



Self-Adjusting Multi-Rate Runge-Kutta Methods: Analysis and Efficient Implementation in An Open Source Framework

Bernhard Bachmann¹ · Luca Bonaventura²  · Francesco Casella³ · Soledad Fernández-García⁴ · Macarena Gómez-Mármol⁵ · Philip Hannebohm¹

Received: 17 July 2024 / Revised: 10 July 2025 / Accepted: 3 August 2025
© The Author(s) 2025

Abstract

We present an approach for the efficient implementation of self-adjusting multi-rate Runge-Kutta methods and we introduce a novel stability analysis, that covers the multi-rate extensions of all standard Runge-Kutta methods and allows to assess the impact of different interpolation methods for the latent variables and of the use of an arbitrary number of sub-steps for the active variables. The stability analysis applies successfully to the model problem typically used in the literature for multi-rate methods. Furthermore, we also propose a physically motivated model problem that can be used to assess stability to problems with purely imaginary eigenvalues and in situations closer to those arising in applications. Finally, we present an efficient implementation of multi-rate Runge-Kutta methods in the framework of the *OpenModelica* open-source modelling and simulation software. Results of several numerical experiments, performed with this implementation of the proposed methods, demonstrate the efficiency gains deriving from the use of the proposed multi-rate approach for physical modelling problems with multiple time scales.

✉ Luca Bonaventura
luca.bonaventura@polimi.it

Bernhard Bachmann
bernhard.bachmann@hsbi.de

Francesco Casella
francesco.casella@polimi.it

Soledad Fernández-García
soledad@us.es

Macarena Gómez-Mármol
macarena@us.es

Philip Hannebohm
philip.hannebohm@hsbi.de

¹ Faculty of Engineering and Mathematics, University of Applied Sciences and Arts, Bielefeld, Germany

² Dipartimento di Matematica, Politecnico di Milano, Milan, Italy

³ Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

⁴ Departamento de Ecuaciones Diferenciales y Análisis Numérico & IMUS, Universidad de Sevilla, Seville, Spain

⁵ Departamento de Ecuaciones Diferenciales y Análisis Numérico, Universidad de Sevilla, Seville, Spain

Keywords Multi-rate methods · Multi-scale problems · Stiff ODE problems · Runge-Kutta methods · ESDIRK methods

Mathematics Subject Classification 65L04 · 65L05 · 65L07 · 65M12 · 65M20

1 Introduction

We consider numerical methods for systems of Ordinary Differential Equations (ODEs) that can be partitioned into components y_s, y_f with slow and fast dynamics, respectively, as

$$y' = \begin{bmatrix} y'_s \\ y'_f \end{bmatrix} = \begin{bmatrix} f_s(y_s, y_f, t) \\ f_f(y_s, y_f, t) \end{bmatrix} = f(y, t) \quad (1)$$

for $t \in [0, T]$. For such systems, several methods have been proposed over the last 60 years, in which different time steps are employed for the slow and fast components, see e.g. [1, 12, 34]. These methods are generically known as multi-rate methods, as opposed to standard single-rate methods for ODEs. Variables y_s, y_f are also known in the literature as latent and active variables, respectively. More recently, a number of methods have been proposed that do not require *a priori* partitioning as in (1), but rather allow to identify fast and slow variables at runtime by their compliance, or lack of compliance, with a given error indicator, so that the system has form (1) with a right-hand side partition that can be different for each time discretization interval. A review of these methods is presented in [6], where a variant of this more general, self-adjusting multi-rate approach was introduced. The proposal in [6] was tailored on the specific implicit TR-BDF2 method [4, 20], along with a general stability analysis for one-step multi-rate methods.

