

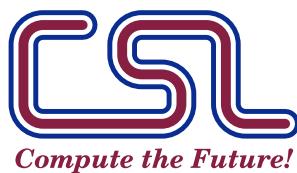
Computational Science
Laboratory Report CSL-TR-21-3
December 22, 2021

S. González-Pinto, D. Hernández-Abreu,
M. S. Pérez-Rodríguez, A. Sarshar, S.
Roberts, A. Sandu

*“A unified formulation of
splitting-based implicit time integration
schemes”*

Computational Science Laboratory
“Compute the Future!”

Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
Phone: (540) 231-2193
Fax: (540) 231-6075
Email: sandu@cs.vt.edu
Web: <https://csl.cs.vt.edu>



A unified formulation of splitting-based implicit time integration schemes

Severiano González-Pinto^{a,1}, Domingo Hernández-Abreu^{a,1}, María S. Pérez-Rodríguez^{a,1}, Arash Sarshar^{b,2}, Steven Roberts^{b,2,*}, Adrian Sandu^{b,2}

^a*Departamento de Análisis Matemático, Universidad de La Laguna, 38200-La Laguna, Spain*

^b*Computational Science Laboratory, Department of Computer Science, Virginia Tech, Blacksburg, VA 24060*

Abstract

Splitting-based time integration approaches such as fractional step, alternating direction implicit, operator splitting, and locally one dimensional methods partition the system of interest into components, and solve individual components implicitly in a cost-effective way. This work proposes a unified formulation of splitting time integration schemes in the framework of general-structure additive Runge–Kutta (GARK) methods. Specifically, we develop implicit-implicit (IMIM) GARK schemes, provide the order conditions for this class, and explain their application to partitioned systems of ordinary differential equations. We show that classical splitting methods belong to the IMIM GARK family, and therefore can be studied in this unified framework. New IMIM-GARK splitting methods are developed and tested using parabolic systems.

Keywords: general-structure additive Runge–Kutta methods, alternating direction implicit, implicit-explicit, implicit-implicit methods

2020 MSC: 65L05, 65L07, 65L020

*Corresponding author

Email addresses: spinto@ull.edu.es (Severiano González-Pinto), dhabreu@ull.edu.es (Domingo Hernández-Abreu), sperezr@ull.edu.es (María S. Pérez-Rodríguez), sarshar@vt.edu (Arash Sarshar), steven94@vt.edu (Steven Roberts), sandu@cs.vt.edu (Adrian Sandu)

¹The work of these authors has been partially supported by University of La Laguna, Canary Islands.

²The work of Sandu, Sarshar, and Roberts has been supported in part by NSF through awards NSF ACI-1709727, NSF CDS&E-MSS 1953113, DOE ASCR DE-SC0021313, and by the Computational Science Laboratory at Virginia Tech.

1. Introduction

Many applications require the solution of partitioned initial value problems

$$y' = f(t, y) = \sum_{m=1}^N f^{\{m\}}(t, y), \quad y(t_0) = y_0 \in \mathbb{R}^d, \quad (1)$$

where the right-hand side function $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is additively split into N parts. The partitioning may be informed by spatial dimensions, physical processes, stiffness, linearity, time-scales, or computational costs. Additive partitioning also includes the special case of component partitioning where the state vector y is split into disjoint sets [49].

The earliest approaches to efficiently tackle partitioned problems (1) involve solving the individual component systems one at a time, in an alternating fashion. This idea led to several closely connected families of schemes including fractional step methods, alternating direction implicit (ADI) methods, operator splitting methods, and locally one dimensional (LOD) methods. An early survey of fractional step methods is given in Yanenko's monograph [62]. ADI methods for the heat equation were proposed by Douglas [18], Douglas and Rachford [19], and Peaceman and Rachford [40]; the idea is to solve multidimensional partial differential equation via a sequence of one-dimensional problems, each aligned to a different spatial dimension. Recent theoretical results on stability and convergence for this kind of methods are obtained in [20, 21, 25]. These splittings are further explored in [57–59] for efficient time-stepping of Maxwell's equations with special interest in reducing right-hand side function evaluations. Bujanda and Jorge [8, 9] propose fractional step Runge–Kutta methods (FSRK) for solving multidimensional parabolic PDEs, by means of linearly implicit time integration processes. They extend these methods to semi-linear parabolic problems in an IMEX/ARK fashion [10, 12] where a FSRK implicit method used to solve the linear non-homogeneous terms is paired with an explicit Runge–Kutta method for solving nonlinear terms. An extension of ADI strategies for parabolic problems to general linear methods is discussed in [51]. The traditional first order operator splitting method [62] was extended to a symmetric, second order splitting approach by Strang [56], and to higher order splitting approaches by Yoshida [63].

The development of Runge–Kutta methods that are tailored to the partitioned system (1) started with the early work of Rice [43]. Hairer [29] developed the concept of P-trees and laid the foundation for the modern order conditions theory for partitioned Runge–Kutta methods. The investigation of practical partitioned Runge–Kutta methods with implicit and explicit components was revitalized by the work of Ascher, Ruuth, and Spiteri [2]. The Additive Runge–Kutta framework [16, 42] has served as the foundation for numerous later works on implicit-explicit Runge–Kutta methods [5, 6, 35]. Some partitioning strategies have been discussed by Weiner et al. [61]. IMEX extrapolation methods [13–15, 52] and IMEX general linear methods [7, 47, 53–55, 65, 67] are some of the latest examples in this area. Sandu and Günther have proposed

a generalized-structure for additively partitioned Runge–Kutta (GARK) methods in [49]. This framework provides a unifying theory for the creation and analysis of partitioned Runge–Kutta methods in various applications such as multirate [28, 50], multirate infinitesimal [45, 46], differential-algebraic [27], and linearly implicit [48] integration methods; research implementations are available in [3, 64]. An alternative approach to efficiently solving (1) is to employ splitting at a linear algebra level, rather than splitting the system or its discretization; this led to approximate matrix factorization (AMF) family of schemes [4, 20, 22–24, 26, 66].

This paper proposes a unified formulation of implicit integration schemes based on splitting (fractional step, alternating direction implicit, operator splitting, and locally one dimensional methods) in the framework of general-structure additive Runge–Kutta (GARK) methods. The framework allows for the construction of new schemes of higher classical order suited for serial and parallel partitioned integration.

The remainder of the paper is organized as follows. In Section 2, practical structures for implicit integration are introduced, and section 3 provides insights into the stability of these methods. Sections 4 to 6 are dedicated to deriving GARK formulations of classical ADI schemes, operator splitting schemes and fractional step Runge–Kutta methods. The order condition theory for implicit-implicit GARK methods is discussed in section 7, and two new methods are derived. Finally we present some numerical tests for the new methods in Section 8.

2. Implicit-implicit GARK schemes

We start by considering an N -way partitioned GARK method [49] to solve (1):

$$Y_i^{\{q\}} = y_n + h \sum_{m=1}^N \sum_{j=1}^{s^{\{m\}}} a_{i,j}^{\{q,m\}} f_j^{\{m\}}, \quad (2a)$$

$$q = 1, \dots, N, \quad i = 1, \dots, s^{\{q\}},$$

$$y_{n+1} = y_n + h \sum_{q=1}^N \sum_{i=1}^{s^{\{q\}}} b_i^{\{q\}} f_i^{\{q\}}, \quad (2b)$$

where we denote

$$f_j^{\{m\}} := f^{\{m\}} \left(t_n + c_j^{\{m\}} h, Y_j^{\{m\}} \right), \quad j = 1, \dots, s^{\{m\}}.$$

The GARK method (2) is represented by the Butcher tableau

$$\frac{\mathbf{A}}{\mathbf{b}^T} := \begin{matrix} \mathbf{A}^{\{1,1\}} & \mathbf{A}^{\{1,2\}} & \dots & \mathbf{A}^{\{1,N\}} \\ \mathbf{A}^{\{2,1\}} & \mathbf{A}^{\{2,2\}} & \dots & \mathbf{A}^{\{2,N\}} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}^{\{N,1\}} & \mathbf{A}^{\{N,2\}} & \dots & \mathbf{A}^{\{N,N\}} \\ \hline \mathbf{b}^{\{1\}T} & \mathbf{b}^{\{2\}T} & \dots & \mathbf{b}^{\{N\}T} \end{matrix}, \quad (3)$$

where $\mathbf{A}^{\{q,m\}} \in \mathbb{R}^{s^{\{q\}} \times s^{\{m\}}}$ and $\mathbf{b}^{\{q\}}, \mathbf{c}^{\{q\}} \in \mathbb{R}^{s^{\{q\}}}$ for $q, m = 1, \dots, N$. Thus, $\mathbf{A} \in \mathbb{R}^{s \times s}$ and $\mathbf{b} \in \mathbb{R}^s$, where $s = \sum_{m=1}^N s^{\{m\}}$. The diagonal blocks of the tableau correspond to the “base” methods which are traditional Runge–Kutta schemes with coefficients $(\mathbf{A}^{\{q,q\}}, \mathbf{b}^{\{q\}}, \mathbf{c}^{\{q\}})$.

The GARK order conditions up to order four are given in (A.1) following [49]. The GARK scheme (3) is internally consistent [49] if

$$\mathbf{A}^{\{q,1\}} \mathbf{1}_{s^{\{1\}}} = \dots = \mathbf{A}^{\{q,N\}} \mathbf{1}_{s^{\{N\}}} = \mathbf{c}^{\{q\}}. \quad (4)$$

Internal consistency greatly simplifies the number of coupling order conditions (A.1) [49].

The GARK scheme (3) is called stiffly accurate [49] if its coefficients satisfy

$$\mathbf{b}^T = \mathbf{e}_s^T \mathbf{A} \Leftrightarrow \mathbf{b}^{\{m\}T} = \mathbf{e}_{s^{\{N\}}}^T \mathbf{A}^{\{N,m\}}, \text{ for } m = 1, \dots, N, \quad (5)$$

where $\mathbf{e}_i \in \mathbb{R}^N$ is the i th column of identity matrix $\mathbf{I}_{N \times N}$. In cases where the dimension is ambiguous, we write this as $\mathbf{e}_{i,N}$. Stiffly accurate GARK methods have favorable stability properties [49] and simpler order conditions for differential algebraic equations [60]. Note in some cases, a GARK method may not satisfy (5), but a permutation of stages can yield an equivalent formulation that does. More formally, denote a permutation matrix by $\mathcal{P} \in \mathbb{R}^{s \times s}$. Reordering the stage numbers accordingly leads to permuted coefficients and a rearranged Butcher tableau (3)

$$\frac{\tilde{\mathbf{A}}}{\tilde{\mathbf{b}}^T} := \frac{\mathcal{P} \mathbf{A} \mathcal{P}^T}{\mathbf{b}^T \mathcal{P}^T}. \quad (6)$$

We are now ready to define implicit-implicit GARK schemes.

Definition 1 (Implicit-implicit GARK schemes). A GARK scheme (3) is implicit-implicit (IMIM-GARK) if there exists a permutation $\mathcal{P} \in \mathbb{R}^{s \times s}$ such that the tableau $\tilde{\mathbf{A}} = \mathcal{P} \mathbf{A} \mathcal{P}^T$ (6) is lower triangular.

The calculation of the stage vectors (2a), carried out in the order defined by the permutation matrix \mathcal{P} , is done in an implicit-decoupled manner, i.e., any implicit stage calculation involves a single unknown stage vector $Y_i^{\{q\}}$. There are no nonlinear equations that require solving for multiple stage vectors simul-

taneously.

Remark 1 (Dependency graph for GARK tableau). For the GARK scheme (3), take \mathbf{A} to be the adjacency matrix for a weighted, directed graph G . Cycles in G indicate implicitness, and vertices in a cycle correspond to a stage values that must be solved simultaneously. Therefore, the IMIM property is equivalent to all cycles in G being loops (cycles with only one edge that start and end at same vertex).

