1)} & 0 \end{bmatrix}, \quad \lambda = 1,\ldots,M,$$

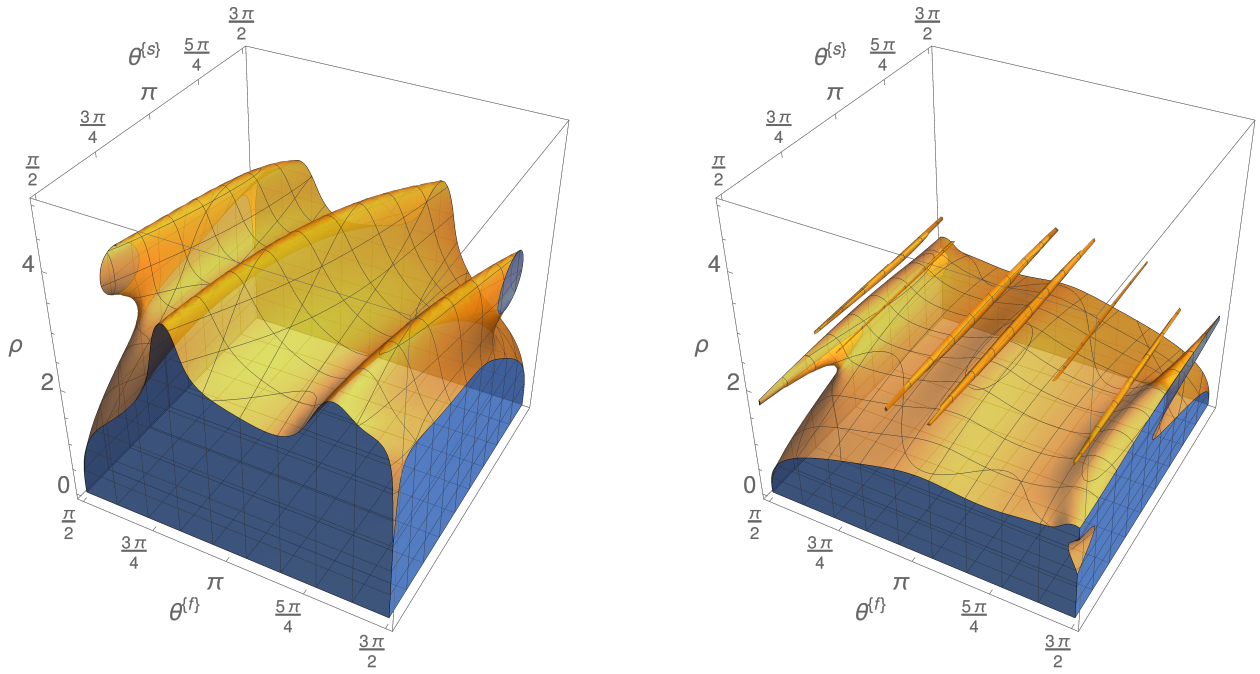
$$A^{\{\mathrm{s},\mathrm{f},1\}} = \begin{bmatrix} M\gamma & 0 & 0 \\ -\frac{M(36M\gamma^4-36\gamma^4-120M\gamma^3+126\gamma^3+108M\gamma^2-138\gamma^2-36M\gamma+51\gamma+4M-6)}{9(2\gamma^2-4\gamma+1)^2} & \frac{4M^2(9\gamma^4-30\gamma^3+27\gamma^2-9\gamma+1)}{9(2\gamma^2-4\gamma+1)^2} & 0 \\ \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{bmatrix},$$

$$A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{bmatrix}, \quad \lambda = 2,\ldots,M,$$

$$b^{\{\mathrm{f}\}} = \begin{bmatrix} \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{bmatrix}^T, \quad b^{\{\mathrm{s}\}} = \begin{bmatrix} \frac{4\gamma-1}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & -\frac{3(2\gamma^2-4\gamma+1)^2}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & \gamma \end{bmatrix}^T,$$

$$\widehat{b}^{\{\mathrm{f}\}} = \begin{bmatrix} \frac{1}{40} & \frac{37}{40} & \frac{1}{20} \end{bmatrix}^T, \quad \widehat{b}^{\{\mathrm{s}\}} = \begin{bmatrix} \frac{-6\gamma^2+6\gamma-1}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & \frac{3(4\gamma^3-10\gamma^2+6\gamma-1)}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & 0 \end{bmatrix}^T,$$

$$\gamma = \tfrac{1}{2}\left(2+\sqrt{6}\sin\left(\tfrac{1}{3}\cot^{-1}2\sqrt{2}\right)-\sqrt{2}\cos\left(\tfrac{1}{3}\cot^{-1}2\sqrt{2}\right)\right) \approx 0.43586652150845899942.$$



(a) $M = 2$. (b) $M = 4$.

Figure B.8: EX3-IM3 3(2)[A] stability regions.

B.1.9 MrGARK IM3-EX3 3(2)[A]

This implicit-explicit method uses a fast method from [2] and a slow method from [86] and is stiffly accurate (3.21) in the fast partition.

$$A^{\{\mathsf{f},\mathsf{f}\}} = \begin{bmatrix} \gamma & 0 & 0 \\ -\frac{2(3\gamma^3-9\gamma^2+6\gamma-1)}{3(2\gamma^2-4\gamma+1)} & \gamma & 0 \\ \frac{4\gamma-1}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & -\frac{3(2\gamma^2-4\gamma+1)^2}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & \gamma \end{bmatrix}, \quad A^{\{\mathsf{s},\mathsf{s}\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{4} & 0 \end{bmatrix},$$

$$A^{\{\mathsf{f},\mathsf{s},\lambda\}} = \begin{bmatrix} \frac{\gamma+\lambda-1}{M} & 0 & 0 \\ \frac{6\lambda\gamma^2-12\lambda\gamma+3\gamma+3\lambda-1}{3M(2\gamma^2-4\gamma+1)} & 0 & 0 \\ \frac{\lambda}{M} & 0 & 0 \end{bmatrix}, \qquad \lambda = 1, \ldots, M-1,$$

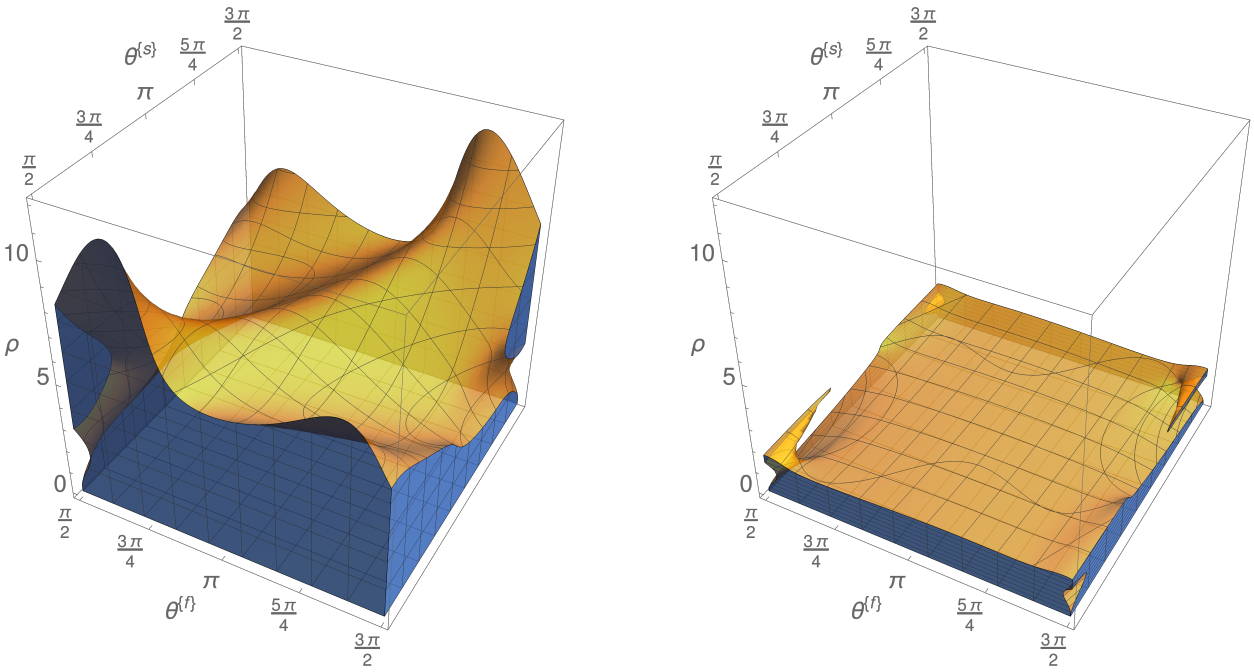
$$A^{\{\mathsf{f},\mathsf{s},M\}} = \begin{bmatrix} \frac{M+\gamma-1}{M} & 0 & 0 \\ \frac{12M^2\gamma^3-36M\gamma^3+18\gamma^3-36M^2\gamma^2+108M\gamma^2-42\gamma^2+24M^2\gamma-60M\gamma+21\gamma-4M^2+9M-3}{9M(2\gamma^2-4\gamma+1)^2} & -\frac{4(M-3\gamma)(3\gamma^3-9\gamma^2+6\gamma-1)}{9(2\gamma^2-4\gamma+1)^2} & 0 \\ \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{bmatrix},$$