The present work has three main goals. Firstly, we propose a novel approach that combines the self-adjusting multi-rate technique with standard time step adaptation methods. By marking as fast variables only a small percentage of the variables associated with the largest values of an error estimator, we obtain a more effective multi-rate implementation of a generic Runge-Kutta (RK) method. Whenever the global time step is sufficient to guarantee a given error tolerance for the slow variables, but not for the fast ones, the multi-rate procedure is employed to achieve uniform accuracy at reduced computational cost. This approach is similar to that used successfully in spatially adaptive finite element techniques, see e.g. [2, 3, 30, 31], where a fixed percentage of the simulation Degrees of Freedom (DOF) can be marked for refinement if error indicator values exceed a given tolerance. We then review the literature on the stability analysis of multi-rate methods and highlight the lack of a general analysis for multi-rate RK methods. We present a stability analysis for a general multi-rate RK method, that allows to study linear stability for a generic interpolation procedure and for an arbitrary number of sub-steps for the active components. The analysis covers the cases of Explicit RK methods (ERK), Diagonally Implicit RK methods (DIRK) and Singly Diagonally Implicit Runge-Kutta methods with Explicit first stage (ESDIRK). As discussed in the detailed review of the analyses of multi-rate methods available in the literature, a linear stability analysis in this very general framework does not appear to have been presented so far. We apply the analysis results to the typical 2-DOF model problem used in the literature to assess the stability of multi-rate extensions of classical RK methods and the impact of different interpolation methods for the latent variables. We show that the multi-rate extension does not entail any loss of stability in this case. Indeed, for problems with fast variables only mildly

coupled to the slow ones, the stability regions of the standard single rate RK methods are effectively increased in size by the multi-rate extension.

In order to apply this methodology also to a case that is more relevant for applications, we then propose a more physically motivated 4-DOF model problem. Also in this case, we verify that, in regimes of mild coupling between fast and slow variables and in the presence of sufficient dissipation, multi-rate approaches generally maintain the stability properties of the corresponding single-rate methods, while a loss of unconditional stability of implicit methods arises in the case of stronger coupling and purely imaginary eigenvalues.

Finally, we present the first implementation of multi-rate extensions of classical RK methods in an advanced open-source software package for industrial simulations. More specifically, we have implemented the proposed multi-rate approach in the framework of the OpenModelica software [11]. We present the results of the application of this implementation to a number of numerical benchmark problems. Due to their computational advantages, we will focus in particular on the application of multi-rate versions of higher order ESDIRK methods. A very comprehensive review of these methods was presented in [24, 25], where the properties and potential advantages of this class of methods are discussed in detail. The results demonstrate the efficiency gains deriving from the use of the proposed multi-rate approach for problems with multiple time scales. Furthermore, the results of the numerical experiments highlight how different optimization strategies should be applied in the multi-rate and single-rate case, in order to take full advantage of the multi-rate DOF reduction.

The outline of the paper is the following. In Section 2, the self-adjusting multi-rate approach is presented. In Section 3, a thorough review of the literature on multi-rate stability analysis is presented. In Section 4, a linear stability analysis of the proposed methods is performed, which considers an arbitrary number of sub-steps for the active variables. In Section 5, a specific model problem is considered, to which the previously introduced analysis template is applied. The resulting analysis highlights the impact on multi-rate stability of the different choices for the interpolation operator of the latent variables and of the number of sub-steps for the active variables. In Section 6, we briefly introduce the OpenModelica software package and describe the implementation of multi-rate methods in that context. In Section 7, numerical results are presented, which demonstrate the good performance of the proposed methods even against optimal implementations of single-rate solvers. In Section 8, we summarize our results and discuss possible future developments.

2 Effective Implementation of Multi-Rate Methods

Multi-rate methods have been defined in a number of previous papers. Here, we follow the general outline of [6], but we introduce several simplifications and extensions which allow to achieve a more efficient implementation. Consider the generic nonlinear system $y' = f(y, t)$, where $y : [0, T] \rightarrow \mathbb{R}^N$ is the solution of the continuous first order ODE problem and $f : \mathbb{R}^N \times [0, T] \rightarrow \mathbb{R}^N$ the ODE right-hand side, which is assumed to satisfy the usual regularity requirements to guarantee local existence and smoothness of the solutions. Let then $t_n, n \in \mathbb{N}$ denote a set of discrete time levels such that $t_0 = 0, t_{n+1} = t_n + h_n$ and h_n denotes a generic time step. At each time level n , two approximations u_{n+1} and \hat{u}_{n+1} of the solution $y(t_{n+1})$ are computed, with convergence orders p and \hat{p} , respectively. We set $q = \min(p, \hat{p})$. In the case of RK methods, approximation \hat{u}_{n+1} is usually obtained from an embedded method, see e.g. [18], to which we refer for all the general concepts on ODE

methods used in the following. An error estimator for the less accurate approximation is then given by $\|u_{n+1} - \hat{u}_{n+1}\|$.