2.1. Implicit-implicit GARK methods with a special coupling structure

Of special interest in this work are IMIM-GARK methods where the Butcher tableau (3) uses the same coefficients within the lower triangular, diagonal, and upper triangular blocks:

$$\frac{\mathbf{A}}{\mathbf{b}^T} := \begin{matrix} \mathbf{A}^{\{D\}} & \mathbf{A}^{\{U\}} & \dots & \mathbf{A}^{\{U\}} \\ \mathbf{A}^{\{L\}} & \mathbf{A}^{\{D\}} & \dots & \mathbf{A}^{\{U\}} \\ \vdots & \vdots & & \vdots \\ \mathbf{A} & \mathbf{A}^{\{L\}} & \dots & \mathbf{A}^{\{D\}} \\ \mathbf{b}^{\{D\}T} & \mathbf{b}^{\{D\}T} & \dots & \mathbf{b}^{\{D\}T} \end{matrix}. \quad (7)$$

This structure requires $\mathbf{A}^{\{U\}}, \mathbf{A}^{\{D\}}, \mathbf{A}^{\{L\}} \in \mathbb{R}^{s^{\{D\}} \times s^{\{D\}}}$ where $s^{\{D\}} := s^{\{1\}} = \dots s^{\{N\}}$. The blocks in (7) correspond to explicit and diagonally implicit Runge–Kutta schemes, as follows:

- $\mathbf{A}^{\{U\}}$ is a strictly lower triangular matrix ($a_{i,j}^{\{U\}} = 0$ for $i \leq j$) corresponding to an explicit Runge–Kutta scheme $(\mathbf{A}^{\{U\}}, \mathbf{b}^{\{D\}}, \mathbf{c}^{\{D\}})$;
- $\mathbf{A}^{\{D\}}$ is a lower triangular matrix ($a_{i,j}^{\{D\}} = 0$ for $i < j$) corresponding to a diagonally implicit Runge–Kutta scheme $(\mathbf{A}^{\{D\}}, \mathbf{b}^{\{D\}}, \mathbf{c}^{\{D\}})$; and
- $\mathbf{A}^{\{L\}}$ is a lower triangular matrix ($a_{i,j}^{\{L\}} = 0$ for $i < j$) corresponding to either an explicit or a diagonally implicit Runge–Kutta scheme $(\mathbf{A}^{\{L\}}, \mathbf{b}^{\{D\}}, \mathbf{c}^{\{D\}})$.

Note that the internal consistency property (4) and the explicit nature of $\mathbf{A}^{\{U\}}$ require that $\mathbf{A}^{\{D\}}$ and $\mathbf{A}^{\{L\}}$ have an explicit first stage (e.g., correspond to ESDIRK schemes). The stiff accuracy property (5) is equivalent to:

$$\mathbf{b}^{\{D\}T} = \mathbf{e}_s^T \mathbf{A}^{\{L\}} = \mathbf{e}_s^T \mathbf{A}^{\{D\}}.$$

The method (7) advances the solution using the following computational

process:

for $i = 1, \dots, s^{\{D\}}$ **do:**

$$Y_i^{\{q\}} = y_n + h \sum_{m < q} \sum_{j=1}^i a_{i,j}^{\{L\}} f_j^{\{m\}} + h \sum_{j=1}^i a_{i,j}^{\{D\}} f_j^{\{q\}} \quad (8a)$$

$$+ h \sum_{m > q} \sum_{j=1}^{i-1} a_{i,j}^{\{U\}} f_j^{\{m\}}, \quad q = 1, \dots, N;$$

$$y_{n+1} = y_n + h \sum_{q=1}^N \sum_{i=1}^{s^{\{D\}}} b_i^{\{D\}} f_i^{\{q\}}. \quad (8b)$$

The method computes the first stage $Y_1^{\{q\}}$ for all components in the order $q = 1, \dots, N$. Then the method computes $Y_i^{\{q\}}$, $i \geq 2$, using the previous stages $Y_1^{\{m\}} \dots Y_{i-1}^{\{m\}}$ for all components m , and the already computed stages $Y_i^{\{1\}} \dots Y_i^{\{q-1\}}$. The lower triangular structure of $\mathbf{A}^{\{D\}}$ implies that $Y_i^{\{q\}}$ is computed implicitly in a SDIRK-like manner when $a_{i,i}^{\{D\}} > 0$, and explicitly when $a_{i,i}^{\{D\}} = 0$. The order in which stages are evaluated in (8) corresponds to the permutation

$$\mathcal{P} = \sum_{i=1}^N \mathbf{e}_{i,N}^T \otimes \mathbf{I}_{s \times s} \otimes \mathbf{e}_{i,N} = \sum_{i=1}^s \mathbf{e}_{i,s}^T \otimes \mathbf{I}_{N \times N} \otimes \mathbf{e}_{i,s}^T, \quad (9)$$

which is known as the vec-permutation matrix [31].

We consider two subclasses of IMIM-GARK schemes, ADI-GARK and parallel ADI-GARK, for their practical appeal.

Definition 2 (ADI-GARK). An Alternating Direction Implicit GARK (ADI-GARK) scheme is an IMIM-GARK method with the structure (7) where $\mathbf{A}^{\{L\}} = \mathbf{A}^{\{D\}} = \mathbf{A}^{\{I\}}$, $\mathbf{A}^{\{U\}} = \mathbf{A}^{\{E\}}$, $\mathbf{b}^{\{D\}} = \mathbf{b}^{\{I\}}$, and $\mathbf{c}^{\{D\}} = \mathbf{c}^{\{I\}}$.

We note that an ADI-GARK method is stiffly accurate (5) iff the implicit component $(\mathbf{A}^{\{I\}}, \mathbf{b}^{\{I\}}, \mathbf{c}^{\{I\}})$ is stiffly accurate in the Runge–Kutta sense.

Definition 3 (Parallel ADI-GARK). A parallel ADI-GARK scheme is an IMIM-GARK method with the structure (7) where $\mathbf{A}^{\{L\}} = \mathbf{A}^{\{U\}} = \mathbf{A}^{\{E\}}$, $\mathbf{A}^{\{D\}} = \mathbf{A}^{\{I\}}$, $\mathbf{b}^{\{D\}} = \mathbf{b}^{\{I\}}$, and $\mathbf{c}^{\{D\}} = \mathbf{c}^{\{I\}}$. The nonlinear systems for stages $Y_i^{\{q\}}$ (8a) are solved in parallel for a given i for all components $q = 1, \dots, N$.

Remark 2. Owing to their structure, the order conditions of ADI-GARK and parallel ADI-GARK schemes are much simpler than the general GARK conditions (A.1) [49]. These order conditions depend on the underlying implicit and explicit methods and are the same for both ADI-GARK and parallel ADI-GARK schemes; the difference between the methods is given by the way these blocks are assembled in the Butcher tableau (7).

In section 7, we provide the coefficients of new ADI-GARK and parallel ADI-GARK schemes of orders three and four.

3. Linear stability analysis

Consider an IMIM-GARK method (7) and the corresponding permuted tableau (6) using the permutation matrix (9). We apply this method to the linear, partitioned system

$$y' = \sum_{m=1}^N \lambda^{\{m\}} y, \quad (10)$$

where $\lambda^{\{m\}} \in \mathbb{C}$. We employ the helpful notation

$$\begin{aligned} \mathbf{z} &:= [z^{\{1\}} \ \dots \ z^{\{N\}}]^T, & \widehat{\mathbf{Z}} &:= \text{diag}(z^{\{1\}}, \dots, z^{\{N\}}), \\ \mathbf{Z}_s &:= \widehat{\mathbf{Z}} \otimes \mathbf{I}_{s \times s}, & \widetilde{\mathbf{Z}}_s &:= \mathbf{I}_{s \times s} \otimes \widehat{\mathbf{Z}}, \end{aligned}$$

where $z^{\{m\}} := h \lambda^{\{m\}}$. From [49], it is known that

$$y_{n+1} = R(\mathbf{z}) y_n,$$

where the stability function $R(\mathbf{z})$ can be written compactly as

$$\begin{aligned} R(\mathbf{z}) &= 1 + \mathbf{b}^T \mathbf{Z}_s (\mathbf{I}_{Ns \times Ns} - \mathbf{A} \mathbf{Z}_s)^{-1} \mathbb{1}_{Ns} \\ &= 1 + \mathbf{b}^T (\mathbf{Z}_s^{-1} - \mathbf{A})^{-1} \mathbb{1}_{Ns} \\ &= 1 + \widetilde{\mathbf{b}}^T (\widetilde{\mathbf{Z}}_s^{-1} - \widetilde{\mathbf{A}})^{-1} \mathbb{1}_{Ns}. \end{aligned} \quad (11)$$

For stiffly accurate methods, this simplifies to

$$R(\mathbf{z}) = \mathbf{e}_{Ns}^T (\mathbf{I}_{Ns \times Ns} - \mathbf{A} \mathbf{Z}_s)^{-1} \mathbb{1}_{Ns},$$

and in the case where \mathbf{A} is invertible, it holds that [49]

$$\lim_{z^{\{N\}} \rightarrow -\infty} R(\mathbf{z}) = 1 - \widetilde{\mathbf{b}}^T \widetilde{\mathbf{A}}^{-1} \mathbb{1}_{Ns} = 1 - \mathbf{b}^T \mathbf{A}^{-1} \mathbb{1}_{Ns}. \quad (12)$$

These results hold for the broad class of IMIM-GARK methods and the many special subclasses. Further stability results for GARK methods applied to scalar and 2×2 linear test problems can be found in [44]. Notably, the decoupled nature of IMIM-GARK methods prevents them from achieving A-stability for 2×2 , linear ODEs [44, Theorem 3.4].

4. Classical LOD and ADI methods in the GARK framework

This section is devoted to showing that classical Locally One-Dimensional (LOD) and Alternating Direction Implicit (ADI) schemes appearing in the liter-

ature can be formulated within the IMIM-GARK framework. In particular, the consistency of such methods can be treated in a unified way within the GARK formalism. To this end we consider the non-autonomous ODE problem (1). We shall deal with GARK methods of the form (3). The consistency order of the following methods can be checked by using the order conditions indicated in Appendix A at the end of the manuscript.

4.1. LOD-Backward Euler method

The LOD-Backward Euler method [34, page 348] reads

$$\begin{aligned} v_0 &= y_n, \\ v_q &= v_{q-1} + h f^{\{q\}}(t_{n+1}, v_q), \quad q = 1, \dots, N, \\ y_{n+1} &= v_N. \end{aligned} \tag{13}$$

We rewrite (13) in GARK notation with the stages $Y^{\{q\}} := v_q$:

$$\begin{aligned} Y^{\{q\}} &= y_n + h \sum_{m=1}^q f^{\{m\}}(t_{n+1}, Y^{\{m\}}), \quad q = 1, \dots, N, \\ y_{n+1} &= Y^{\{N\}} = y_n + h \sum_{m=1}^N f^{\{m\}}(t_{n+1}, Y^{\{m\}}). \end{aligned} \tag{14}$$

This method has order of consistency one, and it is stiffly accurate but is not internally consistent. It has the form (7) and further is an ADI-GARK method with the coefficients

$$\begin{aligned} \mathbf{A}^{\{q,m\}} &= \mathbf{A}^{\{1\}} = [1], \text{ for } m \leq q, & \mathbf{A}^{\{q,m\}} &= \mathbf{A}^{\{E\}} = [0], \text{ for } m > q, \\ \mathbf{b}^{\{q\}} &= \mathbf{b}^{\{1\}} = [1], & \mathbf{c}^{\{q\}} &= \mathbf{c}^{\{1\}} = [1], \end{aligned}$$

for $q, m = 1, \dots, N$.

4.2. Yanenko's LOD-Crank-Nicolson method

Yanenko's LOD-Crank-Nicolson method [34, page 351] reads

$$\begin{aligned} v_0 &= y_n, \\ v_q &= v_{q-1} + \frac{h}{2} \left(f^{\{q\}}(t_n + c_{q-1} h, v_{q-1}) + f^{\{q\}}(t_n + c_q h, v_q) \right), \\ q &= 1, \dots, N, \\ y_{n+1} &= v_N, \end{aligned} \tag{15}$$

with $c_0 = 0$, $c_N = 1$, and $c_q = 1/2$ for $q = 1, \dots, N-1$. Iterating the stage values in (15) leads to the following formula:

$$v_q = y_n + \frac{h}{2} \sum_{m=1}^q \left(f^{\{m\}}(t_n + c_{m-1} h, v_{m-1}) + f^{\{m\}}(t_n + c_m h, v_m) \right).$$

The method is cast in the GARK form (3) by defining the stages

$$Y^{\{1\}} := \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}, \quad \dots \quad Y^{\{q\}} := \begin{bmatrix} v_{q-1} \\ v_q \end{bmatrix}, \quad \dots \quad Y^{\{N\}} := \begin{bmatrix} v_{N-1} \\ v_N \end{bmatrix}.$$

Note that $Y_1^{\{q\}} = Y_2^{\{q-1\}}$ for $q \geq 2$. We rewrite (15) in GARK notation as follows:

$$\begin{aligned} Y_1^{\{q\}} &= y_n + \frac{h}{2} \sum_{m=1}^{q-1} \left(f^{\{m\}}(t_n + c_{m-1} h, Y_1^{\{m\}}) + f^{\{m\}}(t_n + c_m h, Y_2^{\{m\}}) \right), \\ Y_2^{\{q\}} &= y_n + \frac{h}{2} \sum_{m=1}^q \left(f^{\{m\}}(t_n + c_{m-1} h, Y_1^{\{m\}}) + f^{\{m\}}(t_n + c_m h, Y_2^{\{m\}}) \right), \\ y_{n+1} &= Y_2^{\{N\}}. \end{aligned} \tag{16}$$

The GARK Butcher tableau (7) is defined by the coefficient matrices:

$$\begin{aligned} \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{L\}} &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \text{ for } m < q, & \mathbf{A}^{\{q,q\}} = \mathbf{A}^{\{D\}} &= \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \\ \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{U\}} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{ for } m > q, & \mathbf{b}^{\{q\}} = \mathbf{b}^{\{D\}} &= \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \\ \mathbf{c}^{\{q\}} = \mathbf{c}^{\{D\}} &= \begin{bmatrix} c_{q-1} \\ c_q \end{bmatrix}, & m, q &= 1, \dots, N. \end{aligned}$$

We then immediately observe that (16) is only order one. Furthermore, it is stiffly accurate and not internally consistent. On the other hand, the matrix \mathbf{A} is already in lower triangular form since the stages values $Y_i^{\{q\}}$ are displayed in the same way they are actually computed.