$$A^{\{\mathsf{s},\mathsf{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ -\frac{3(12\gamma^3+6M\gamma^2-18\gamma^2-12M\gamma+6\gamma+3M-1)}{32(3\gamma^3-9\gamma^2+6\gamma-1)} & \frac{9(M+6\gamma-3)(2\gamma^2-4\gamma+1)}{32(3\gamma^3-9\gamma^2+6\gamma-1)} & 0 \end{bmatrix}, \qquad \lambda = 1, \ldots, M,$$

$$b^{\{\mathsf{f}\}} = \begin{bmatrix} \frac{4\gamma-1}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & -\frac{3(2\gamma^2-4\gamma+1)^2}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & \gamma \end{bmatrix}^T, \qquad b^{\{\mathsf{s}\}} = \begin{bmatrix} \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{bmatrix}^T,$$

$$\widehat{b}^{\{\mathsf{f}\}} = \begin{bmatrix} \frac{-6\gamma^2+6\gamma-1}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & \frac{3(4\gamma^3-10\gamma^2+6\gamma-1)}{4(3\gamma^3-9\gamma^2+6\gamma-1)} & 0 \end{bmatrix}^T, \qquad \widehat{b}^{\{\mathsf{s}\}} = \begin{bmatrix} \frac{1}{40} & \frac{37}{40} & \frac{1}{20} \end{bmatrix}^T.$$

$$\gamma = \tfrac{1}{2}\left(2 + \sqrt{6}\sin\left(\tfrac{1}{3}\cot^{-1} 2\sqrt{2}\right) - \sqrt{2}\cos\left(\tfrac{1}{3}\cot^{-1} 2\sqrt{2}\right)\right) \approx 0.43586652150845899942.$$



(a) $M = 2$. (b) $M = 4$.

Figure B.9: IM3-EX3 3(2)[A] stability regions.

B.1.10 MrGARK EX5-EX5 4(3)[A]

This explicit method uses a base method from [116], is telescopic eq. (3.6) and has the First same as last (FSAL) property.

$$
A^{\{\mathrm{f},\mathrm{f}\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{5} & 0 & 0 & 0 & 0 & 0 \\ -\frac{3}{20} & \frac{3}{4} & 0 & 0 & 0 & 0 \\ \frac{19}{44} & -\frac{15}{44} & \frac{10}{11} & 0 & 0 & 0 \\ \frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72} & 0 & 0 \end{bmatrix}, \quad A^{\{\mathrm{s},\mathrm{s}\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{5} & 0 & 0 & 0 & 0 & 0 \\ -\frac{3}{20} & \frac{3}{4} & 0 & 0 & 0 & 0 \\ \frac{19}{44} & -\frac{15}{44} & \frac{10}{11} & 0 & 0 & 0 \\ \frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72} & 0 & 0 \end{bmatrix},
$$

$$
A^{\{\mathrm{f},\mathrm{s},1\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{5M} & 0 & 0 & 0 & 0 & 0 \\ \frac{3(10M^3-90M^2+22M-1)}{20M(3M-4)} & -\frac{3(2M^3-6M^2+2M+3)}{4M(3M-4)} & 0 & 0 & 0 & 0 \\ 0 & \frac{3(10M^3-50M^2+116M-83)}{22M(3M-4)} & -\frac{30M^3+150M^2-282M+161}{22M(3M-4)} & 0 & 0 & 0 \\ \frac{11}{72M} & \frac{25}{72M} & \frac{25}{72M} & \frac{11}{72M} & 0 & 0 \end{bmatrix},
$$

$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72} & 0 & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} \frac{11(\lambda-1)}{72M} & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{11(\lambda-1)}{72M} & 0 \\ \frac{11\lambda}{72M} & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{f},\mathrm{s},\lambda\}} = \begin{bmatrix} \frac{25(\lambda-1)}{72M} & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{25\lambda}{72M} & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{s},\mathrm{f},1\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2M}{5} & 0 & 0 & 0 & 0 & 0 \\ -\frac{3}{20}M(5M-4) & \frac{3M^2}{4} & 0 & 0 & 0 & 0 \\ \frac{1}{44}M\left(56M^2-81M+44\right) & -\frac{5}{44}M^2(16M-13) & -\frac{5}{11}(M-3)M^2 & (M-1)M^2 & 0 & 0 \\ \frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72} & 0 & 0 \end{bmatrix},
$$

$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72} & 0 & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{5M} & 0 & 0 & 0 & 0 & 0 \\ \frac{3(10M^3-90M^2+22M-1)}{20M(3M-4)} & -\frac{3(2M^3-6M^2+2M+3)}{4M(3M-4)} & 0 & 0 & 0 & 0 \\ 0 & \frac{3(10M^3-50M^2+116M-83)}{22M(3M-4)} & -\frac{30M^3+150M^2-282M+161}{22M(3M-4)} & 0 & 0 & 0 \\ \frac{11}{72M} & \frac{11(\lambda-1)}{72M} & \frac{25}{72M} & \frac{25(\lambda-1)}{72M} & 0 & 0 \\ -\frac{450M^2+956\lambda M-497M-956\lambda+776}{450(M-1)M} & \frac{450M^2-506\lambda M+227M+506\lambda-506}{450(M-1)M} & & & & \\ -\frac{900M^3+1239\lambda M^2+2217M^2-2891\lambda M-971M+1652\lambda-1562}{660M(3M^2-7M+4)} & \frac{900M^3+561\lambda M^2-2937M^2-1309\lambda M+1777M+748\lambda+602}{660M(3M^2-7M+4)} & & & & \\ 0 & \frac{-90M^3+99\lambda M^2+197M^2-231\lambda M-205M+132\lambda+117}{22M(3M^2-7M+4)} & & & & \\ \frac{11\lambda}{72M} & \frac{25\lambda}{72M} & & & & \end{bmatrix},
$$

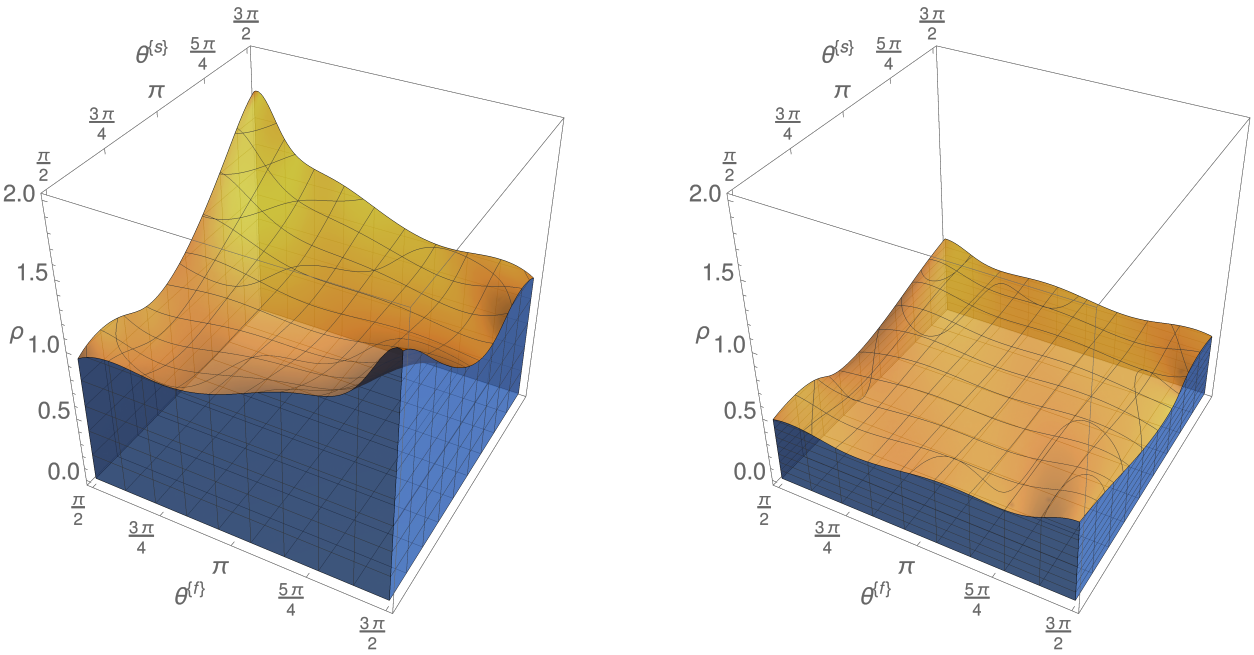
$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} \frac{11(\lambda-1)}{72M} & 0 \\ 0 & 0 \\ -\frac{11(\lambda-1)}{72M} & 0 \\ \frac{11\lambda}{72M} & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{f},\mathrm{s},\lambda\}} = \begin{bmatrix} \frac{25(\lambda-1)}{72M} & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{25\lambda}{72M} & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{11}{72} & 0 \end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
A^{\{\mathrm{s},\mathrm{f},\lambda\}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{3240M^3-8265\lambda M^2-7455M^2+1925\lambda M+8227M-1100\lambda-4696}{792M(3M^2-7M+4)} & -\frac{11(\lambda-1)}{72M} \\ \frac{25\lambda}{72M} & \frac{11\lambda}{72M} \end{bmatrix},
$$