In order to comply with assigned error tolerances τ^r , τ^a for relative and absolute errors, respectively, standard single-rate implementations of RK methods introduce for each time step $i = 1, \dots, N$ the quotient

$$\eta_i^{n+1} = \frac{|u_i^{n+1} - \hat{u}_i^{n+1}|}{\tau^r |u_i^{n+1}| + \tau^a} \tag{2}$$

and require that the inequality

$$\eta = \max_{i=1, \dots, N} \eta_i^{n+1} \leq 1$$

holds. An optimal choice of the new time step value for an adaptive single-rate implementation is then given by

$$h_{new} = h_n \eta^{-\frac{1}{q+1}}.$$

If $\eta \leq 1$ is satisfied, the solution is advanced with u_{n+1} and the new step size is chosen as $h_{n+1} = h_{new}$. Often some user defined safety parameter $\beta \in (0, 1)$ is introduced, so that the condition to accept the new time step will be $\eta \leq \beta$. Otherwise, the step is rejected and the computation is repeated with $h_n = h_{new}$. In practice, the optimal step size is calculated using appropriate safety factors, so that

$$h_{new} = h_n \min \left\{ \alpha_{max}, \max \left\{ \alpha_{min}, \alpha \eta^{-\frac{1}{q+1}} \right\} \right\}. \tag{3}$$

Reasonable default values of these safety factors can be selected as $\alpha_{max} = 1.2$, $\alpha_{min} = 0.5$, and $\alpha = 0.9$. It is important to remark that, in the application of these definitions in a multi-rate framework, the value of α_{min} has an impact on the maximum number of sub-steps chosen for the active components, so that it may have to be adjusted depending on the time scales involved in the specific problem to be solved.

In order to define the multi-rate extension of a given RK method, we denote by

$$u_{n+1} = \mathcal{S}(u_n, h_n), \quad \mathcal{S} : \mathbb{R}^N \times \mathbb{R}^+ \rightarrow \mathbb{R}^N$$

the global step of the basic single-rate method. We will denote as $\mathcal{V}_n^f \subset \mathbb{R}^N$ the linear subspace of fast variables at time level n . It is implicitly assumed that $d_n = \dim(\mathcal{V}_n^f) \ll N$. We also denote as $P_n^f : \mathbb{R}^N \rightarrow \mathbb{R}^{d_n}$ the projector on the linear subspace of fast variables, while $P_n^s : \mathbb{R}^N \rightarrow \mathbb{R}^{N-d_n}$ denotes the projection on the linear subspace of slow variables \mathcal{V}_n^s . For brevity, we will also denote $u_n^s = P_n^s u_n$ and $u_n^f = P_n^f u_n$. The operator

$$u_{n+1}^f = \mathcal{S}^f(u_n^f, w, h_n) : \mathbb{R}^{d_n} \times \mathbb{R}^{N-d_n} \rightarrow \mathbb{R}^{d_n}$$

will then denote the application of the basic method \mathcal{S} to the subsystem of $y' = f(y, t)$ obtained by projecting u_n onto \mathcal{V}_n^f and assuming that the remaining components of u_n belonging to \mathcal{V}_n^s are given by the given vector w . Furthermore, setting $\zeta = t_n + \tau h_n$ for each $\tau \in [0, 1]$, we will denote as $Q(\tau)$ a generic operator that interpolates known values of slow variables at time level ζ . Notice that, contrarily to what is stated in [28], the use of interpolation methods at this stage does not alter the one-step nature of the resulting multi-rate methods, as will be better explained in the following. This operator is used to provide intermediate values of the slow variables for the application of \mathcal{S}^f and can be given, for example, by the dense output approximation associated to a given ODE method, see e.g.

[18] for relevant examples. Given these definitions, the general multi-rate approach we will implement can be defined as in the following pseudo-code.