Remark 3. Symmetric and parallel versions of the method (15) are known to provide respective second order methods at the expense of doubling the computational cost [34, page 351-352]. Both methods can also be expressed as GARK methods. First, the symmetric Yanenko's LOD-Crank-Nicolson method [34,

page 351] is given by

$$\begin{aligned}
v_0 &= y_n, \\
v_q &= v_{q-1} + \frac{h}{4} \left(f^{\{q\}} \left(t_n + \frac{c_{q-1}}{2} h, v_{q-1} \right) + f^{\{q\}} \left(t_n + \frac{c_q}{2} h, v_q \right) \right), \\
q &= 1, \dots, N, \\
w_0 &= v_N, \\
w_q &= w_{q-1} + \frac{h}{4} \left(f^{\{N+1-q\}} \left(t_n + \frac{1+c_{q-1}}{2} h, w_{q-1} \right) \right. \\
&\quad \left. + f^{\{N+1-q\}} \left(t_n + \frac{1+c_q}{2} h, w_q \right) \right), \quad q = 1, \dots, N, \\
y_{n+1} &= w_N.
\end{aligned} \tag{17}$$

This computational process is mapped to GARK stages by

$$Y^{\{1\}} := \begin{bmatrix} v_0 \\ v_1 \\ w_{N-1} \\ w_N \end{bmatrix}, \quad \dots \quad Y^{\{q\}} := \begin{bmatrix} v_{q-1} \\ v_q \\ w_{N-q} \\ w_{N+1-q} \end{bmatrix}, \quad \dots \quad Y^{\{N\}} := \begin{bmatrix} v_{N-1} \\ v_N \\ w_0 \\ w_1 \end{bmatrix}. \tag{18}$$

Now, (17) can be formulated in the GARK framework as follows:

$$\begin{aligned}
Y_1^{\{q\}} &= y_n + \frac{h}{4} \sum_{m=1}^{q-1} \left(f^{\{m\}} \left(t_n + \frac{c_{m-1}}{2} h, Y_1^{\{m\}} \right) + f^{\{m\}} \left(t_n + \frac{c_m}{2} h, Y_2^{\{m\}} \right) \right), \\
Y_2^{\{q\}} &= Y_1^{\{q+1\}}, \quad q = 1, \dots, N, \\
Y_3^{\{q\}} &= y_n + \frac{h}{4} \sum_{m=1}^N \left(f^{\{m\}} \left(t_n + \frac{c_{m-1}}{2} h, Y_1^{\{m\}} \right) + f^{\{m\}} \left(t_n + \frac{c_m}{2} h, Y_2^{\{m\}} \right) \right) \\
&\quad + \frac{h}{4} \sum_{m=q+1}^N \left(f^{\{m\}} \left(t_n + \frac{1+c_{N-m}}{2} h, Y_3^{\{m\}} \right) \right. \\
&\quad \left. + f^{\{m\}} \left(t_n + \frac{1+c_{N+1-m}}{2} h, Y_4^{\{m\}} \right) \right), \\
Y_4^{\{q\}} &= Y_3^{\{q-1\}}, \quad q = N, \dots, 1, \\
y_{n+1} &= Y_4^{\{1\}}.
\end{aligned} \tag{19}$$

The corresponding coefficient matrices are then defined by

$$\begin{aligned}
\mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{L\}} &= \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 \end{bmatrix}, \text{ for } m < q, \\
\mathbf{A}^{\{q,q\}} = \mathbf{A}^{\{D\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}, \\
\mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{U\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}, \text{ for } m > q, \\
\mathbf{b}^{\{q\}T} = \mathbf{b}^{\{D\}T} &= \left[\frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \right], \\
\mathbf{c}^{\{q\}T} = \mathbf{c}^{\{D\}T} &= \left[\frac{c_{q-1}}{2} \quad \frac{c_q}{2} \quad \frac{1+c_{N-q}}{2} \quad \frac{1+c_{N+1-q}}{2} \right],
\end{aligned}$$

for $q, m = 1, \dots, N$. Now it is readily checked that (19) reaches order two. Furthermore, observe that the method expressed this way does not satisfy the stiff accuracy condition (5). However, the stages can be reordered into a lower triangular form that is stiffly accurate. The permuted tableau (6) is given by

	$f_1^{\{1\}}$	$f_2^{\{1\}}$	\dots	$f_1^{\{N\}}$	$f_2^{\{N\}}$	$f_3^{\{N\}}$	$f_4^{\{N\}}$	\dots	$f_3^{\{1\}}$	$f_4^{\{1\}}$
$Y_1^{\{1\}}$	0	0	\dots	0	0	0	0	\dots	0	0
$Y_2^{\{1\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	0	0	0	0	\dots	0	0
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$Y_1^{\{N\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	0	0	0	0	\dots	0	0
$Y_2^{\{N\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$	0	0	\dots	0	0
$Y_3^{\{N\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$	0	0	\dots	0	0
$Y_4^{\{N\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	0	0
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$Y_3^{\{1\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	0	0
$Y_4^{\{1\}}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$
y_{n+1}	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	\dots	$\frac{1}{4}$	$\frac{1}{4}$

and in this form, the stiff accuracy is evident.

Secondly, for the parallel Yanenko's LOD-Crank-Nicolson method [34, page

352]

$$\begin{aligned}
v_0 &= y_n, \\
v_q &= v_{q-1} + \frac{h}{2} \left(f^{\{q\}}(t_n + c_{q-1} h, v_{q-1}) + f^{\{q\}}(t_n + c_q h, v_q) \right), \\
q &= 1, \dots, N, \\
w_0 &= y_n, \\
w_q &= w_{q-1} + \frac{h}{2} \left(f^{\{N+1-q\}}(t_n + c_{q-1} h, w_{q-1}) + f^{\{N+1-q\}}(t_n + c_q h, w_q) \right), \\
q &= 1, \dots, N, \\
y_{n+1} &= \frac{1}{2}(v_N + w_N),
\end{aligned} \tag{20}$$

we similarly consider the stages (18) and rewrite (20) in GARK notation as

$$\begin{aligned}
Y_1^{\{q\}} &= y_n + \frac{h}{2} \sum_{m=1}^{q-1} \left(f^{\{m\}}(t_n + c_{m-1} h, Y_1^{\{m\}}) + f^{\{m\}}(t_n + c_m h, Y_2^{\{m\}}) \right), \\
Y_2^{\{q\}} &= Y_1^{\{q+1\}}, \quad q = 1, \dots, N, \\
Y_3^{\{q\}} &= y_n + \frac{h}{4} \sum_{m=q+1}^N \left(f^{\{m\}}(t_n + c_{N-m} h, Y_3^{\{m\}}) \right. \\
&\quad \left. + f^{\{m\}}(t_n + c_{N+1-m} h, Y_4^{\{m\}}) \right), \\
Y_4^{\{q\}} &= Y_3^{\{q-1\}}, \quad q = N, \dots, 1, \\
y_{n+1} &= \frac{1}{2} (Y_2^{\{N\}} + Y_4^{\{1\}}).
\end{aligned} \tag{21}$$

Its GARK Butcher tableau (7) is then defined by the following coefficient ma-

trices:

$$\begin{aligned}
\mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{\text{L}\}} &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ for } m < q, \\
\mathbf{A}^{\{q,q\}} = \mathbf{A}^{\{\text{D}\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \\
\mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{\text{U}\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \text{ for } m > q, \\
\mathbf{b}^{\{q\}T} = \mathbf{b}^{\{\text{D}\}} &= \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}, \\
\mathbf{c}^{\{q\}T} = \mathbf{c}^{\{\text{D}\}T} &= [c_{q-1} \quad c_q \quad c_{N-q} \quad c_{N+1-q}].
\end{aligned}$$

From here, it is readily checked that (21) reaches order two.

4.3. Trapezoidal Splitting

The Trapezoidal Splitting scheme [34, page 359] reads

$$\begin{aligned}
v_0 &= y_n, \\
v_q &= v_{q-1} + \frac{h}{2} f^{\{q\}}(t_n, v_{q-1}), \quad q = 1, \dots, N, \\
v_{N+q} &= v_{N+q-1} + \frac{h}{2} f^{\{N+1-q\}}(t_{n+1}, v_{N+q}), \quad q = 1, \dots, N, \\
y_{n+1} &= v_{2N}.
\end{aligned} \tag{22}$$

The method is cast in the GARK form (3) by defining the stages

$$Y^{\{1\}} := \begin{bmatrix} v_0 \\ v_{2N} \end{bmatrix}, \quad \dots \quad Y^{\{q\}} := \begin{bmatrix} v_{q-1} \\ v_{2N-q+1} \end{bmatrix}, \quad \dots \quad Y^{\{N\}} := \begin{bmatrix} v_{N-1} \\ v_{N+1} \end{bmatrix}.$$

Iterating the stages (22) leads to

$$\begin{aligned}
Y_1^{\{q\}} &= y_n + \frac{h}{2} \sum_{m=1}^{q-1} f^{\{m\}}(t_n, Y_1^{\{m\}}), \\
Y_2^{\{q\}} &= y_n + \frac{h}{2} \sum_{m=1}^N f^{\{m\}}(t_n, Y_1^{\{m\}}) + \frac{h}{2} \sum_{m=q}^N f^{\{m\}}(t_{n+1}, Y_2^{\{m\}}), \quad (23) \\
q &= N, \dots, 1 \\
y_{n+1} &= Y_2^{\{1\}},
\end{aligned}$$

and the coefficient matrices

$$\begin{aligned}
\mathbf{A}^{\{q,m\}} &= \mathbf{A}^{\{1\}} = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 0 \end{bmatrix}, \text{ for } m < q, & \mathbf{b}^{\{q\}} &= \mathbf{b}^{\{D\}} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \\
\mathbf{A}^{\{q,m\}} &= \mathbf{A}^{\{D\}} = \mathbf{A}^{\{u\}} = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \text{ for } m \geq q, & \mathbf{c}^{\{q\}} &= \mathbf{c}^{\{D\}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
\end{aligned}$$

From (A.1), it is observed that (23) is a second order method and is not internally consistent. The method (23) also admits the following lower triangular formulation where the stage values are presented in the way they are solved:

$$\begin{array}{c|cccc|cccc}
& f_1^{\{1\}} & f_1^{\{2\}} & \dots & f_1^{\{N\}} & f_2^{\{N\}} & f_2^{\{N-1\}} & \dots & f_2^{\{1\}} \\
\hline
Y_1^{\{1\}} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
Y_1^{\{2\}} & \frac{1}{2} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\
Y_1^{\{N\}} & \frac{1}{2} & \frac{1}{2} & \dots & 0 & 0 & 0 & \dots & 0 \\
\hline
Y_2^{\{N\}} & \frac{1}{2} & \frac{1}{2} & \dots & \frac{1}{2} & \frac{1}{2} & 0 & \dots & 0 \\
Y_2^{\{N-1\}} & \frac{1}{2} & \frac{1}{2} & \dots & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \dots & 0 \\
\vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\
Y_2^{\{1\}} & \frac{1}{2} & \frac{1}{2} & \dots & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \dots & \frac{1}{2} \\
\hline
y_{n+1} & \frac{1}{2} & \frac{1}{2} & \dots & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \dots & \frac{1}{2}
\end{array}$$

In this permuted form, the stiff accuracy condition holds.