$$
b^{\{\mathrm{f}\}} = b^{\{\mathrm{s}\}} = \begin{bmatrix} \frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72} & 0 \end{bmatrix}^T, \quad \widehat{b}^{\{\mathrm{f}\}} = \widehat{b}^{\{\mathrm{s}\}} = \begin{bmatrix} \frac{1251515}{8970912} & \frac{3710105}{8970912} & \frac{2519695}{8970912} & \frac{61105}{8970912} & \frac{119041}{747576} \end{bmatrix}^T.
$$



(a) $M = 2$. (b) $M = 4$.

Figure B.10: EX5-EX5 4(3)[A] stability regions.

B.1.11 MrGARK EX6-IM5 4(3)[A]

This explicit-implicit method uses a fast method from [38] and a slow method from [64]. The multirate scheme is stiffly accurate (3.21) in the slow partition.

$$
A^{\{f,f\}} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{32} & \frac{9}{32} & 0 & 0 & 0 & 0 & 0 \\
\frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & 0 & 0 & 0 & 0 \\
\frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & 0 & 0 & 0 \\
-\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{40} & 0 &
\end{bmatrix}, \quad
A^{\{s,s\}} = \begin{bmatrix}
\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
\frac{13}{20} & \frac{1}{4} & 0 & 0 & 0 & 0 \\
\frac{580}{1287} & -\frac{175}{5148} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{12698}{37375} & -\frac{201}{2990} & \frac{891}{11500} & \frac{1}{4} & 0 & \\
\frac{944}{1365} & -\frac{400}{819} & \frac{99}{35} & -\frac{575}{252} & \frac{1}{4} & \\
\end{bmatrix},
$$

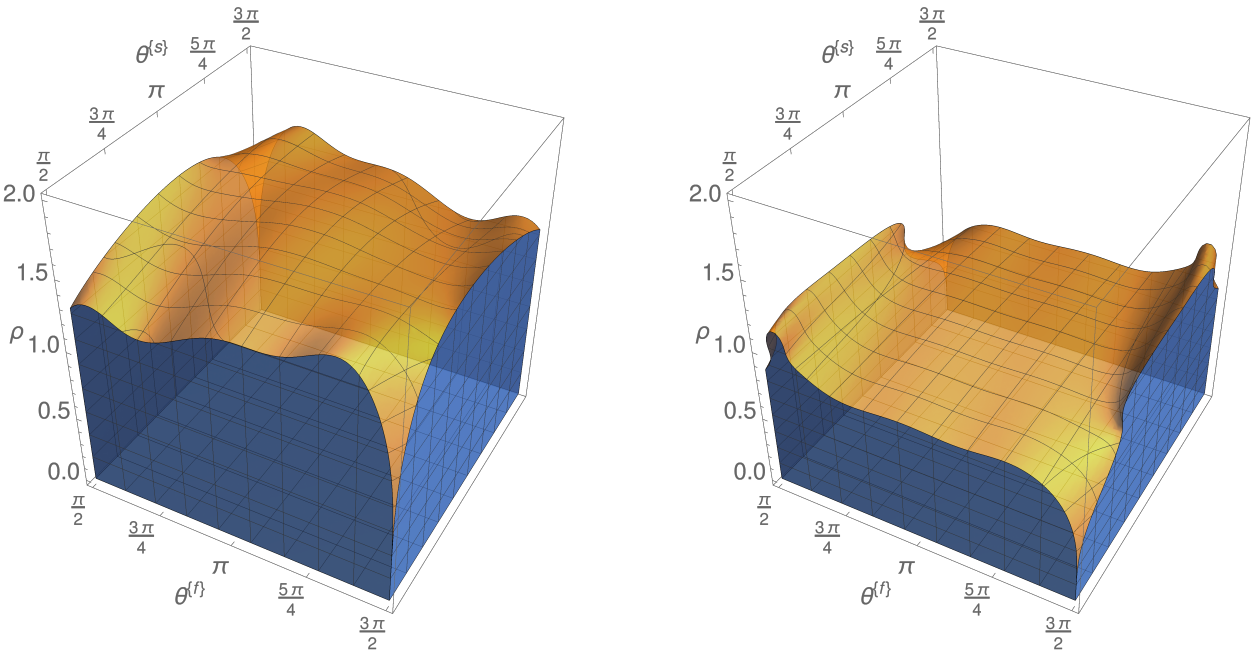
$$
A^{\{s,f,1\}} = \begin{bmatrix}
\frac{M}{4} & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{100}M(169M-90) & \frac{169M^2}{100} & 0 & 0 & 0 & 0 \\
-\frac{1}{198}M(155M-132) & \frac{155M^2}{198} & 0 & 0 & 0 & 0 \\
-\frac{1}{920}M(497M-552) & \frac{144M^2}{23} & -\frac{896M^2}{10925} & \frac{1183M^2}{87400} & 0 & 0 \\
\frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0
\end{bmatrix},
$$

$$
A^{\{f,s,\lambda\}} = \begin{bmatrix}
\frac{\lambda-1}{M} & 0 & 0 & 0 & 0 & 0 \\
\frac{4\lambda-3}{4M} & 0 & 0 & 0 & 0 & 0 \\
\frac{45M^3-90M^2+551\lambda M-335M+90\lambda-90}{416M^2} & -\frac{15(3M^3-6M^2+9\lambda M-5M+6\lambda-6)}{416M^2} & 0 & 0 & 0 & 0 \\
\frac{1440M^3-2880M^2+6517\lambda M-3709M-3960\lambda+3960}{2197M^2} & -\frac{60(24M^3-48M^2+72\lambda M-59M-66\lambda+66)}{2197M^2} & 0 & 0 & 0 & 0 \\
\frac{560M^3-362M^2+386\lambda M-529M-1155\lambda+1155}{273M^2} & -\frac{5(672M^3-2439M^2+4046\lambda M-2590M-1386\lambda+1386)}{1638M^2} & -\frac{33(11M-14\lambda+7)}{28M} & \frac{575(3M-2\lambda+1)}{252M} & 0 & \\
0 & 0 & \frac{160M^3-109M^2-300M+165}{32M^2} & \frac{-160M^3+109M^2+32\lambda M+284M-165}{32M^2} & 0 &
\end{bmatrix}, \quad \lambda = 1, \ldots, M,
$$

$$
A^{\{f,s,\lambda\}} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0
\end{bmatrix}, \quad \lambda = 2, \ldots, M,
$$

$$
b^{\{f\}} = \begin{bmatrix} \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0 \end{bmatrix}^T, \qquad
b^{\{s\}} = \begin{bmatrix} \frac{944}{1365} & -\frac{400}{819} & \frac{99}{35} & -\frac{575}{252} & \frac{1}{4} \end{bmatrix}^T,
$$

$$
\widehat{b}^{\{f\}} = \begin{bmatrix} \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55} \end{bmatrix}^T, \qquad
\widehat{b}^{\{s\}} = \begin{bmatrix} \frac{41911}{60060} & -\frac{83975}{144144} & \frac{3393}{1120} & -\frac{27025}{11088} & \frac{103}{352} \end{bmatrix}^T.
$$



(a) $M = 2$. (b) $M = 4$.

Figure B.11: EX6-IM5 4(3)[A] stability regions.