Multirate algorithm:

- 1) Compute a tentative global step $u_{n+1} = \mathcal{S}(u_n, h_n)$ and the additional approximation \hat{u}_{n+1} .
- 2) For each component $u_i^{n+1}, i = 1, \dots, N$, compute the error estimation η_i^{n+1} as described in expression (2).
- 3) Let $\tilde{\eta}_i^{n+1}$ denote the components of a vector obtained by sorting η_{n+1} in descending order and $s = s(i), i = 1, \dots, N$, the map that assigns to each component of η_{n+1} the index of its location in $\tilde{\eta}_{n+1}$. Define $m \in \{1, \dots, N\}$ as the only integer such that $m/N \leq \phi < (m + 1)/N$, where $\phi \in (0, 1)$ is a user defined parameter determining the maximum fraction of fast variables that will be allowed in the simulation. Define also the index subset

$$\mathcal{S}_n = \{i \in \{1, \dots, N\} : s(i) > m\}.$$

- 4) If

$$\eta_s = \max_{i \in \mathcal{S}_n} \eta_i^{n+1} > \beta,$$

reject the global time step and recompute it using the time step given by expression (3) with respect to η_s .

- 5) If

$$\eta_f = \max_{i \in \mathcal{S}_n^c} \eta_i^{n+1} \leq \beta,$$

accept the global time step and compute the next using the time step given by expression (3) with respect to η_s .

- 6) If $\eta_s \leq \beta$ and $\eta_f > \beta$, go multirate:

- 6.1) Define \mathcal{V}_n^f as the subspace of \mathbb{R}^N whose coordinates have indices i such that $\eta_i^{n+1} > \beta$ and $\mathcal{V}_n^s = \mathcal{V}_n^{f^c}$ and \mathcal{S}_n^f the corresponding index subset. Notice that $\mathcal{S}_n^f \subset \mathcal{S}_n^c$.
- 6.2) Partition the state space as $\mathbb{R}^N = \mathcal{V}_n^f \oplus \mathcal{V}_n^s$, and set $u_{n+1}^s = P_n^s u_{n+1}$;
- 6.3) Set $u_{n,0}^f = u_n^f$. Compute $u_{n,l+1}^f = \mathcal{S}^f(u_{n,l}^f, u_{n,l}^s, h_n^f)$ with local time steps determined using time step control based on expression (3) with respect to

$$\hat{\eta} = \max_{i \in \mathcal{S}_n^f} \eta_i^{n,l+1}.$$

- 6.4) Proceed until the time of the global time step has been reached. Apply the same logic for step rejection and acceptance (see 4) and 5)) with respect to $\hat{\eta}$. Compute necessary values of the slow variables at intermediate time levels using the interpolation operator $Q(\cdot)$. Let M_n denote the number of local steps taken.
- 6.5) Set $u_{n+1}^f = u_{n,M_n}^f$ and determine a new global time step based on expression (3) with respect to η_s , then go to 1).

As remarked previously, the value of α_{min} plays a role in determining the maximum number of sub-steps performed for the active variables. Therefore, it may need to be adjusted depending on specific accuracy or stability features of the method or problem under consideration. Furthermore, as discussed in Section 1, the set of active variables is chosen here in a way that has already proven to be successful in adaptive finite element techniques, see e.g. [2, 3, 30, 31]. This enables us to reduce or avoid conflicts between the global time step adaptation and the multi-rate strategy, which is instead employed to achieve the same level of accuracy as the corresponding single-rate implementation at a reduced computational cost.

Last, but not least, an important remark is due for the case of strongly non-linear ODEs and implicit Runge-Kutta multi-rate integration methods. The equations of implicit Runge-Kutta methods are usually solved using iterative Newton-Raphson methods, with an initial guess computed by extrapolation of the solution found at previous time steps. In case the RHS $f(y, t)$ of the ODEs is strongly non-linear, if the time step is too large this may cause convergence problems or even failure to converge, because the initial guess could be too far from the solution. This issue is much more critical for multi-rate integration, since the global time step is chosen without considering the errors of the variables belonging to the fast partition, so it is in general much larger than in the case of single-rate methods. For efficiency, it is therefore essential to limit the maximum number of Newton iterations to a relatively small value such as 20, after which a shorter time step should be attempted. Otherwise, a large amount of time and computational effort could be wasted in the futile attempt to achieve convergence from an initial guess which is too far from the solution. It is also advisable to choose conservative values of α_{max} , such as $\alpha_{max} = 1.2$, to avoid getting again into convergence issues at the next global time step.