4.4. Douglas method

We now consider the partitioned system

$$y' = f(t, y) = \sum_{m=0}^N f^{\{m\}}(t, y). \quad (24)$$

Compared to (1), there is the additional partition $f^{\{0\}}$, which is nonstiff and can be treated explicitly. The Douglas splitting scheme [34, page 373] applied to (24) reads

$$\begin{aligned} v_0 &= y_n + h f(t_n, y_n), \\ v_q &= v_{q-1} + \theta h \left(f^{\{q\}}(t_{n+1}, v_q) - f^{\{q\}}(t_n, y_n) \right), \quad q = 1, \dots, N, \\ y_{n+1} &= v_N. \end{aligned} \quad (25)$$

Note that an explicit Euler step for the entire system is followed by stabilizing corrections, which are implicit stages in only one component at a time. The method (25) is cast in the GARK formulation by defining the stages

$$\begin{aligned} Y_1^{\{q\}} &= y_n, \quad q = 0, \dots, N, \\ Y_2^{\{q\}} &= y_n + h f^{\{0\}}(t_n, Y_1^{\{0\}}) \\ &\quad + (1 - \theta) h \sum_{m=1}^q f^{\{m\}}(t_n, Y_1^{\{m\}}) + h \sum_{m=q+1}^N f^{\{m\}}(t_n, Y_1^{\{m\}}) \\ &\quad + \theta h \sum_{m=1}^q f^{\{m\}}(t_{n+1}, Y_2^{\{m\}}), \quad q = 1, \dots, N, \\ y_{n+1} &= Y_2^{\{N\}}. \end{aligned}$$

From here, we can see this corresponds to an ADI-GARK scheme with coefficient matrices

$$\begin{aligned} \mathbf{A}^{\{q,m\}} &= \mathbf{A}^{\{1\}} = \begin{bmatrix} 0 & 0 \\ 1 - \theta & \theta \end{bmatrix}, \text{ for } m \leq q, & \mathbf{b}^{\{q\}} &= \mathbf{b}^{\{1\}} = \begin{bmatrix} 1 - \theta \\ \theta \end{bmatrix}, \\ \mathbf{A}^{\{q,m\}} &= \mathbf{A}^{\{2\}} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \text{ for } m > q, & \mathbf{c}^{\{q\}} &= \mathbf{c}^{\{2\}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \end{aligned} \quad (26a)$$

for $q, m = 1, \dots, N$. The nonstiff partition $f^{\{0\}}$ prepends the coefficients

$$\mathbf{A}^{\{0,0\}} = \mathbf{c}^{\{0\}} = [0], \quad \mathbf{A}^{\{q,0\}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{A}^{\{0,q\}} = [0 \ 0], \quad \mathbf{b}^{\{0\}} = [1], \quad (26b)$$

to the tableau, with $q = 1, \dots, N$. We observe that the method is internally consistent and stiffly accurate. Considering (A.1), we immediately check that (26) is a first order method for arbitrary $f^{\{0\}}$ and all $\theta \in \mathbb{R}$. In fact, the method reaches order two iff $f^{\{0\}} \equiv 0$ and $\theta = \frac{1}{2}$.

Remark 4. Modified versions of the Douglas scheme (25) have been considered in [1, 33] in such a way that these modifications provide second order approximation when $\theta = \frac{1}{2}$ regardless of the splitting term $f^{\{0\}}$. A first modified Douglas method is obtained by introducing an initial stabilizing correction for

$f^{\{0\}}$ performed as

$$\begin{aligned}\hat{v}_0 &= y_n + h f(t_n, y_n) \\ v_0 &= \hat{v}_0 + \theta h \left(f^{\{0\}}(t_{n+1}, \hat{v}_0) - f^{\{0\}}(t_n, y_n) \right), \\ v_q &= v_{q-1} + \theta h \left(f^{\{q\}}(t_{n+1}, v_q) - f^{\{q\}}(t_n, y_n) \right), \quad q = 1, \dots, N, \\ y_{n+1} &= v_N.\end{aligned}\tag{27}$$

The method (27) has GARK stages

$$\begin{aligned}Y_1^{\{q\}} &= y_n, \quad q = 0, \dots, N, \\ Y_2^{\{0\}} &= y_n + h \sum_{m=0}^N f^{\{m\}}(t_n, Y_1^{\{m\}}), \\ Y_2^{\{q\}} &= y_n + (1 - \theta) h \sum_{m=0}^q f^{\{m\}}(t_n, Y_1^{\{m\}}) + h \sum_{m=q+1}^N f^{\{m\}}(t_n, Y_1^{\{m\}}) \\ &\quad + \theta h \sum_{m=0}^q f^{\{m\}}(t_{n+1}, Y_2^{\{m\}}), \quad q = 1, \dots, N, \\ y_{n+1} &= Y_2^{\{N\}}.\end{aligned}$$

The coefficient matrices for partitions 1 to N are identical to the original Douglas splitting coefficients (26a). For partition $f^{\{0\}}$, however, we replace (26b) with

$$\begin{aligned}\mathbf{A}^{\{0,0\}} &= \mathbf{A}^{\{0,q\}} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}^{\{q,0\}} = \begin{bmatrix} 0 & 0 \\ 1 - \theta & \theta \end{bmatrix}, \\ \mathbf{b}^{\{0\}T} &= [1 - \theta \quad \theta], \quad \mathbf{c}^{\{0\}T} = [0 \quad 1],\end{aligned}$$

for $q = 1, \dots, N$. From here, (27) is a first order method for arbitrary $f^{\{0\}}$ and all $\theta \in \mathbb{R}$. Moreover, the method reaches order two iff $\theta = \frac{1}{2}$, regardless of $f^{\{0\}}$.

On the other hand, a second modification of the Douglas method (25) is obtained by introducing a stabilizing correction for $f^{\{0\}}$ performed at the end of a step:

$$\begin{aligned}v_0 &= y_n + h f(t_n, y_n) \\ v_q &= v_{q-1} + \theta h \left(f^{\{q\}}(t_{n+1}, v_q) - f^{\{q\}}(t_n, y_n) \right), \quad q = 1, \dots, N, \\ y_{n+1} &= v_N + \theta h \left(f^{\{0\}}(t_{n+1}, v_N) - f^{\{0\}}(t_n, y_n) \right).\end{aligned}\tag{28}$$

The corresponding stages are

$$\begin{aligned}
Y_1^{(q)} &= y_n, \quad q = 0, \dots, N, \\
Y_2^{(q)} &= y_n + h f^{(0)}(t_n, Y_1^{(0)}) \\
&\quad + (1 - \theta) h \sum_{m=1}^q f^{(m)}(t_n, Y_1^{(m)}) + h \sum_{m=q+1}^N f^{(m)}(t_n, Y_1^{(m)}) \\
&\quad + \theta h \sum_{m=1}^q f^{(m)}(t_{n+1}, Y_2^{(m)}), \quad q = 1, \dots, N, \\
Y_2^{(0)} &= y_n + h f^{(0)}(t_n, Y_1^{(0)}) \\
&\quad + h \sum_{m=1}^N \left((1 - \theta) f^{(m)}(t_n, Y_1^{(m)}) + \theta f^{(m)}(t_{n+1}, Y_2^{(m)}) \right), \\
y_{n+1} &= y_n + h \sum_{m=0}^N \left((1 - \theta) f^{(m)}(t_n, Y_1^{(m)}) + \theta f^{(m)}(t_{n+1}, Y_2^{(m)}) \right).
\end{aligned}$$

Once again, the coefficient matrices for partitions 1 to N coincide with (26a), and partition $f^{(0)}$ has the coefficients

$$\begin{aligned}
\mathbf{A}^{(0,0)} &= \mathbf{A}^{(q,0)} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}^{(0,q)} = \begin{bmatrix} 0 & 0 \\ 1 - \theta & \theta \end{bmatrix}, \\
\mathbf{b}^{(0)T} &= [1 - \theta \quad \theta], \quad \mathbf{c}^{(0)T} = [0 \quad 1],
\end{aligned}$$

for $q = 1, \dots, N$. We observe that in this case the method (28) is not stiffly accurate. Anyway, it is a first order method for arbitrary $f^{(0)}$ and all $\theta \in \mathbb{R}$, and it reaches order two iff $\theta = \frac{1}{2}$, regardless of $f^{(0)}$.

4.5. Modified Craig-Sneyd scheme

We consider the split system (24) with one non-stiff component $f^{(0)}$. The second order Modified Craig-Sneyd scheme [32] reads:

$$\begin{aligned}
v_0 &= y_n + h f(t_n, y_n), \\
v_q &= v_{q-1} + \theta h \left(f^{(q)}(t_{n+1}, v_q) - f^{(q)}(t_n, y_n) \right), \quad q = 1, \dots, N, \\
\hat{v}_0 &= v_0 + \sigma h \left(f^{(0)}(t_{n+1}, v_N) - f^{(0)}(t_n, y_n) \right), \\
w_0 &= \hat{v}_0 + \mu h (f(t_{n+1}, v_N) - f(t_n, y_n)), \\
w_q &= w_{q-1} + \theta h \left(f^{(q)}(t_{n+1}, w_q) - f^{(q)}(t_n, y_n) \right), \quad q = 1, \dots, N, \\
y_{n+1} &= w_N.
\end{aligned} \tag{29}$$

Equation (29) is order two iff $\sigma = \theta$ and $\mu = \frac{1}{2} - \theta$. In the special case $\mu = 0$, we recover the so called Craig-Sneyd scheme [17].

The method is cast in the GARK form (3) by defining the stages

$$Y^{\{0\}} := \begin{bmatrix} y_n \\ v_N \end{bmatrix}, \quad Y^{\{q\}} := \begin{bmatrix} y_n \\ v_q \\ v_N \\ w_q \end{bmatrix}, \text{ for } q = 1, \dots, N. \quad (30)$$

The coefficients correspond to that of an ADI-GARK method:

$$\begin{aligned} \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{1\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \\ 1-\mu-\theta & 0 & \mu & \theta \end{bmatrix}, \text{ for } m \leq q, \\ \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{E\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \\ 1-\mu & 0 & \mu & 0 \end{bmatrix}, \text{ for } m > q, \\ \mathbf{b}^{\{q\}T} = \mathbf{b}^{\{1\}T} &= [1-\mu-\theta \ 0 \ \mu \ \theta], \quad \mathbf{c}^{\{q\}T} = \mathbf{c}^{\{1\}T} = [0 \ 1 \ 1 \ 1]. \end{aligned}$$

The nonstiff partition introduces the following additional coefficients:

$$\begin{aligned} \mathbf{A}^{\{0,0\}} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, & \mathbf{A}^{\{0,q\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \end{bmatrix}, \\ \mathbf{A}^{\{q,0\}} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1-\sigma-\mu & \sigma+\mu \end{bmatrix}, & \mathbf{b}^{\{0\}} &= \begin{bmatrix} 1-\sigma-\mu \\ \sigma+\mu \end{bmatrix}, & \mathbf{c}^{\{0\}} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned}$$

4.6. Hundsdorfer–Verwer scheme

We consider the split system (24) with one non-stiff component $f^{\{0\}}$. The Hundsdorfer–Verwer scheme [32] reads:

$$\begin{aligned} v_0 &= y_n + h f(t_n, y_n), \\ v_q &= v_{q-1} + \theta h \left(f^{\{q\}}(t_{n+1}, v_q) - f^{\{q\}}(t_n, y_n) \right), \quad q = 1, \dots, N, \\ w_0 &= v_0 + \mu h \left(f(t_{n+1}, v_N) - f(t_n, y_n) \right), \\ w_q &= w_{q-1} + \theta h \left(f^{\{q\}}(t_{n+1}, w_q) - f^{\{q\}}(t_{n+1}, v_N) \right), \quad q = 1, \dots, N, \\ y_{n+1} &= w_N. \end{aligned} \quad (31)$$

The method is cast in the GARK form (3) using the same stages as (30) and the coefficient matrices

$$\begin{aligned}\mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{\mathbb{I}\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \\ 1-\mu & 0 & \mu-\theta & \theta \end{bmatrix}, \text{ for } m \leq q; \\ \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{\mathbb{E}\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \\ 1-\mu & 0 & \mu & 0 \end{bmatrix}, \text{ for } m > q, \\ \mathbf{b}^{\{q\}T} = \mathbf{b}^{\{\mathbb{I}\}T} &= [1-\mu \ 0 \ \mu-\theta \ \theta], \quad \mathbf{c}^{\{q\}T} = \mathbf{c}^{\{\mathbb{I}\}T} = [0 \ 1 \ 1 \ 1],\end{aligned}$$

together with

$$\begin{aligned}\mathbf{A}^{\{0,0\}} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}^{\{0,q\}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1-\theta & \theta & 0 & 0 \end{bmatrix}, \quad \mathbf{A}^{\{q,0\}} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1-\mu & \mu \end{bmatrix}, \\ \mathbf{b}^{\{0\}} &= \begin{bmatrix} 1-\mu \\ \mu \end{bmatrix}, \quad \mathbf{c}^{\{0\}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.\end{aligned}$$

The Hundsdorfer–Verwer Splitting method (31) is an ADI-GARK scheme, is stiffly accurate, and has order two iff $\mu = \frac{1}{2}$.