B.1.12 MrGARK IM6-EX4 4(2)[A]

This implicit-explicit method uses a slow method from [116] and is stiffly accurate (3.21) in the fast partition.

$$
A^{\{1,1\}} = \begin{bmatrix}
\frac{191}{1000} & 0 & 0 & 0 & 0 \\
\frac{209}{1000} & \frac{191}{1000} & 0 & 0 & 0 \\
\frac{8466728223}{12920014250} & -\frac{12729769579}{51686057000} & \frac{191}{1000} & 0 & 0 \\
\frac{1020936935125334486340705965597 71}{2228191313057444256311660020570 00} & -\frac{1724815120396388289389468 4614}{6809875653904169485750507341 25} & \frac{783289327941232988291717301}{1938400447113914098574736860} & \frac{191}{1000} & 0 \\
\frac{1837041228720545025852501951582239534326}{2195453146940870392428577778808091404375} & -\frac{1218153257338607745438284842754184 6123}{1762842727418687408857823582535587500 00} & \frac{4528991149246665992465958589058858289}{8624433917634487442857055656527718750 0} & \frac{837501605426861 87}{1000} & 0
\end{bmatrix}, \quad
A^{\{s,s\}} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
\frac{2}{5} & 0 & 0 & 0 & 0 \\
-\frac{3}{20} & \frac{3}{4} & 0 & 0 & 0 \\
\frac{19}{44} & -\frac{15}{44} & \frac{10}{11} & 0 & 0
\end{bmatrix},
$$

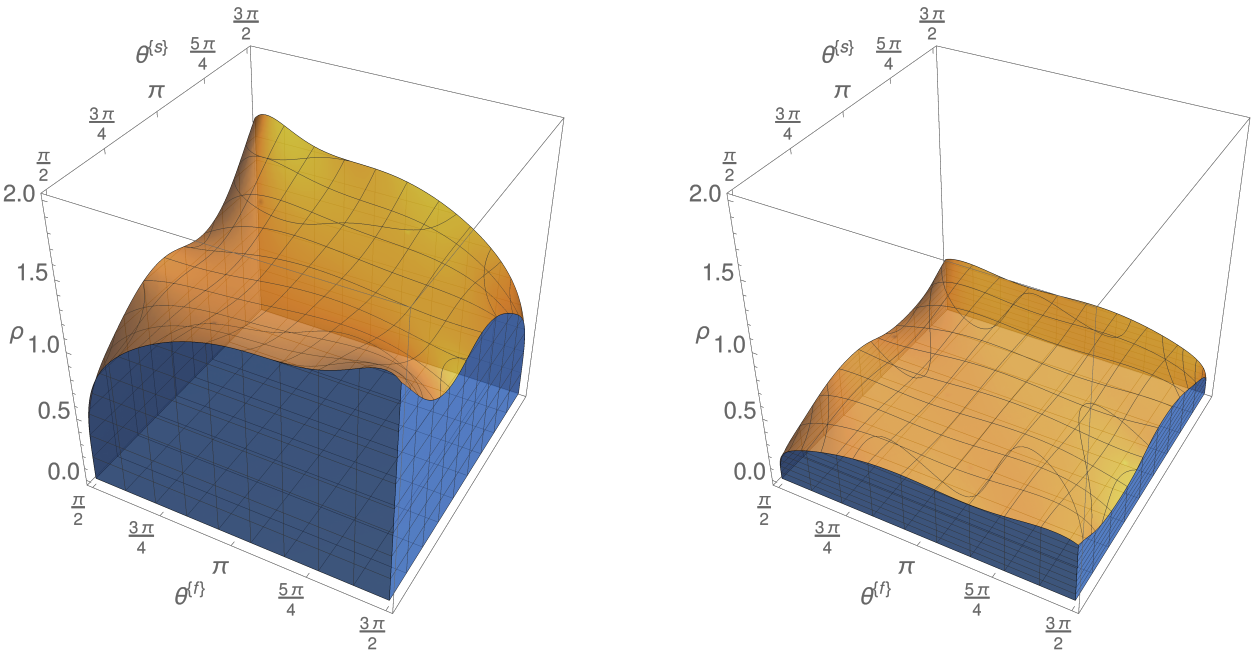
$$
A^{\{1,1\}} = \begin{bmatrix}
\frac{2288000}{4732539} & -\frac{2203}{14250} & \frac{247273}{613500} & \frac{30767}{152250} & \frac{191}{1000} \\
-\frac{1762842727418687408857823582535587500 0}{2195453146940870392428577778808091404375} & \frac{121815325733860774543828484275418 46123}{1762842727418687408857823582535587500 00} & \frac{4528991149246665992465958589058858289}{8624433917634487442857055656527718750 0} & \frac{837501605426861 87}{1000} & 0
\end{bmatrix}^T, \quad
\widehat{b}^{\{s\}} = \begin{bmatrix}
\frac{1}{5} & \frac{1}{4} & \frac{3}{8} & \frac{7}{40}
\end{bmatrix}^T.
$$

$$
A^{\{1,s,\lambda\}} = \begin{bmatrix}
\frac{1000\lambda-809}{1000M} & 0 & 0 & 0 \\
\frac{5\lambda-3}{5M} & 0 & 0 & 0 \\
\frac{5\lambda-2}{5M} & 0 & 0 & 0 \\
\frac{5\lambda-1}{5M} & 0 & 0 & 0 \\
\frac{\lambda}{M} & 0 & 0 & 0 \\
\frac{\lambda}{M} & 0 & 0 & 0
\end{bmatrix}, \quad \lambda = 1, \ldots, M-1,
$$

$$
A^{\{1,s,M\}} = \begin{bmatrix}
\frac{1000M-809}{1000M} & 0 & 0 & 0 \\
\frac{1138019507045 3M^3-184088956711 88M^2+1447705508128 2M-501686712 0000}{8361445200000M} & \frac{209(5445069411 7M^2-88080840532M+29261291298)}{8361445200000M} & 0 & 0 \\
\frac{-4572847560966352 51M^3-42117704549104000 0M^2+701106234145018506M-206755963893960000}{5168899097349000000M} & \frac{409(1118055638377 39M^2+10297727273619 56M-45040666114 9434)}{5168899097349000000M} & 0 & 0 \\
\frac{3132523904037186017M^3-45723258000177292 32M^2+26520877959097739 8M-514513168936 80000}{25725658446840000 0M} & -\frac{203(2350256923212 739M^2-21865549138804 4M-533759817986984)}{25725658446840000} & \frac{203(8850000M^2+70000M-628199)}{282071856} & 0 \\
0 & -\frac{885000M^2+885000M+831041}{859500} & \frac{5900000M^2-5900000M-114791}{573000} & \frac{2101}{9000} \\
\frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72}
\end{bmatrix},
$$

$$
A^{\{s,f,\lambda\}} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
\frac{2}{5} & 0 & 0 & 0 & 0 \\
\frac{3(5000\lambda-409)}{1045} & -\frac{6(250M-309)}{1045} & 0 & 0 & 0 \\
\frac{5470086378426598 63-8862817802551614 44M}{1520259952073533729} & \frac{3(2856036493343421M-3906629787402737)}{2288803570033750} & -\frac{741819(1303514627M-4489322119)}{2239523110391875} & -\frac{8391(1856541237 9M-249370273 63)}{2020996627735 0} & 0
\end{bmatrix}, \quad \lambda = 1, \ldots, M,
$$

$$
b^{\{f\}} = \begin{bmatrix}
\frac{2288000}{4732539} & -\frac{2203}{14250} & \frac{247273}{613500} & \frac{30767}{152250} & -\frac{1}{8} & \frac{191}{1000}
\end{bmatrix}^T, \quad
b^{\{s\}} = \begin{bmatrix}
\frac{11}{72} & \frac{25}{72} & \frac{25}{72} & \frac{11}{72}
\end{bmatrix}^T,
$$



(a) $M = 2$. (b) $M = 4$.

FIGURE B.12: IM6-EX4 4(2)[A] stability regions.

B.2 ADI-GLMs

This section includes the newly developed ADI-DIMSIMs of orders two, three, and four. MATLAB files containing these coefficients are also available in [dataset] [103].