3 Review of Multi-Rate Stability Analyses

The linear stability of multi-rate methods has been studied in a number of papers. The first contribution in this respect appears to be [12]. In this paper, the intrinsic difficulty of performing a stability analysis valid for the multi-rate case is acknowledged, showing that the typical reduction to a scalar problem is impossible in this case and that model problems with at least two degrees of freedom must be considered. A very limited stability result for the multi-rate explicit Euler method with linear interpolation is presented.

Other early attempts at similar stability analyses were presented in [15, 37, 45, 46]. The results of these papers were reviewed and found limited and inconclusive in [28], where a first attempt at a more complete analysis was presented. In [28] the impact on stability of the number of sub-steps employed for the fast variables is considered. However, this is done only for a combination of implicit and explicit Euler methods for the slow and fast components, respectively. Furthermore, in [28] methods using interpolation to provide the intermediate values of the slow components are excluded from consideration, based on the incorrect consideration that these methods result in effectively two-step approaches. However, this is not the case for the methods proposed in our work, as will be clear from the analysis presented in Section 4. Stability analyses for multi-rate versions of the BDF and linear multistep methods were then presented in [36, 50] and also tested on the same 2-DOF model problem.

Multi-rate methods based on implicit RK solvers and similar in spirit to the approach pursued here were first introduced in [41, 42] and their stability analysis was presented in [21]. More specifically, in [21] the multi-rate extension of the θ -method is analyzed, employing mostly linear interpolation. However, the stability results are only obtained under rather restrictive assumptions on the general system structure and for the case of two substeps only. Furthermore, the application to the 2-DOF model problem presents the results as a function of parameters that are specific for the θ -method only. In our previous paper [6], we generalized the approach of [21] to the case of the multi-rate version of the TR-BDF2 method, but still with the limitation to the two substep case.

More recently, the multi-rate extension of Generalized Additive Runge-Kutta methods has been proposed in [16] and a number of multi-rate GARK methods have been proposed in

the literature, see e.g. [17, 35, 38–40]. In [16], a very general nonlinear stability analysis was proposed for dissipative coercive problems. Furthermore, in the same work it was also shown how, conceptually, a multi-rate approach based on interpolation like that presented in this work can be reinterpreted as a variant of the more general multi-rate GARK framework. However, no explicit form of the amplification matrix is given for any of the methods discussed, nor is it explicitly shown how the interpolation procedure contributes to the amplification matrix of the resulting multi-rate method, thus making it difficult to apply those results in practice to the methods discussed in this paper. Therefore, the more general nature of the analysis in [16] does not affect the novelty and the usefulness of the analysis we will present in Section 4.

Stability analyses for specific applications to structural mechanics problems and variational integrators were proposed, among others, in [5, 8, 9, 47] and later in [10, 13]. A nonlinear stability analysis based on the TVD property was presented in [7] for explicit RK methods applied to spatially partitioned discretization of hyperbolic conservation laws. Several papers have then presented applications of multi-rate (or, in the domain specific jargon, local time-stepping) approaches to partial differential equations, see e.g. among others [14, 22, 23, 26, 43, 44, 48], but none of these works presents a detailed stability analysis of the methods employed.

4 Stability Analysis of Runge-Kutta Multi-Rate Methods with Generic Interpolation

We consider a generic linear homogeneous, constant coefficient ODE system $y' = Ly$ with $y \in \mathbb{R}^N$, $L \in \mathcal{M}_{N,N}(\mathbb{R})$ and $\sigma(L) \subset \mathbb{R}^- \cup \{0\}$. This assumption will allow to consider also systems with purely oscillatory behaviour, for which the application of multi-rate techniques is of great practical interest. We assume that the state space is partitioned *a priori* as $\mathbb{R}^N = \mathcal{V}_s \oplus \mathcal{V}_f$, where \mathcal{V}_s denotes the subspace of the slow or latent variables and \mathcal{V}_f denotes the subspace of the fast or active variables. We also set $d = \dim(\mathcal{V}_f)$. Introducing the identity matrix $\mathbb{I}_m \in \mathcal{M}_{m,m}(\mathbb{R})$ and the zero matrix $\mathbb{O}_{m,n} \in \mathcal{M}_{m,n}(\mathbb{R})$, one can represent the projections onto $\mathcal{V}_s, \mathcal{V}_f$ by the matrices