5. Classical operator splitting methods in the GARK framework

We next describe how classical operator splitting schemes appearing in the literature can be formulated within the GARK framework.

5.1. The second order Strang splitting scheme

A Strang splitting [56] integrates individual sub-systems in sequence, a symmetrical order. For an arbitrary number N of partitions, the Strang split formulation [56] is

$$\begin{aligned}v_0 &= y_n \\ v'_q &= f^{\{q\}}(t, v_q), \quad v_q(t_n) = v_{q-1}(t_n + \frac{h}{2}), \quad t_n \leq t \leq t_n + \frac{h}{2}, \\ q &= 1, \dots, N, \\ w_0 &= v_N, \\ w'_q &= f^{\{N-q+1\}}(t, w_q), \quad w_q(t_n + \frac{h}{2}) = w_{q-1}(t_n + h), \quad t_n + \frac{h}{2} \leq t \leq t_n + h, \\ q &= 1, \dots, N, \\ y_{n+1} &= w_N(t_n + h).\end{aligned}$$

Assume that each integration is carried out by one step of an arbitrary, s stage Runge–Kutta scheme (A, b, c) . The method is cast in the GARK form (3) by defining the stages

$$Y^{\{q\}} = \begin{bmatrix} Y_{1:s}^{\{q\}} & \text{(from integration of } v_q) \\ Y_{s+1:2s}^{\{q\}} & \text{(from integration of } w_{N-q+1}) \end{bmatrix}.$$

The Butcher tableau is of the form (7) with the coefficient matrices

$$\begin{aligned} \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{\text{L}\}} &= \begin{bmatrix} \frac{1}{2} \mathbf{1} b^T & 0 \\ \frac{1}{2} \mathbf{1} b^T & 0 \end{bmatrix}, \text{ for } m < q, \\ \mathbf{A}^{\{q,q\}} = \mathbf{A}^{\{\text{D}\}} &= \begin{bmatrix} \frac{1}{2} A & 0 \\ \frac{1}{2} \mathbf{1} b^T & \frac{1}{2} A \end{bmatrix}, \quad \mathbf{b}^{\{q\}} = \mathbf{b}^{\{\text{D}\}} = \begin{bmatrix} \frac{1}{2} b \\ \frac{1}{2} b \end{bmatrix}, \\ \mathbf{A}^{\{q,m\}} = \mathbf{A}^{\{\text{U}\}} &= \begin{bmatrix} 0 & 0 \\ \frac{1}{2} \mathbf{1} b^T & \frac{1}{2} \mathbf{1} b^T \end{bmatrix}, \text{ for } m > q. \end{aligned}$$

5.2. High order Yoshida splitting schemes

Yoshida proposed high order splitting schemes [63] that require backward-in-time integration. Let $y_{n+1} = S_h y_n$ denote formally the Strang splitting solution operator. Then Yoshida’s fourth order splitting is obtained by a repeated application of Strang solutions:

$$y_{n+1} = S_{\theta h} S_{(1-2\theta)h} S_{\theta h} y_n, \quad \theta = \frac{1}{2 - 2^{1/3}} \approx 1.35121. \quad (32)$$

General Yoshida schemes applied to a two-way partitioned system read

$$\begin{aligned} v_\ell(t_n + c_{\ell-1}^\alpha h) &= \begin{cases} y_n & \ell = 1 \\ w_{\ell-1}(t_n + c_{\ell-1}^\beta h), & \text{otherwise} \end{cases} \\ v'_\ell &= f^{\{1\}}(v_\ell), \quad t_n + c_{\ell-1}^\alpha h \leq t \leq t_n + c_\ell^\alpha h, \\ w_\ell(t_n + c_{\ell-1}^\beta h) &= v_\ell(t_n + c_\ell^\alpha h), \\ w'_\ell &= f^{\{2\}}(w_\ell), \quad t_n + c_{\ell-1}^\beta h \leq t \leq t_n + c_\ell^\beta h, \\ &\ell = 1, \dots, L \\ y_{n+1} &= w_L(t_n + h), \end{aligned}$$

where

$$c_0^\alpha = 0, \quad c_0^\beta = 0, \quad c_\ell^\alpha = \sum_{i=1}^{\ell} \alpha_i, \quad c_\ell^\beta = \sum_{i=1}^{\ell} \beta_i, \quad c_L^\alpha = 1, \quad c_L^\beta = 1.$$

The scheme of order four (32) uses $L = 4$ and the coefficients

$$\begin{aligned}\alpha_1 = \alpha_4 &= \frac{1}{2(2 - 2^{1/3})}, \quad \alpha_2 = \alpha_3 = \frac{1 - 2^{1/3}}{2(2 - 2^{1/3})}, \\ \beta_1 = \beta_3 &= \frac{1}{2 - 2^{1/3}}, \quad \beta_2 = -\frac{2^{1/3}}{2 - 2^{1/3}}, \quad \beta_4 = 0.\end{aligned}$$

Assume that each integration is carried out with a s stage Runge–Kutta scheme (A, b, c) . Then, we can define the GARK stages

$$\begin{aligned}Y^{\{1\}} &= \begin{bmatrix} Y_{1:s}^{\{1\}} & \text{(from integration of } v_1) \\ \vdots & \\ Y_{(L-1)s+1:Ls}^{\{1\}} & \text{(from integration of } v_L) \end{bmatrix}, \\ Y^{\{2\}} &= \begin{bmatrix} Y_{1:s}^{\{2\}} & \text{(from integration of } w_1) \\ \vdots & \\ Y_{(L-1)s+1:Ls}^{\{2\}} & \text{(from integration of } w_L) \end{bmatrix}.\end{aligned}$$

The GARK tableau (3) for (32) reads

$c_0^\alpha \mathbb{1} + \alpha_1 c$	$\alpha_1 A$							
$c_1^\alpha \mathbb{1} + \alpha_2 c$	$\alpha_1 \mathbb{1} b^T$	$\alpha_2 A$			$\beta_1 \mathbb{1} b^T$			
$c_2^\alpha \mathbb{1} + \alpha_3 c$	$\alpha_1 \mathbb{1} b^T$	$\alpha_2 \mathbb{1} b^T$	$\alpha_3 A$		$\beta_1 \mathbb{1} b^T$	$\beta_2 \mathbb{1} b^T$		
$c_3^\alpha \mathbb{1} + \alpha_4 c$	$\alpha_1 \mathbb{1} b^T$	$\alpha_2 \mathbb{1} b^T$	$\alpha_3 \mathbb{1} b^T$	$\alpha_4 A$	$\beta_1 \mathbb{1} b^T$	$\beta_2 \mathbb{1} b^T$	$\beta_3 \mathbb{1} b^T$	
$c_0^\beta \mathbb{1} + \beta_1 c$	$\alpha_1 \mathbb{1} b^T$				$\beta_1 A$			
$c_1^\beta \mathbb{1} + \beta_2 c$	$\alpha_1 \mathbb{1} b^T$	$\alpha_2 \mathbb{1} b^T$			$\beta_1 \mathbb{1} b^T$	$\beta_2 A$		
$c_2^\beta \mathbb{1} + \beta_3 c$	$\alpha_1 \mathbb{1} b^T$	$\alpha_2 \mathbb{1} b^T$	$\alpha_3 \mathbb{1} b^T$		$\beta_1 \mathbb{1} b^T$	$\beta_2 \mathbb{1} b^T$	$\beta_3 A$	
$c_3^\beta \mathbb{1} + \beta_4 c$	$\alpha_1 \mathbb{1} b^T$	$\alpha_2 \mathbb{1} b^T$	$\alpha_3 \mathbb{1} b^T$	$\alpha_4 \mathbb{1} b^T$	$\beta_1 \mathbb{1} b^T$	$\beta_2 \mathbb{1} b^T$	$\beta_3 \mathbb{1} b^T$	$\beta_4 A$
	$\alpha_1 b^T$	$\alpha_2 b^T$	$\alpha_3 b^T$	$\alpha_4 b^T$	$\beta_1 b^T$	$\beta_2 b^T$	$\beta_3 b^T$	$\beta_4 b^T$

6. Fractional step Runge–Kutta methods in the GARK framework

Consider the additively split ODE (24) where the first N components are one-dimensional split linear operators, and the last partition is a nonlinear term:

$$y' = f(t, y) = \sum_{m=1}^N \left(\mathbf{L}^{\{m\}}(t) y + \phi^{\{m\}}(t) \right) + g(t, y). \quad (33)$$

Specifically, $\mathbf{L}^{\{m\}}(t)$ represents the diffusion operator in the m spatial direction, $\phi^{\{m\}}(t)$ the corresponding directional boundary conditions and source terms, and $g(t, y)$ is a nonlinear term.

Fractional step Runge–Kutta (FSRK) methods [9–12] solve (33) in the al-

ternating implicit fashion

$$Y_i = y_n + h \sum_{j=1}^i a_{i,j}^{\{m_j\}} \left(\mathbf{L}^{\{m_j\}}(T_j) Y_j + \phi^{\{m_j\}}(T_j) \right) + h \sum_{j=1}^{i-1} a_{i,j}^{\{0\}} g(T_j, Y_j),$$

$$y_{n+1} = y_n + h \sum_{j=1}^s b_j^{\{m_j\}} \left(\mathbf{L}^{\{m_j\}}(T_j) Y_j + \phi^{\{m_j\}}(T_j) \right) + h \sum_{j=1}^s b_j^{\{0\}} g(T_j, Y_j),$$

where $T_j = t_n + c_j^{\{m_j\}}$. We note that only one dimension m_j is associated with each stage j , and that each stage solution requires solving a linear system corresponding to a one-dimensional problem. As stated in [9], FSRK are a special case of the additive Runge–Kutta (ARK) method. The GARK formulation provides a simpler and more compact representation. First, let us define the index sets

$$S_q = \{\ell \in \{1, \dots, s\} \mid m_\ell = q\} = \{\ell_{q,1}, \dots, \ell_{q,s^{\{q\}}}\}, \quad q = 1, \dots, N. \quad (34)$$

This provides a way to map stage $i \in \{1, \dots, s\}$ of a FSRK method to a stage $j \in \{1, \dots, s^{\{q\}}\}$ in partition q of a GARK method. Now the GARK coefficients can be expressed as

$$\mathbf{A}^{\{q,m\}} = \left[a_{\ell_{q,i}, \ell_{m,j}}^{\{m\}} \right]_{i=1, \dots, s^{\{q\}}}^{j=1, \dots, s^{\{m\}}}, \quad \mathbf{b}^{\{q\}} = \left[b_{\ell_{q,i}}^{\{q\}} \right]_{i=1, \dots, s^{\{q\}}}, \quad q, m = 1, \dots, N, \quad (35)$$

with partition zero using the coefficients

$$\mathbf{A}^{\{0,0\}} = \left[a_{i,j}^{\{0\}} \right]_{i=1, \dots, s}^{j=1, \dots, s}, \quad \mathbf{A}^{\{0,q\}} = \left[a_{i,\ell_{q,j}}^{\{0\}} \right]_{i=1, \dots, s}^{j=1, \dots, s^{\{q\}}},$$

$$\mathbf{A}^{\{q,0\}} = \left[a_{\ell_{q,i}, j}^{\{0\}} \right]_{i=1, \dots, s^{\{q\}}}^{j=1, \dots, s}, \quad \mathbf{b}^{\{0\}} = \left[b_i^{\{0\}} \right]_{i=1, \dots, s}, \quad q = 1, \dots, N.$$

Example 1. Consider the second order FSRK used in [41, Section 5.1] with coefficients

$$\mathbf{A}^{\{1\}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{A}^{\{2\}} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix},$$

$$\mathbf{b}^{\{1\}T} = [0 \ 1 \ 0], \quad \mathbf{b}^{\{2\}T} = [\frac{1}{2} \ 0 \ \frac{1}{2}].$$

This method is for system with $N = 2$ partitions, and we will ignore partition zero, i.e., $g(t, y) = 0$. There are $s = 3$ stages which map to partition 1 or 2 by

$$m_j = \begin{cases} 1, & j = 2, \\ 2, & j = 1 \text{ or } 3. \end{cases}$$

To cast this scheme into the GARK framework, we compute the index sets

(34):

$$S_1 = \{\ell_{1,1}\} = \{2\}, \quad S_2 = \{\ell_{2,1}, \ell_{2,2}\} = \{1, 3\}.$$

Thus, $s^{\{1\}} = |S_1| = 1$ and $s^{\{2\}} = |S_2| = 2$. By (35), the GARK tableau is given by

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

Note that the FSRK formulation requires columns of zeros in $\mathbf{A}^{\{1\}}$ and $\mathbf{A}^{\{2\}}$, whereas the GARK formulation is more compact. By reading the diagonal blocks of the GARK tableau, we can immediately see this method couples the implicit midpoint method with the implicit trapezoidal method.