B.2.1 ADI-DIMSIM2

We use an L-stable implicit base method from [16] for ADI-DIMSIM2.

$$\mathbf{A}^{\{I\}} = \begin{bmatrix} \frac{2-\sqrt{2}}{2} & 0 \\ \frac{2(\sqrt{2}+3)}{7} & \frac{2-\sqrt{2}}{2} \end{bmatrix}, \qquad \mathbf{B}^{\{I\}} = \begin{bmatrix} \frac{73-34\sqrt{2}}{28} & \frac{4\sqrt{2}-5}{4} \\ \frac{3(29-16\sqrt{2})}{28} & \frac{34\sqrt{2}-45}{28} \end{bmatrix},$$

$$\mathbf{W}^{\{I\}} = \begin{bmatrix} 1 & \frac{\sqrt{2}-2}{2} & 0 \\ 1 & \frac{3(\sqrt{2}-4)}{14} & \frac{\sqrt{2}-1}{2} \end{bmatrix}, \qquad \mathbf{A}^{\{E\}} = \begin{bmatrix} 0 & 0 \\ \frac{3}{2} & 0 \end{bmatrix},$$

$$\mathbf{B}^{\{E\}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{3-\sqrt{2}}{4} \\ \frac{\sqrt{2}-1}{2} & \frac{3-\sqrt{2}}{4} \end{bmatrix}, \qquad \mathbf{W}^{\{E\}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

$$v = \begin{bmatrix} \frac{3-\sqrt{2}}{2} & \frac{\sqrt{2}-1}{2} \end{bmatrix}^T, \qquad \mathbf{c} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T.$$

B.2.2 ADI-DIMSIM3

We use an L-stable implicit base method from [20] for ADI-DIMSIM3. The following coefficients are rational approximations to the exact coefficients accurate to 24 digits.

$$\mathbf{A}^{\{I\}} = \begin{bmatrix} \frac{129981159316}{298213221025} & 0 & 0 \\ \frac{472981046840}{1888035733227} & \frac{129981159316}{298213221025} & 0 \\ -\frac{408860438935}{337456558734} & \frac{1049716501919}{1048380236594} & \frac{129981159316}{298213221025} \end{bmatrix},$$

$$\mathbf{B}^{\{I\}} = \begin{bmatrix} \frac{818629988268}{981817092145} & \frac{735879558291}{1139134361459} & -\frac{96693387431}{306159262034} \\ \frac{435713380671}{718693545019} & \frac{3397277300866}{2639826970205} & -\frac{581689679739}{1212506039656} \\ -\frac{164008995335}{531777165056} & \frac{3204278525979}{842472621931} & -\frac{1170634530631}{1044535547981} \end{bmatrix},$$

$$\mathbf{W}^{\{I\}} = \begin{bmatrix} 1 & -\frac{129981159316}{298213221025} & 0 & 0 \\ 1 & -\frac{63231801579}{339260252164} & -\frac{94226735668}{1013918320559} & -\frac{50172116077}{1490999795865} \\ 1 & \frac{1224205243956}{1580735023225} & -\frac{377260820095}{864278390147} & -\frac{145496067686}{824686465859} \end{bmatrix},$$

$$\mathbf{A}^{\{E\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{692830401049}{1119419041371} & 0 & 0 \\ -\frac{974910195245}{1036334372568} & \frac{1458124485343}{1218848111125} & 0 \end{bmatrix},$$

$$\mathbf{B}^{\{E\}} = \begin{bmatrix} \frac{274198327012}{348784765929} & \frac{335124252337}{1242427076379} & \frac{256046237035}{1044616400532} \\ \frac{2367946890051}{2381074405894} & -\frac{395462379375}{996294720374} & \frac{391448928279}{669688356392} \\ \frac{1211513153203}{1601457627995} & \frac{473388990672}{901108379101} & \frac{1335987676745}{1749669440649} \end{bmatrix},$$

$$\mathbf{W}^{\{E\}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -\frac{105007291910}{883010702197} & \frac{1}{8} & \frac{1}{48} \\ 1 & \frac{6500435948486}{8732264247243} & -\frac{119638187109}{1218848111125} & \frac{25266119777}{1475180609484} \end{bmatrix},$$

$$v = \begin{bmatrix} \frac{1611220452657}{2918396719813} & \frac{626900045900}{853091602939} & -\frac{165394139815}{576391394057} \end{bmatrix}^T,$$

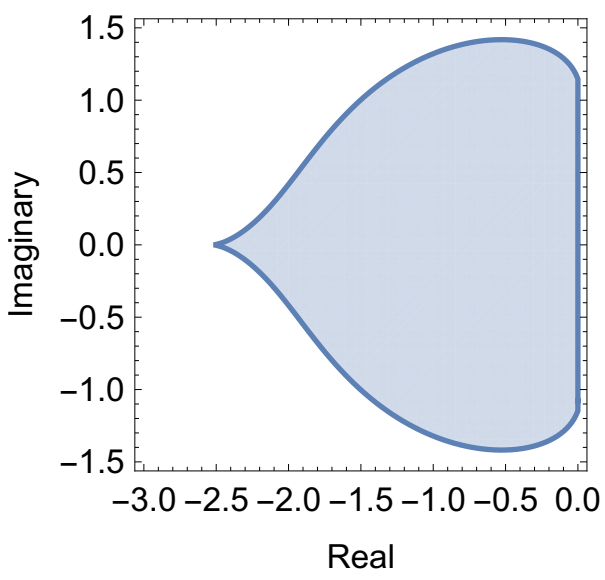
$$\mathbf{c} = \begin{bmatrix} 0 & \frac{1}{2} & 1 \end{bmatrix}^T.$$

$$
\mathbf{A}^{\{I\}} = \begin{bmatrix} \frac{2}{5} & 0 & 0 & 0 & 0 \\ \frac{1}{155} & \frac{2}{5} & 0 & 0 & 0 \\ -\frac{3}{127} & \frac{31}{72} & \frac{2}{5} & 0 & 0 \\ \frac{6}{139} & \frac{12}{19} & \frac{29}{95} & \frac{2}{5} & 0 \end{bmatrix}, \quad \mathbf{B}^{\{I\}} = \begin{bmatrix} \frac{256402750338 59}{233564187988800} & \frac{40516968 7}{5406153 60} & \frac{10897727 29}{8109230400} & \frac{704451 77}{4268016 00} \\ \frac{898704267307 79}{233564187988800} & -\frac{5459599 87}{1621846080} & \frac{13906861889}{8109230400} & -\frac{1223893451}{441028320 0} \\ \frac{292292722987739}{233564187988800} & -\frac{5722388059}{1621846080} & \frac{525192608 1}{901025600} & -\frac{1156463340 41}{5420380320 0} \\ \frac{125916292681628 81}{443771957178720 0} & -\frac{4936252337}{5406153 60} & \frac{10261520332 9}{8109230400} & -\frac{5841129112303}{1127183025600} \end{bmatrix},
$$

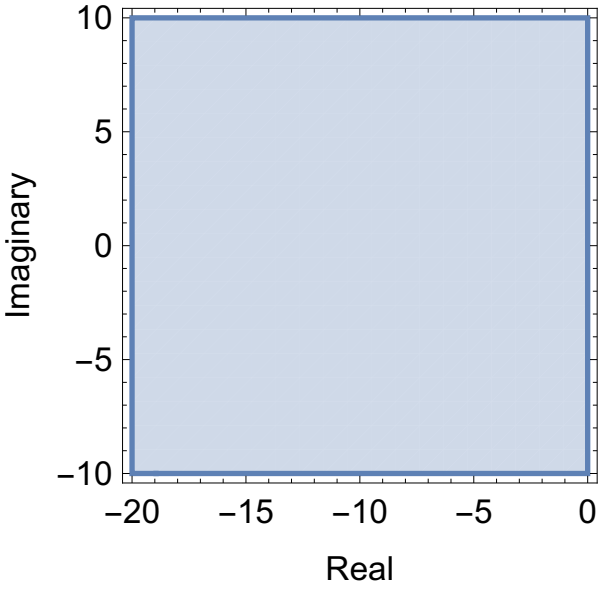
$$
\mathbf{W}^{\{I\}} = \begin{bmatrix} 1 & -\frac{2}{5} & 0 & 0 & 0 \\ 1 & -\frac{34}{465} & -\frac{7}{90} & -\frac{13}{810} & -\frac{19}{9720} \\ 1 & -\frac{6413}{45720} & -\frac{203}{1080} & -\frac{137}{2160} & -\frac{827}{58320} \\ 1 & -\frac{5018}{13205} & -\frac{179}{570} & -\frac{233}{1710} & -\frac{2707}{61560} \end{bmatrix}, \quad \mathbf{A}^{\{E\}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{768}{7129} & 0 & 0 & 0 & 0 \\ \frac{2699}{8714} & \frac{4969}{11444} & 0 & 0 & 0 \\ \frac{2629}{3049} & \frac{2643}{20780} & \frac{11707}{22938} & 0 & 0 \end{bmatrix},
$$

$$
\mathbf{B}^{\{E\}} = \begin{bmatrix} \frac{98875144419778 75393}{80840619606081 11040} & \frac{75125403707867}{12687014733264 00} & \frac{20004128690 9}{3263325033 60} & -\frac{592474 7}{8536032 0} \\ \frac{88770066969018 61513}{80840619606081 11040} & \frac{72709699416 7267}{12687014733264 00} & -\frac{6737007 0011}{3263325033 60} & \frac{11901930035 9}{2028445737 60} \\ \frac{17771936994130966829533}{23128501269299805685440} & \frac{24906774287 7763}{14096683036 9600} & -\frac{5199371826743 17}{3112124307 04320} & \frac{94067697106 4501}{10640485696 40640} \\ \frac{85664932449116727594047 29}{32110678261545596037650880} & \frac{17071987325364576461}{48502457325268272 00} & -\frac{2570984966412689}{67811894198208} & \frac{27515934006270 7361}{20675846541033 6192} \end{bmatrix},
$$