$$P_s = [\mathbb{I}_{N-d} \ \mathbb{O}_{N-d,d}] \quad P_f = [\mathbb{O}_{d,N-d} \ \mathbb{I}_d] \tag{4}$$

and the corresponding embedding operators of $\mathcal{V}_s, \mathcal{V}_f$ into \mathbb{R}^N by P_s^T, P_f^T , respectively, as often done in the literature on domain decomposition methods, see e.g. [33]. As a consequence, $P_s^T P_s$ is the operator that sets to zero all the components of a vector in \mathbb{R}^N corresponding to the fast variables; $P_f^T P_f$ acts analogously on the slow variables and $P_s^T P_s + P_f^T P_f = \mathbb{I}_N$. This entails that the model system $y' = Ly$ can be written in terms of the partitioned matrix

$$L = \begin{bmatrix} P_s L P_s^T & P_s L P_f^T \\ P_f L P_s^T & P_f L P_f^T \end{bmatrix} = \begin{bmatrix} L_{ss} & L_{sf} \\ L_{fs} & L_{ff} \end{bmatrix}, \tag{5}$$

or, equivalently, introducing the notation $y_f = P_f y, y_s = P_s y$, as

$$\begin{aligned} y'_s &= P_s L P_s^T y_s + P_s L P_f^T y_f \\ y'_f &= P_f L P_s^T y_s + P_f L P_f^T y_f. \end{aligned} \tag{6}$$

We assume that at a given time level t_n a time step $h = h_s$ is employed for the slow variables and $h_f = h_s/M$ for the $l = 0, \dots, M$ substeps of the fast variables, corresponding to the time levels $t_{n+l} = t_n + lh^f$. If the multi-rate method is based on a generic one step time discretization method whose amplification function can be written as $R(z) = N(z)/D(z)$, where N, D are polynomials in z , the value of the slow variables at the new time steps is given by $u_{n+1}^s = P_s R(h_s L) u_n$. As discussed in Section 2, values $u_{n+\tau}$ at intermediate time levels $\zeta = t_n + \tau h_s \in [t_n, t_{n+1}]$, where $\tau \in [0, 1]$ denotes the fraction of the global time step at which the interpolated value is required, can be computed from the values of u_n, u_{n+1} by appropriate interpolation techniques, represented formally by a family of operators parameterized by τ :

$$Q(\tau) : \mathbb{R}^N \rightarrow \mathbb{R}^N.$$

The simplest example of such operator is given by linear interpolation. Since in this case

$$u_{n+\tau} = (1 - \tau)u_n + \tau u_{n+1} = (1 - \tau)u_n + \tau R(h_s L)u_n,$$

interpolation can be represented in this case by the application of the operator

$$Q(\tau) = (1 - \tau)\mathbb{I}_N + \tau R(h_s L) \tag{7}$$

to u_n . The slow variables at intermediate substeps are then given by $u_{n+\tau}^s = P_s Q(\tau)u_n$. Another simple option is cubic Hermite interpolation, which is easy to implement and convenient for methods up to fourth order. In this case, also approximations of $y'(t_{n+1}), y'(t_n)$ are needed, which are provided by Lu_{n+1} and Lu_n . The interpolation operator reads in this case:

$$Q(\tau) = (1 + 2\tau)(1 - \tau)^2\mathbb{I}_N + (3 - 2\tau)\tau^2 R(h_s L) + h_s \tau(1 - \tau)^2 L + h_s(\tau - 1)\tau^2 L R(h_s L). \tag{8}$$

It should be remarked, however, that the analysis presented in [27] shows how Hermite interpolation may introduce instabilities in multi-rate procedures. More accurate interpolation procedures are provided by dense output versions of the approximation methods considered, see e.g. [18] for a general discussion and [24] for the details of some dense output DIRK and ESDIRK methods. The precise definition of the $Q(\tau)$ operators associated to dense output interpolators will be described in the following. For all these interpolation procedures, however, as well as for the continuous output interpolators discussed later (see equation (14)) it should be clear that, contrarily to what stated in [28], the one-step nature of the resulting multi-rate methods is preserved for RK methods.