7. High-order ADI-GARK methods

In this section we develop two new ADI-GARK methods (Definition 2) of order three and four. The GARK order conditions (A.1) particularized to an IMIM-GARK method with structure (7) lead to the following result.

Theorem 1 (Order conditions for the special class of IMIM-GARK). *Consider an IMIM-GARK method of type (7) that satisfies the internal consistency condition (4). Then we have the following:*

- The method has order $p \in \{1, 2, 3\}$ iff each component Runge–Kutta method $(\mathbf{A}^{\{D\}}, \mathbf{b}^{\{D\}}, \mathbf{c}^{\{D\}})$, $(\mathbf{A}^{\{L\}}, \mathbf{b}^{\{D\}}, \mathbf{c}^{\{D\}})$, and $(\mathbf{A}^{\{U\}}, \mathbf{b}^{\{D\}}, \mathbf{c}^{\{D\}})$ has order at least p .
- The method has order $p = 4$ iff each component Runge–Kutta method $(\mathbf{A}^{\{D\}}, \mathbf{b})$, $(\mathbf{A}^{\{L\}}, \mathbf{b}^{\{D\}})$, and $(\mathbf{A}^{\{U\}}, \mathbf{b}^{\{D\}})$ has order at least 4, and, in addition, the following coupling order conditions are satisfied:

$$\mathbf{b}^{\{D\}T} \mathbf{A}^{\{D\}} \mathbf{A}^{\{U\}} \mathbf{c}^{\{D\}} = \frac{1}{24}, \quad \mathbf{b}^{\{D\}T} \mathbf{A}^{\{U\}} \mathbf{A}^{\{D\}} \mathbf{c}^{\{D\}} = \frac{1}{24}, \quad (36a)$$

$$\mathbf{b}^{\{D\}T} \mathbf{A}^{\{L\}} \mathbf{A}^{\{U\}} \mathbf{c}^{\{D\}} = \frac{1}{24}, \quad \mathbf{b}^{\{D\}T} \mathbf{A}^{\{U\}} \mathbf{A}^{\{L\}} \mathbf{c}^{\{D\}} = \frac{1}{24}, \quad (36b)$$

$$\mathbf{b}^{\{D\}T} \mathbf{A}^{\{D\}} \mathbf{A}^{\{L\}} \mathbf{c}^{\{D\}} = \frac{1}{24}, \quad \mathbf{b}^{\{D\}T} \mathbf{A}^{\{L\}} \mathbf{A}^{\{D\}} \mathbf{c}^{\{D\}} = \frac{1}{24}. \quad (36c)$$

PROOF. We set $\mathbf{b}^{\{\sigma\}} = \mathbf{b}^{\{D\}}$ and $\mathbf{A}^{\{\sigma, \mu\}} \mathbf{1}_{s^{\{\mu\}}} = \mathbf{c}^{\{D\}}$ for all $\sigma, \mu \in \{L, D, U\}$ in (A.1). If the base methods satisfy their own order conditions, no third order coupling conditions remain. At order four, the only remaining coupling conditions are listed in (36). \square

Remark 5. The order conditions for ADI-GARK methods of Definition 2, as well as those for parallel ADI-GARK methods of Definition 3, reduce to (36a), since (36b) and (36c) are redundant.

7.1. (Parallel) ADI-GARK method of order 3

For the derivation of a third order ADI-GARK method, we start by selecting an implicit scheme. We use the optimal, L-stable, 4 stage ESDIRK method described in [36, Section 5.1.1]. For the explicit counterpart, it must use the $\mathbf{b}^{\{I\}}$ and $\mathbf{c}^{\{I\}}$ coefficients, leaving $\mathbf{A}^{\{E\}}$ as free parameters. Theorem 1 requires the explicit method to satisfy classical order conditions up to order three, and we also impose the simplifying assumption $D(1)$ [30, page 208]. The one remaining parameter in $\mathbf{A}^{\{E\}}$ is determined by

$$\mathbf{b}^{\{I\}T} \mathbf{A}^{\{E\}} \mathbf{A}^{\{E\}} \mathbf{c}^{\{I\}} = \frac{5}{268}.$$

This ensures the linear stability function satisfies $|R([z, z])| \leq 1$ for all z in the left-half plane. This stability region is plotted in fig. 1a, and the method coefficients are listed in table 1. By remark 2, the coefficients also define a third order parallel ADI-GARK method.

$$\begin{aligned} \mathbf{A}^{\{I\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ \gamma & \gamma & 0 & 0 \\ \frac{215\gamma+424}{2624-1536\gamma} & \frac{264-841\gamma}{1536\gamma+448} & \gamma & 0 \\ \frac{2\gamma+1}{4\gamma+8} & \frac{31-14\gamma}{352-900\gamma} & \frac{320\gamma+224}{575-477\gamma} & \gamma \end{bmatrix}, \\ \mathbf{A}^{\{E\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2\gamma & 0 & 0 & 0 \\ \frac{12526987\gamma+655304}{8876160\gamma+7175968} & \frac{15(215\gamma+152)}{2144(92\gamma-9)} & 0 & 0 \\ \frac{2370311\gamma-563481}{134(17071\gamma+921)} & \frac{380783-137789\gamma}{134(17727\gamma-15511)} & \frac{1000-304\gamma}{1371\gamma+379} & 0 \end{bmatrix}, \\ \mathbf{b}^{\{I\}T} &= \begin{bmatrix} \frac{2\gamma+1}{4\gamma+8} & \frac{31-14\gamma}{352-900\gamma} & \frac{320\gamma+224}{575-477\gamma} & \gamma \end{bmatrix}, \\ \mathbf{c}^{\{I\}T} &= [0 \quad 2\gamma \quad \frac{\gamma+2}{4} \quad 1]. \end{aligned}$$

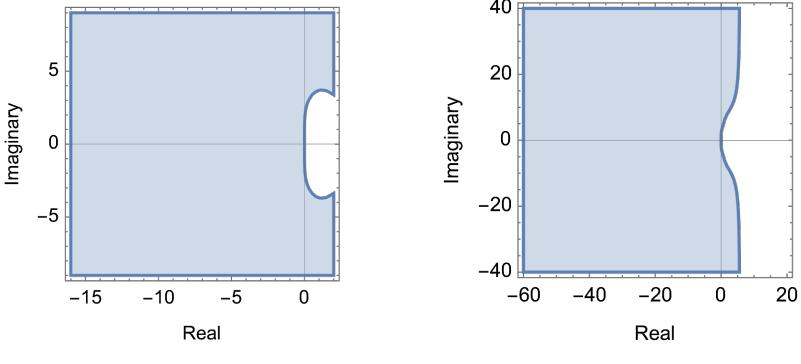
Table 1: Coefficients of the new third order (parallel) ADI-GARK method. Here $\gamma \approx 0.43586652150845900$ is the middle root of the polynomial $0 = -1 + 9\gamma - 18\gamma^2 + 6\gamma^3$.

7.2. (Parallel) ADI-GARK method of order 4

We follow a similar method derivation process for a fourth order ADI-GARK method. The implicit part is the L-stable method ESDIRK4(3)6L[2]SA from [36, Table 16]. This must be paired with a six stage explicit Runge–Kutta method of order four. Again, we enforce the first column simplifying assumption $D(1)$ but now need the coupling conditions (36). This leaves four unspecified coefficients that we use to control the stability. We minimize the value of $|R([z, z])|$ at a sample of points along the imaginary axis so that it is stable in the entire left-half plane. Table 2 gives the resulting coefficients, and fig. 1b plots its stability.

$$\begin{aligned}
\mathbf{A}^{\{1\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1-\sqrt{2}}{8} & \frac{1-\sqrt{2}}{8} & \frac{1}{4} & 0 & 0 \\ \frac{5-7\sqrt{2}}{64} & \frac{5-7\sqrt{2}}{64} & 0 & 0 & 0 \\ \frac{-54539\sqrt{2}-13796}{125000} & \frac{-54539\sqrt{2}-13796}{125000} & \frac{7(\sqrt{2}+1)}{32} & 0 & 0 \\ \frac{1181-987\sqrt{2}}{13782} & \frac{1181-987\sqrt{2}}{13782} & \frac{132109\sqrt{2}+506605}{437500} & \frac{166(376\sqrt{2}-97)}{109375} & 0 \\ \frac{1181-987\sqrt{2}}{13782} & \frac{1181-987\sqrt{2}}{13782} & \frac{47(1783\sqrt{2}-267)}{273343} & \frac{16(\frac{109375}{-3525\sqrt{2}+22922})}{571953} & \frac{15625(-376\sqrt{2}-97)}{90749876} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}, \\
\mathbf{A}^{\{E\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{b}^{\{1\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & 0 & 0 \end{bmatrix}, \\
\mathbf{c}^{\{1\}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{2-\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} & 0 & 0 \\ \frac{5-\sqrt{2}}{8} & \frac{5-\sqrt{2}}{8} & \frac{5-\sqrt{2}}{8} & 0 & 0 \\ \frac{1-\sqrt{2}}{4} & \frac{1-\sqrt{2}}{4} & \frac{1-\sqrt{2}}{4} & 0 & 0 \end{bmatrix}.
\end{aligned}$$

Table 2: Coefficients of the new fourth order (parallel) ADI-GARK method.



(a) Order 3 ADI-GARK method from table 1 (b) Order 4 ADI-GARK method from table 2

Figure 1: The linear stability regions for new ADI-GARK methods applied to a two-way partitioned system with partitions of equal stiffness. That is, the $z \in \mathbb{C}$ such that $|R([z, z])| \leq 1$.

8. Numerical experiments

We test the accuracy of the newly developed ADI-GARK methods on two parabolic PDEs from [51, Section 7]. The first is the two-dimensional problem

$$\begin{aligned} u_t &= u_{xx} + u_{yy} + h(x, y, t) \\ h(x, y, t) &= e^t(1-x)x(1-y)y + e^t \left((x + \frac{1}{3})^2 + (y + \frac{1}{4})^2 - 4 \right) \\ &\quad + 2e^t(1-x)x + 2e^t(1-y)y, \end{aligned} \quad (37)$$

posed on the unit square $[0, 1] \times [0, 1]$. Boundary and initial conditions come from the exact solution

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

The second problem is three-dimensional and reads

$$\begin{aligned} u_t &= u_{xx} + u_{yy} + u_{zz} + g(x, y, z, t), \\ g(x, y, z, t) &= e^t(1-x)x(1-y)y(1-z)z + 2e^t(1-x)x(1-y)y \\ &\quad + 2e^t(1-x)x(1-z)z + 2e^t(1-y)y(1-z)z - 6e^t \\ &\quad + e^t \left((x + \frac{1}{3})^2 + (y + \frac{1}{4})^2 + (z + \frac{1}{2})^2 \right), \end{aligned} \quad (38)$$

on the unit cube $[0, 1] \times [0, 1] \times [0, 1]$. For (38) the exact solution is

$$\begin{aligned} u(x, y, z, t) &= e^t(1-x)x(1-y)y(1-z)z \\ &\quad + e^t \left((x + \frac{1}{3})^2 + (y + \frac{1}{4})^2 + (z + \frac{1}{2})^2 \right). \end{aligned}$$

For both (37) and (38), we discretize spatial derivatives with second order central finite differences on a uniform mesh with N_p points in each direction. Given the exact solutions are quadratic in space, this ensures there are no spatial errors. The 2D problem is partitioned such that $f^{\{1\}}$ is the discretization of the directional derivative along the x -axis and $f^{\{2\}}$ discretizes $u_{yy} + h(x, y, t)$. Similarly, in the 3D problem, $f^{\{1\}}$ and $f^{\{2\}}$ are finite difference operators for u_{xx} and u_{yy} , respectively, while $f^{\{3\}}$ represents the remaining terms $u_{zz} + g(x, y, z, t)$. The integration timespan is $[0, 1]$, and temporal error is measured in the ℓ^2 norm with respect to the exact solution at $t = 1$ evaluated at the mesh points.