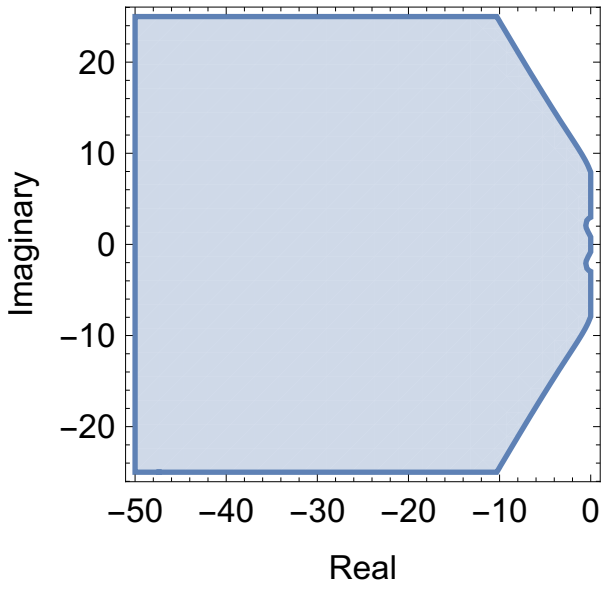
$$
\mathbf{W}^{\{E\}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & \frac{4825}{21387} & \frac{1}{18} & \frac{1}{162} & \frac{1}{1944} \\ 1 & -\frac{11557817}{149584524} & \frac{7981}{102996} & \frac{46831}{1853928} & \frac{30869}{5561784} \\ 1 & -\frac{36319351315 3}{72665542518 0} & \frac{8390470 3}{7149774 60} & \frac{19812197 3}{4289864 760} & \frac{30265043 9}{19304391420} \end{bmatrix}, \quad v = \begin{bmatrix} \frac{3}{40} & -\frac{77}{277} & -\frac{41}{107} & \frac{1880483}{1185560} \end{bmatrix}^T, \quad \mathbf{c} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix}^T.
$$



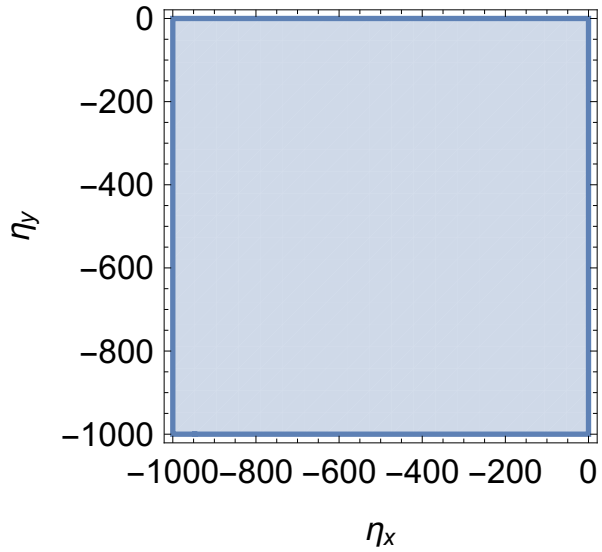
(a) $\mathcal{S}^{\{E\}}$ stability region



(b) $\mathcal{S}^{\{I\}}$ stability region

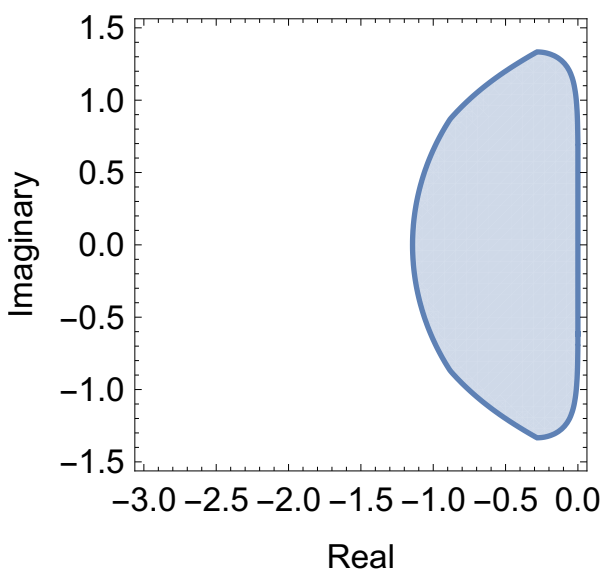


(c) $\mathcal{S}_{\text{Cplx}}$ stability region with $\alpha = 60°$

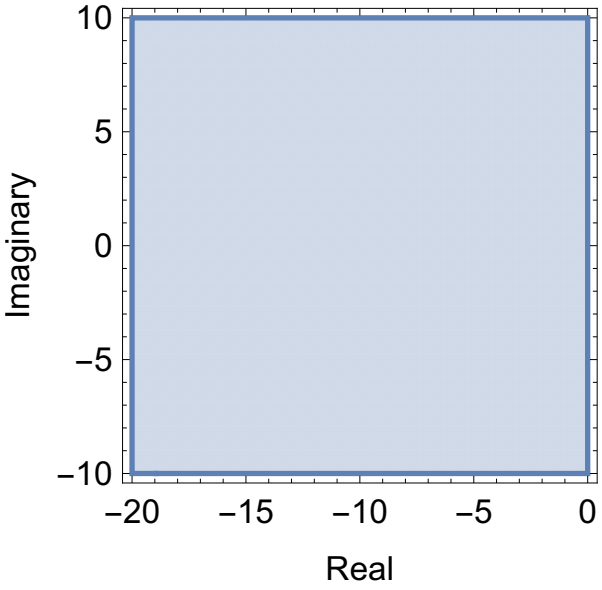


(d) $\mathcal{S}_{\text{Real}}$ stability region

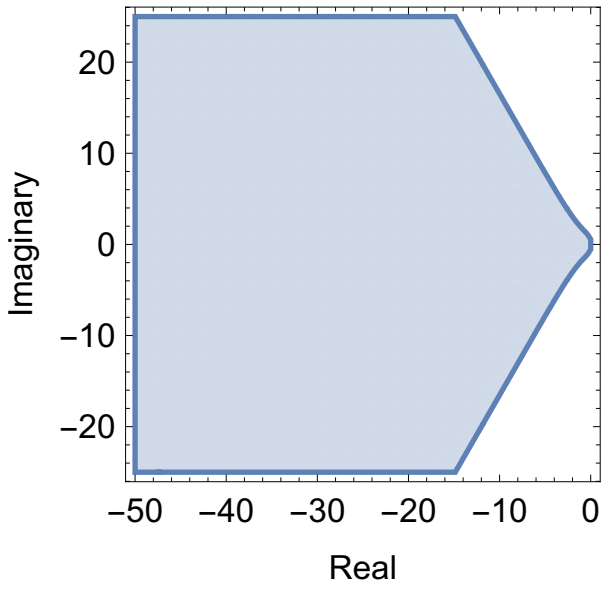
Figure B.13: Stabilty plots for linearly implicit GLM2



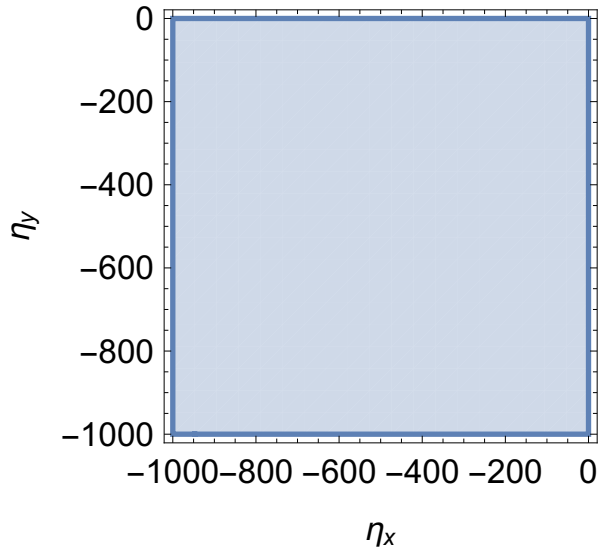
(a) $\mathcal{S}^{\{E\}}$ stability region