Figures 2 and 3 show convergence plots for the methods documented in section 7 applied to the 2D and 3D problems respectively when the ADI structure in definition 2 is considered. The uniform mesh and the singly-diagonally implicit structure of the method allows us to use the same matrix factorization when computing different directional stages. For small values of N_p the methods work at their nominal order of convergence. However, numerical convergence rates in figs. 2a, 2b and 3b indicate that as the mesh size gets smaller, the problem becomes stiffer and numerical order reduction is observed. An order reduction in the classical convergence order of one-step methods when applied to PDE problems with time dependent boundary conditions is generally present. This can be seen in the papers by Ostermann and Roche [39] and by Lubich and Ostermann [37, 38], where it is shown that for Runge–Kutta methods and Rosenbrock-type methods an order reduction takes place even in the case of time-independent boundary conditions. The order reduction is more dramatic for the case of time-dependent BCs. Besides in case of splitting methods (e.g. the GARK methods here considered) the effects of the order reduction are quite often more pronounced due to the effect of the additional errors introduced in the splitting terms. See, e.g. [20], where some convergence results about the PDE-order (in several L_p norms) of some splitting W-methods are given.

Figure 4 shows convergence results for methods of orders 3 and 4 when the parallel ADI scheme described in definition 3 is used on the 2D problem (37). This scheme has the added computational benefit that directional stages at the same time argument $t_n + c_i h$ can be computed in parallel. For the set of mesh sizes used in this experiment, we observe the classical order of convergence for the methods.

We perform efficiency experiments using the new methods in section 7. We compare the runtime and the error of the final solution against GLM-ADI methods reported in [51]. In order to highlight the benefits of design strategies utilized for creating the new methods, we also compare them with a generic fourth order IMEX method from [49, Example 3] used in ADI mode. Figure 5 shows the error versus cputime for $N_p \in \{16, 32\}$ on the 2D test problem (37). We observe good performance for the ADI-GARK methods, only being outperformed by the GLM-ADI 4. We also note that, owing to its optimized error, the ADI-GARK 3 method shows performance close to that of the IMEX-RK 4 method.

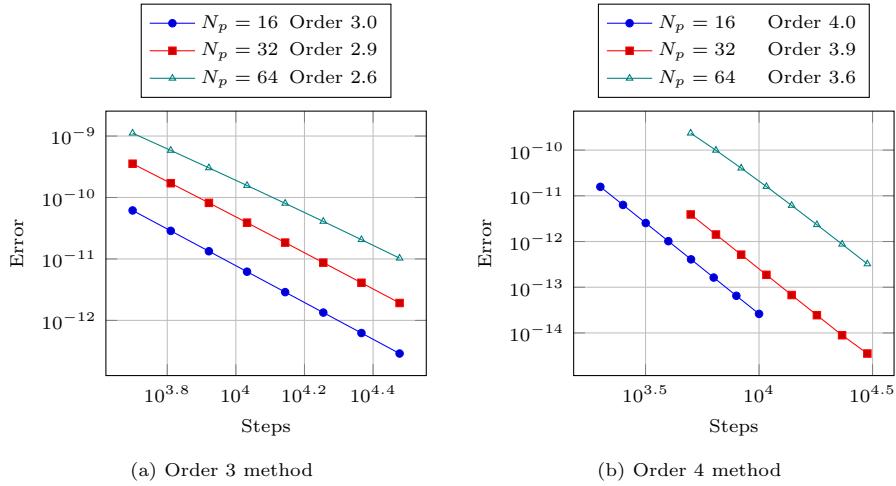


Figure 2: Convergence plots for ADI-GARK methods on the 2D test problem (37).

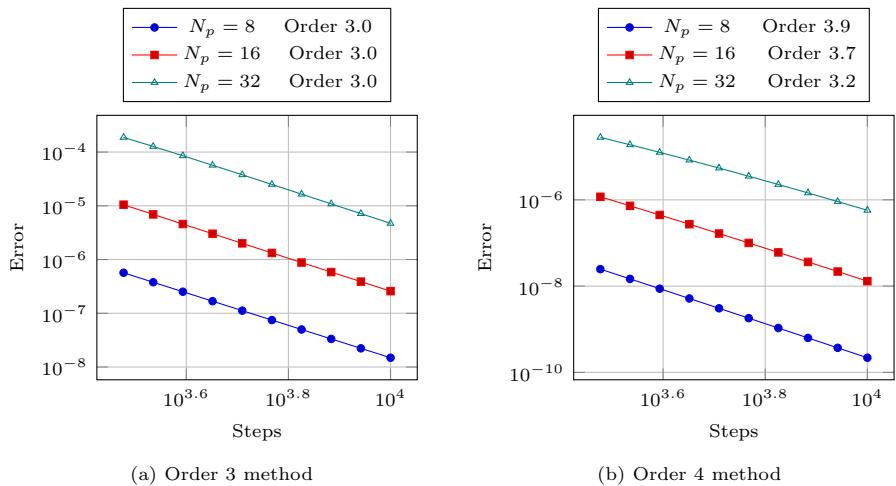


Figure 3: Convergence plots for ADI-GARK methods on the 3D test problem (38).

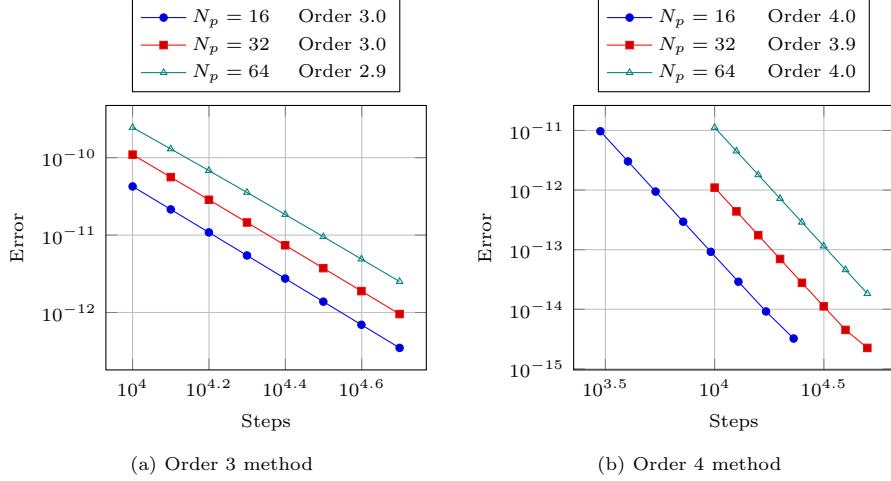


Figure 4: Convergence plots for parallel ADI-GARK methods on the 2D test problem (37).

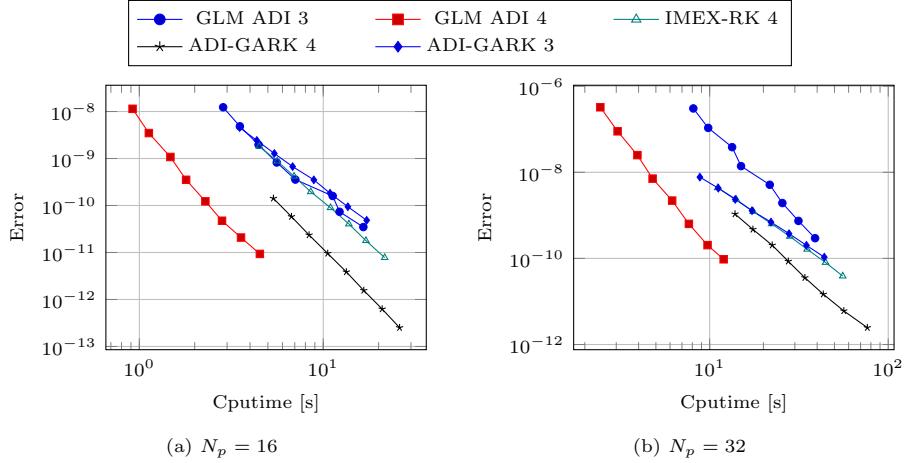


Figure 5: Error versus cputime for ADI-GARK methods on the 2D test problem (37) compared to GLM ADI methods from [51] and an ADI method created from the IMEX pair reported in [49, Example 3].

9. Conclusions

This work introduces the implicit-implicit GARK family of methods in the general-structure additive Runge–Kutta framework. The IMIM-GARK family is of interest since it provides the general computational template for implicit time integration based on splitting. Existing partitioned approaches such as fractional step, alternating direction implicit integration, operator splitting, and locally one dimensional integration, are formulated as IMIM-GARK methods. All these methods can be studied in a unified way using the order conditions and stability analyses provided herein. New splitting methods of (classical) order three and four with optimized stability and error constants are developed using the IMIM-GARK framework. Numerical experiments verify the accuracy and the efficiency of these new schemes.

References

- [1] A. Arrarás, K.J. in ’t Hout, W. Hundsdorfer, L. Portero, Modified Douglas splitting methods for reaction–diffusion equations, *BIT Numerical Mathematics* 57 (2017) 261–285.
- [2] U. Ascher, S. Ruuth, R. Spiteri, Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations, *Applied Numerical Mathematics* 25 (1997) 151–167.
- [3] A. Augustine, A. Sandu, MATLODE: A Matlab suite for ODE integration and sensitivity analysis, *Submitted to ACM TOMS* (2017).
- [4] R.M. Beam, R.F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law form, *Journal of Computational Physics* 22 (1976) 87–110.
- [5] S. Boscarino, Error analysis of IMEX Runge–Kutta methods derived from differential-algebraic systems, *SIAM Journal on Numerical Analysis* (2007).
- [6] S. Boscarino, On an accurate third order implicit-explicit Runge–Kutta method for stiff problems, *Applied Numerical Mathematics* 59 (2009) 1515–1528.
- [7] M. Braś, G. Izzo, Z. Jackiewicz, Accurate Implicit–Explicit General Linear Methods with Inherent Runge–Kutta Stability, *Journal of Scientific Computing* 70 (2017) 1105–1143.
- [8] B. Bujanda, J. Jorge, Stability results for fractional-step discretizations of time dependent coefficient evolutionary problems, *Applied Numerical Mathematics* 38 (2001) 69–86.
- [9] B. Bujanda, J. Jorge, Fractional-step Runge–Kutta methods for time dependent coefficient parabolic problems, *Applied Numerical Mathematics* 45 (2003) 99–122.

- [10] B. Bujanda, J. Jorge, Efficient linearly implicit methods for nonlinear multidimensional parabolic problems, *Journal of Computational and Applied Mathematics* 164/165 (2004) 159–174.
- [11] B. Bujanda, J. Jorge, Stability results for linearly implicit fractional-step discretizations of nonlinear time dependent parabolic problems, *Applied Numerical Mathematics* 56 (2006) 1061–1076.
- [12] B. Bujanda, J. Jorge, Order conditions for linearly implicit fractional step Runge–Kutta methods, *IMA Journal of Numerical Analysis* 27 (2007) 781–797.
- [13] A. Cardone, Z. Jackiewicz, A. Sandu, H. Zhang, Extrapolated IMEX Runge–Kutta methods, *Mathematical Modelling and Analysis* 19 (2014) 18–43.
- [14] A. Cardone, Z. Jackiewicz, A. Sandu, H. Zhang, Extrapolation-based implicit-explicit general linear methods, *Numerical Algorithms* 65 (2014) 377–399.
- [15] E. Constantinescu, A. Sandu, Extrapolated implicit-explicit time stepping, *SIAM Journal on Scientific Computing* 31 (2010) 4452–4477.
- [16] G. Cooper, A. Sayfy, Additive Runge–Kutta methods for stiff ordinary differential equations, *Mathematics of Computation* 40 (1983) 207–218.
- [17] I.J.D. Craig, A.D. Sneyd, An alternating-direction implicit scheme for parabolic equations with mixed derivatives, *Computational and Applied Mathematics* 16 (1988) 341–350.
- [18] J. Douglas, On the numerical integration of $u_{x,x} + u_{y,y} = u_t$ by implicit methods, *SIAM* 3 (1955) 42–65.
- [19] J. Douglas, H.H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, *Transactions of the American Mathematical Society* 82 (1956) 421–439.
- [20] S. González-Pinto, E. Hairer, D. Hernández-Abreu, Convergence in ℓ_2 and ℓ_∞ norm of one-stage AMF-W-methods for parabolic problems, *SIAM Journal on Numerical Analysis* 58 (2020) 1117–1137.
- [21] S. González-Pinto, E. Hairer, D. Hernández-Abreu, Power boundedness in the maximum norm of stability matrices for adi methods, *BIT Numerical Mathematics* 61 (2021) 805–827.
- [22] S. González-Pinto, E. Hairer, D. Hernández-Abreu, S. Pérez-Rodríguez, AMF-type W-methods for parabolic problems with mixed derivatives, *SIAM Journal on Scientific Computing* 40 (2018) A2905–A2929.

- [23] S. González-Pinto, E. Hairer, D. Hernández-Abreu, S. Pérez-Rodríguez, PDE-W-methods for parabolic problems with mixed derivatives, *Numerical Algorithms* 78(3) (2018) 957–981.
- [24] S. González-Pinto, D. Hernández-Abreu, Splitting-methods based on approximate matrix factorization and Radau-IIA formulas for the time integration of advection diffusion reaction PDEs, *Applied Numerical Mathematics* 104 (2016) 166–181.
- [25] S. González-Pinto, D. Hernández-Abreu, Convergence in the maximum norm of ADI-type methods for parabolic problems, *Applied Numerical Mathematics*. <https://doi.org/10.1016/j.apnum.2021.09.007> (2021).
- [26] S. González-Pinto, D. Hernández-Abreu, S. Pérez-Rodríguez, AMFR-W-methods for parabolic problems with mixed derivatives. applications to the Heston model, *Journal of Computational and Applied Mathematics* 387 (2021) 112518.
- [27] M. Günther, C. Hachtel, A. Sandu, Multirate GARK schemes for multi-physics problems, in: 10th International Conference on Scientific Computing in Electrical Engineering.
- [28] M. Günther, A. Sandu, Multirate generalized additive Runge–Kutta methods, *Numerische Mathematik* 133 (2016) 497–524.
- [29] E. Hairer, Order conditions for numerical methods for partitioned ordinary differential equations, *Numerische Mathematik* 36 (1981) 431–445.
- [30] E. Hairer, G. Wanner, S.P. Nørsett, *Solving Ordinary Differential Equations I: Nonstiff Problems*, number 8 in Springer Series in Computational Mathematics, Springer–Verlag Berlin Heidelberg, 2nd edition, 1993.
- [31] H.V. Henderson, S.R. Searle, The vec-permutation matrix, the vec operator and Kronecker products: a review, *Linear and Multilinear Algebra* 9 (1981) 271–288.
- [32] K.J. in 't Hout, B.D. Welfert, Unconditional stability of second-order ADI schemes applied to multi-dimensional diffusion equations with mixed derivative terms., *Appl. Numer. Math.* 59(3-4) (2009) 677–692.
- [33] W. Hundsdorfer, Accuracy and stability of splitting with stabilizing corrections, *Applied Numerical Mathematics* 42 (2002) 213 – 233. Numerical Solution of Differential and Differential-Algebraic Equations, 4-9 September 2000, Halle, Germany.
- [34] W. Hundsdorfer, J. Verwer, Numerical solution of time-dependent advection-diffusion-reaction equations, volume 33 of *Springer Series in Computational Mathematics*, Springer-Verlag, Berlin, 2003.

- [35] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, *Appl. Numer. Math.* 44 (2003) 139–181.
- [36] C.A. Kennedy, M.H. Carpenter, Diagonally implicit Runge–Kutta methods for ordinary differential equations. A review, Technical Report NASA/TM-2016-219173, NASA, 2016.
- [37] C. Lubich, A. Ostermann, Linearly implicit time discretization of non-linear parabolic equations, *IMA Journal of Numerical Analysis* 15 (1995) 555–583.
- [38] C. Lubich, A. Ostermann, Runge–Kutta approximation of quasi-linear parabolic equations, *Mathematics of Computation* 64 (1995) 601–627.
- [39] A. Ostermann, M. Roche, Runge–Kutta methods for partial differential equations and fractional orders of convergence, *Mathematics of Computation* 59 (1992) 403–420.
- [40] D. Peaceman, H. Rachford, The numerical solution of parabolic and elliptic differential equations, *Journal of Society for Industrial and Applied Mathematics* 3 (1955) 28–42.
- [41] L. Portero, J.C. Jorge, B. Bujanda, Avoiding order reduction of fractional step Runge–Kutta discretizations for linear time dependent coefficient parabolic problems, *Appl. Numer. Math.* 48 (2004) 409–424.
- [42] P. Rentrop, Partitioned Runge–Kutta methods with stepsize control and stiffness detection, *Numerische Mathematik* 47 (1985) 545–564.
- [43] J. Rice, Split Runge–Kutta methods for simultaneous equations, *Journal of Research of the National Institute of Standards and Technology* 64 (1960).
- [44] S. Roberts, J. Loffeld, A. Sarshar, C.S. Woodward, A. Sandu, Implicit multirate GARK methods, *Journal of Scientific Computing* 87 (2021) 4.
- [45] S. Roberts, A. Sarshar, A. Sandu, Coupled multirate infinitesimal GARK methods for stiff differential equations with multiple time scales, *SIAM Journal on Scientific Computing* 42 (2020) A1609–A1638.
- [46] A. Sandu, A class of multirate infinitesimal GARK methods, *SIAM Journal on Numerical Analysis* 57 (2019) 2300–2327.
- [47] A. Sandu, Convergence results for implicit-explicit general linear methods, *Applied Numerical Mathematics* 156 (2020).
- [48] A. Sandu, M. Guenther, S. Roberts, Linearly implicit GARK schemes, *Applied Numerical Mathematics* 161 (2021) 286–310.
- [49] A. Sandu, M. Günther, A generalized-structure approach to additive Runge–Kutta methods, *SIAM Journal on Numerical Analysis* 53 (2015) 17–42.

- [50] A. Sarshar, S. Roberts, A. Sandu, Design of high-order decoupled multirate GARK schemes, *SIAM Journal on Scientific Computing* 41 (2019) A816–A847.
- [51] A. Sarshar, S. Roberts, A. Sandu, Alternating directions implicit integration in a general linear method framework, *Journal of Computational and Applied Mathematics* 387 (2021) 112619. Numerical Solution of Differential and Differential-Algebraic Equations. Selected Papers from NUMDIFF-15.
- [52] M. Schneider, J. Lang, W. Hundsdorfer, Extrapolation-based superconvergent implicit-explicit peer methods with A-stable implicit part, *Journal of Computational Physics* 367 (2018) 121–133.
- [53] M. Schneider, J. Lang, R. Weiner, Super-convergent implicit-explicit peer methods with variable step sizes, *Journal of Computational and Applied Mathematics* 387 (2021) 112501. Numerical Solution of Differential and Differential-Algebraic Equations. Selected Papers from NUMDIFF-15.
- [54] B. Soleimani, O. Knoth, R. Weiner, IMEX peer methods for fast-wave–slow-wave problems, *Applied Numerical Mathematics* 118 (2017) 221–237.
- [55] B. Soleimani, R. Weiner, Superconvergent IMEX peer methods, *Applied Numerical Mathematics* 130 (2018) 70–85.
- [56] G. Strang, On the construction and comparison of difference schemes, *SIAM Journal on Numerical Analysis* 5 (1968) 506–517.
- [57] E.L. Tan, Unconditionally stable LOD–FDTD method for 3-D Maxwell’s equations, *IEEE Microwave and Wireless Components Letters* 17 (2007) 85–87.
- [58] E.L. Tan, Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods, *IEEE Transactions on Antennas and Propagation* 56 (2008) 170–177.
- [59] E.L. Tan, Fundamental implicit FDTD schemes for computational electromagnetics and educational mobile apps, *Progress In Electromagnetics Research* 168 (2020) 39–59.
- [60] G.M. Tanner, Generalized additive Runge–Kutta methods for stiff odes, Ph.D. thesis, University of Iowa, 2018.
- [61] R. Weiner, M. Arnold, P. Rentrop, K. Strehmel, Partitioning strategies in Runge–Kutta type methods, *IMA Journal on Numerical Analysis* 13 (1993) 303–319.
- [62] N. Yanenko, *The Method of Fractional-Steps*, Springer, Berlin Heidelberg NewYork, 1971.
- [63] H. Yoshida, Construction of higher order symplectic integrators, *Physics Letters* 150 (1990) 262–268.

- [64] H. Zhang, A. Sandu, FATODE: A library for forward, adjoint and tangent linear integration of stiff systems, SIAM Journal on Scientific Computing 36 (2014) C504–C523.
- [65] H. Zhang, A. Sandu, S. Blaise, Partitioned and implicit-explicit general linear methods for ordinary differential equations, Journal of Scientific Computing 61 (2014) 119–144.
- [66] H. Zhang, A. Sandu, P. Tranquilli, Application of approximate matrix factorization to high-order linearly-implicit Runge–Kutta methods, Journal of Computational and Applied Mathematics 286 (2015) 196–210.
- [67] E. Zharovsky, A. Sandu, H. Zhang, A class of IMEX two-step Runge–Kutta methods, SIAM Journal on Numerical Analysis 53 (2015) 321–341.

Appendix A. GARK order conditions

General order conditions for partitioned GARK methods (2) are described in [49, Theorem 2.6]. Let

$$\mathbf{A}^{\{\sigma,\nu\}} \mathbb{1}_{s^{\{\nu\}}} = \mathbf{c}^{\{\sigma,\nu\}} \quad \forall \sigma, \nu,$$

and assume $\mathbf{c}^{\{\sigma\}} = \mathbf{c}^{\{\sigma,\sigma\}}$ for $\sigma = 1, \dots, N$. This is weaker than the internal consistency condition (4). Then the order condition up to order four are

$$\mathbf{b}^{\{\sigma\}T} \mathbb{1}_{s^{\{\sigma\}}} = 1 \quad \forall \sigma, \quad (\text{order 1}) \quad (\text{A.1a})$$

$$\mathbf{b}^{\{\sigma\}T} \mathbf{c}^{\{\sigma,\nu\}} = \frac{1}{2}, \quad \forall \sigma, \nu, \quad (\text{order 2}) \quad (\text{A.1b})$$

$$\mathbf{b}^{\{\sigma\}T} \left(\mathbf{c}^{\{\sigma,\nu\}} \times \mathbf{c}^{\{\sigma,\mu\}} \right) = \frac{1}{3}, \quad \forall \sigma, \nu, \mu, \quad (\text{order 3}) \quad (\text{A.1c})$$

$$\mathbf{b}^{\{\sigma\}T} \mathbf{A}^{\{\sigma,\nu\}} \mathbf{c}^{\{\nu,\mu\}} = \frac{1}{6}, \quad \forall \sigma, \nu, \mu, \quad (\text{order 3}) \quad (\text{A.1d})$$

$$\mathbf{b}^{\{\sigma\}T} \left(\mathbf{c}^{\{\sigma,\lambda\}} \times \mathbf{c}^{\{\sigma,\mu\}} \times \mathbf{c}^{\{\sigma,\nu\}} \right) = \frac{1}{4}, \quad \forall \lambda, \sigma, \nu, \mu, \quad (\text{order 4}) \quad (\text{A.1e})$$

$$\left(\mathbf{b}^{\{\sigma\}} \times \mathbf{c}^{\{\sigma,\mu\}} \right)^T \mathbf{A}^{\{\sigma,\nu\}} \mathbf{c}^{\{\nu,\lambda\}} = \frac{1}{8}, \quad \forall \lambda, \sigma, \nu, \mu, \quad (\text{order 4}) \quad (\text{A.1f})$$

$$\mathbf{b}^{\{\sigma\}T} \mathbf{A}^{\{\sigma,\lambda\}} \left(\mathbf{c}^{\{\lambda,\mu\}} \times \mathbf{c}^{\{\lambda,\nu\}} \right) = \frac{1}{12}, \quad \forall \lambda, \sigma, \nu, \mu, \quad (\text{order 4}) \quad (\text{A.1g})$$

$$\mathbf{b}^{\{\sigma\}T} \mathbf{A}^{\{\sigma,\lambda\}} \mathbf{A}^{\{\lambda,\nu\}} \mathbf{c}^{\{\nu,\mu\}} = \frac{1}{24}, \quad \forall \lambda, \sigma, \nu, \mu, \quad (\text{order 4}) \quad (\text{A.1h})$$

where $\lambda, \sigma, \nu, \mu \in \{1, 2, \dots, N\}$.