(b) $\mathcal{S}^{\{I\}}$ stability region



(c) $\mathcal{S}_{\text{Cplx}}$ stability region with $\alpha = 55°$



(d) $\mathcal{S}_{\text{Real}}$ stability region

FIGURE B.14: Stabilty plots for linearly implicit GLM3

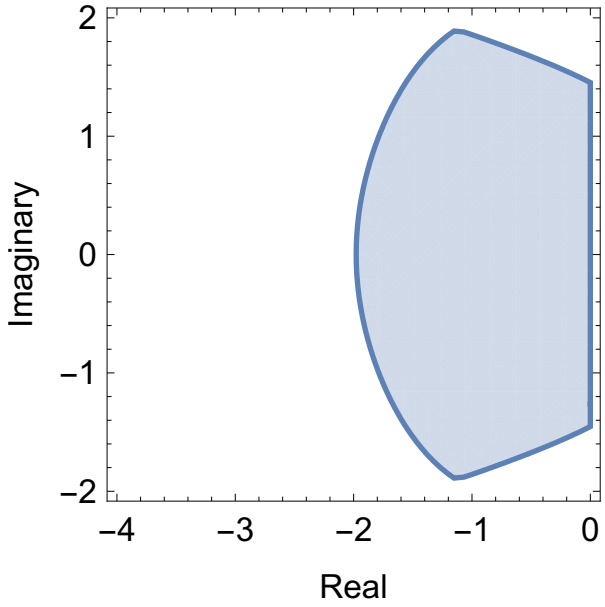
B.2.3 ADI-DIMSIM4

B.3 Stability of ADI-GLMs

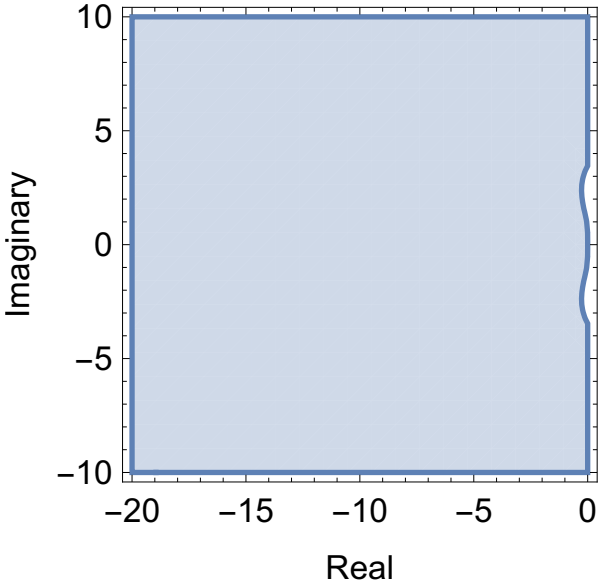
B.4 linearly implicit GLM methods

LIMSIM 3

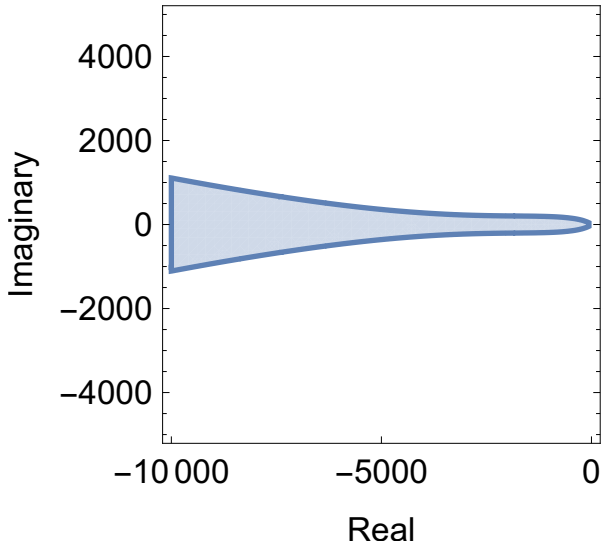
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \end{pmatrix}$$



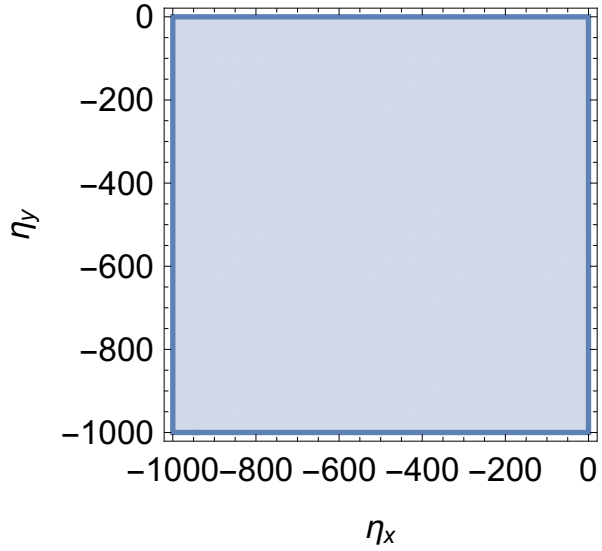
(a) $\mathcal{S}^{\{E\}}$ stability region



(b) $\mathcal{S}^{\{I\}}$ stability region with $\alpha = 83°$



(c) $\mathcal{S}_{\text{Cplx}}$ with $\alpha = 3.7°$



(d) $\mathcal{S}_{\text{Real}}$

FIGURE B.15: Stability plots for linearly implicit GLM4

LIMSIM 4

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{3367}{2911232} & 0 & 0 & 0 & 0 \\ \frac{1533}{20320} & -\frac{19423}{70722} & 0 & 0 & 0 \\ \frac{1234803}{7851008} & -\frac{1849}{3200} & -\frac{607}{4996} & 0 & 0 \\ -\frac{85487}{260640} & \frac{339968}{183699} & -\frac{10059}{5756} & \frac{12238}{30819} & 0 \end{pmatrix},$$

$$\mathbf{U} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{2180057}{2911232} & \frac{815417}{1455616} & \frac{1218075}{2911232} & \frac{56729}{181952} \\ 1 & \frac{502397027}{718535520} & \frac{61203929}{119755920} & \frac{7227751}{19959320} & \frac{17898017}{79837280} \\ 1 & \frac{194164388861}{245147724800} & \frac{90233588327}{122573862400} & \frac{299064458359}{490295449600} & \frac{201312980409}{490295449600} \\ 1 & \frac{65099936835498259}{78643608352094880} & \frac{16868249117333519}{39321804176047440} & \frac{2547146550312569}{26214536117364960} & \frac{745125993133019}{19660902088023720} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} -\frac{29}{96} & \frac{7}{9} & -\frac{1}{4} & \frac{1}{3} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 \\ \frac{13}{6} & -8 & 6 & -\frac{8}{3} & 2 \\ \frac{43}{9} & -\frac{64}{3} & \frac{64}{3} & -\frac{64}{9} & \frac{8}{3} \\ \frac{21}{4} & -\frac{800}{27} & 40 & -\frac{32}{3} & \frac{8}{3} \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} 1 & \frac{55}{288} & \frac{1}{48} & -\frac{1}{32} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & -\frac{1}{3} & -\frac{2}{3} & 0 & 0 \\ 0 & -\frac{823}{108} & -\frac{109}{18} & -\frac{7}{4} & 0 \end{pmatrix},$$

$$\mathbf{\Gamma} = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 \\ -\frac{313}{11372} & \frac{1}{4} & 0 & 0 & 0 \\ -\frac{449}{10160} & \frac{1285}{7858} & \frac{1}{4} & 0 & 0 \\ -\frac{240}{7667} & \frac{1547}{9600} & \frac{4961}{9992} & \frac{1}{4} & 0 \\ \frac{211}{8145} & -\frac{21899}{20411} & \frac{2155}{1439} & -\frac{655}{10273} & \frac{1}{4} \end{pmatrix},$$

$$\mathbf{\Psi} = \begin{pmatrix} 0 & -\frac{1}{4} & -\frac{1}{2} & -\frac{3}{4} & -1 \\ 0 & -\frac{1265}{5686} & -\frac{7277}{22744} & -\frac{61737}{181952} & -\frac{56729}{181952} \\ 0 & -\frac{14743339}{39918640} & -\frac{8121559}{19959320} & -\frac{26416089}{79837280} & -\frac{17898017}{79837280} \\ 0 & -\frac{80562288001}{91930396800} & -\frac{49066928401}{61286931200} & -\frac{292840004409}{490295449600} & -\frac{201312980409}{490295449600} \\ 0 & -\frac{6260149023115573}{9830451044011860} & -\frac{2006130607958233}{4915225522005930} & -\frac{105198462624382}{819204253667655} & -\frac{745125993133019}{19660902088023720} \end{pmatrix},$$

$$\mathbf{c} = \begin{pmatrix} 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 1 \end{pmatrix}^T.$$

LIMSIM 5

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{580528}{7115625} & 0 & 0 & 0 & 0 & 0 \\ -\frac{7453409}{27040625} & \frac{131337}{229024} & 0 & 0 & 0 & 0 \\ -\frac{8759}{21875} & \frac{982283}{915824} & -\frac{141571}{589095} & 0 & 0 & 0 \\ -\frac{85075130577749212485}{116001467401995555072} & \frac{186780365}{60984144} & -\frac{2434553}{476775} & \frac{213337}{126738} & 0 & 0 \\ \frac{7351771}{7102404} & -\frac{86029409}{19994688} & \frac{85782343}{11488878} & -\frac{1015549}{353484} & -\frac{179407}{198324} & 0 \end{pmatrix},$$

$$\mathbf{U} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{6273028}{7115625} & \frac{5715056}{7115625} & \frac{1794928}{2371875} & \frac{5236672}{7115625} & \frac{5234288}{7115625} \\ 1 & \frac{29745156058216120372147}{98437329776058814169088} & -\frac{285393113}{45536412500} & -\frac{1323706191}{22768206250} & \frac{38636113}{669653125} & \frac{3204547204}{11384103125} \\ 1 & -\frac{11442903308130768694988473}{35942678286169463901388800 0} & -\frac{38815470851}{83134143750} & -\frac{16010970402221724302637301}{29952231905141219917824000} & -\frac{23217453584028088827078569}{64183354082445471252480000} & -\frac{32116971744795597901645 99}{11232086964427957469184000 0} \\ 1 & \frac{187018763188389454083940198896 23}{144569029262773130719798493184 00} & \frac{200557999599611479634752826034551}{144569029262773130719798493184000} & \frac{6231472349558239235107365592399 1}{6023709552615547113324937216000 0} & \frac{4651879541232367359765539590629311}{72284514631386565359899246592000000} & \frac{1764357676564689555720261483828751}{36142257315693282679949623296000 00} \\ 1 & \frac{1544065252264913607084 7}{2668929364121991061190 4} & -\frac{810158884793894222196 1}{166808085257624441324 40} & -\frac{584681014476309216595 1}{139006737714687034437 00} & -\frac{800925190567442308351}{5212752664300763791387 5} & \frac{180204893529420689141 27}{104255053286015275827 750} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \frac{689}{972} & -\frac{22225}{5184} & \frac{1175}{162} & -\frac{1025}{324} & -\frac{25}{36} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{101}{24} & -\frac{25}{2} & \frac{25}{2} & -\frac{25}{3} & \frac{25}{8} & \frac{3}{2} \\ \frac{487}{36} & -\frac{4775}{96} & \frac{3475}{54} & -\frac{2825}{72} & \frac{25}{4} & \frac{3}{2} \\ \frac{589}{32} & -\frac{19575}{256} & \frac{16175}{144} & -\frac{2225}{32} & \frac{225}{32} & \frac{9}{8} \\ 15 & -\frac{36095}{512} & \frac{355}{3} & -\frac{1215}{16} & \frac{45}{8} & \frac{27}{40} \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} 1 & \frac{13219}{15552} & -\frac{233}{1944} & -\frac{41}{324} & \frac{43}{243} & \frac{68}{243} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{2891}{864} & \frac{43}{36} & 0 & 0 & 0 \\ 0 & \frac{16375}{2304} & \frac{125}{96} & -\frac{9}{16} & 0 & 0 \\ 0 & \frac{52241}{7680} & -\frac{657}{320} & -\frac{591}{160} & -\frac{13}{10} & 0 \end{pmatrix},$$

$$\mathbf{\Gamma} = \begin{pmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ \frac{16}{3795} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{2077}{25959} & -\frac{2986}{7157} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{869}{7875} & -\frac{9193}{13209} & \frac{2339}{15105} & \frac{1}{3} & 0 & 0 \\ -\frac{3670}{10303} & \frac{7685}{10859} & \frac{259}{4075} & \frac{3399}{4694} & \frac{1}{3} & 0 \\ -\frac{2384}{7307} & \frac{178}{11571} & -\frac{15139}{70919} & -\frac{5707}{19638} & \frac{6947}{33054} & \frac{1}{3} \end{pmatrix},$$

$$\mathbf{\Psi} = \begin{pmatrix} 0 & -\frac{1}{3} & -\frac{2}{3} & -1 & -\frac{4}{3} & -\frac{5}{3} \\ 0 & -\frac{427}{1265} & -\frac{2056}{3795} & -\frac{4128}{6325} & -\frac{66368}{94875} & -\frac{22256}{31625} \\ 0 & \frac{42292}{10928739} & \frac{5875328}{54643695} & \frac{3661462}{18214565} & \frac{19801184}{80358375} & \frac{325676654}{1366092375} \\ 0 & \frac{486000268}{4988048625} & \frac{2196534356}{4988048625} & \frac{1127245114}{1662682875} & \frac{2724956816}{3562891875} & \frac{18220866694}{24940243125} \\ 0 & -\frac{9453732024817201}{6420154597409550} & -\frac{19401943990497478}{16050386493523875} & -\frac{19966598832966692}{26750644155873125} & -\frac{110552146569725504}{401259662338096875} & \frac{78289619977683922}{401259662338096875} \\ 0 & \frac{4192652688585018622}{1544519307940967049 3} & \frac{28251363011285089441}{77225965397048352465} & \frac{75699523449532751519}{257419884656827841550} & \frac{371301397189028640638}{1930649134926208811625} & \frac{413102379314756288699}{3861298269852417623250} \end{pmatrix},$$

$$\mathbf{c} = \begin{pmatrix} 1 & \frac{4}{5} & \frac{50665495807918080047}{84442493013196800000} & \frac{35128077093489869282 77}{87820192733724672000 00} & \frac{105384231280469641}{526921156402348032} & 1 \end{pmatrix}.$$

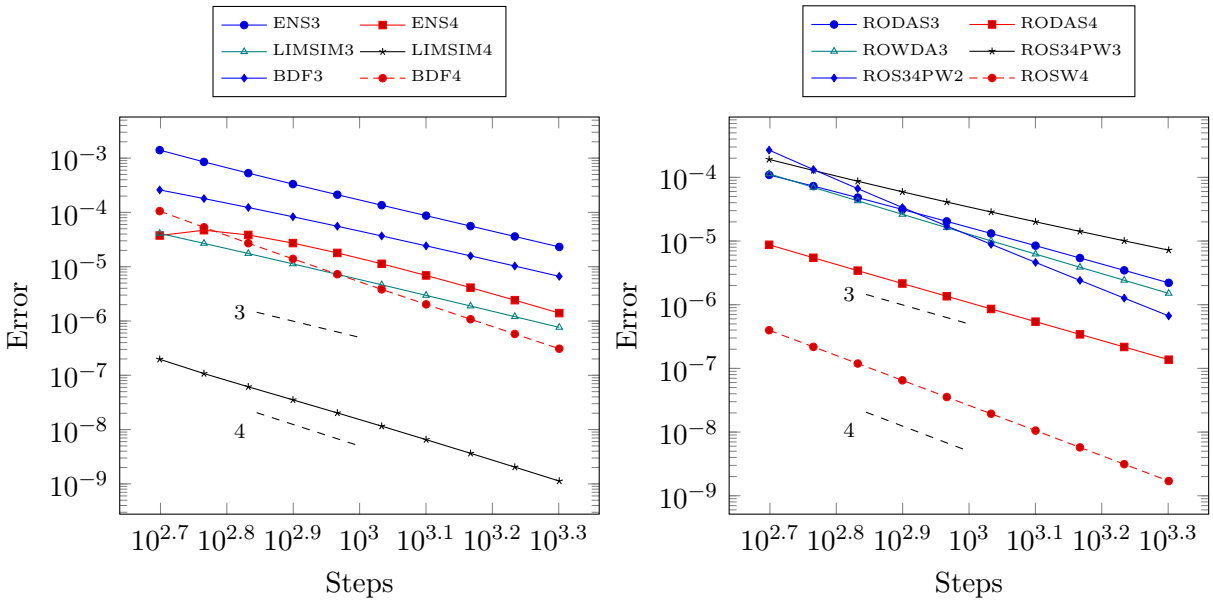


FIGURE B.16: Convergence results for the Brusselator problem

B.5 Confirming the order of the methods on the Brusselator problem