The goal of the stability analysis will be to represent the discrete time evolution as $u_{n+1} = R_{mr}u_n$, where R_{mr} will depend on h_s, M, L , and on the specific properties of the time discretization and interpolation method employed. Since $u_{n+1}^s = P_s R(h_s L)u_n$ and $P_s^T P_s + P_f^T P_f = \mathbb{I}_N$, one has

$$u_{n+1}^s = P_s R(h_s L) P_s^T u_n^s + P_s R(h_s L) P_f^T u_n^f, \tag{9}$$

so that the multi-rate amplification matrix will have the form

$$R_{mr} = \begin{bmatrix} P_s R(h_s L) P_s^T & P_s R(h_s L) P_f^T \\ R_{fs} & R_{ff} \end{bmatrix}, \tag{10}$$

where the operators R_{fs} and R_{ff} depend on the specific properties of the basic single-rate method on which the multi-rate method is based and of the interpolation operator.

We will outline the stability analysis for generic ERK, DIRK and ESDIRK methods, identified as customary by their Butcher tableaux. These consist of $A \in \mathcal{M}_{s,s}(\mathbb{R})$ and $b, c \in \mathbb{R}^s$, where s is the number of stages and the usual simplifying hypotheses are implicitly assumed, see e.g. [18]. The single-rate method applied to the problem $y' = f(y, t)$ can therefore be written, following [25], as

$$\begin{aligned}
 U^{(i)} &= u_n + h \sum_{j=1}^s a_{i,j} f(U^{(j)}, t_n + c_j h) \quad i = 1, \dots, s \\
 u_{n+1} &= u_n + h \sum_{i=1}^s b_i f(U^{(i)}, t_n + c_i h).
 \end{aligned}
 \tag{11}$$

In the linear case $y' = Ly$ this yields

$$\begin{aligned}
 U^{(i)} &= u_n + h \sum_{j=1}^s a_{i,j} L U^{(j)} \quad i = 1, \dots, s \\
 u_{n+1} &= u_n + h \sum_{i=1}^s b_i L U^{(i)}.
 \end{aligned}
 \tag{12}$$

A dense output approximation of y at time level $\zeta = t_n + \tau h_s$, see again [18, 24], can be written as

$$u_{n+\tau} = u_n + h \sum_{i=1}^s b_i^*(\tau) L U^{(i)}, \quad i = 1, \dots, s$$

with $b_i^*(\tau) = \sum_{j=1}^{p^*} b_{i,j}^* \tau^j$ for appropriate method specific values of p^* and $b_{i,j}^*$. For ERK methods

$$R^{(1)} = \mathbb{I} \quad R^{(k)} = \mathbb{I} + h \sum_{j=1}^{k-1} a_{k,j} L R^{(j)},
 \tag{13}$$

so that $U^{(i)} = R^{(i)} u_n$. As a consequence, the operator representation of the dense output interpolator can be written as

$$Q(\tau) = \mathbb{I} + h \sum_{i=1}^s b_i^*(\tau) L R^{(i)}.
 \tag{14}$$

For DIRK methods one defines instead

$$\begin{aligned}
 R^{(1)} &= (\mathbb{I} - h a_{1,1} L)^{-1} \\
 R^{(k)} &= (\mathbb{I} - h a_{k,k} L)^{-1} \left[\mathbb{I} + h \sum_{j=1}^{k-1} a_{k,j} L R^{(j)} \right], \quad k \geq 2,
 \end{aligned}
 \tag{15}$$

so that again the dense output interpolator can be represented by formula (14), where this definition of the $R^{(k)}$ operator is employed. By the same reasoning that allows to derive (14), one also has that

$$R(hL) = \mathbb{I} + h \sum_{i=1}^s b_i L R^{(i)}. \tag{16}$$

Notice that formulae (15) and (16) are actually valid also for the ERK case, by simply assuming $a_{i,i} = 0, i = 1, \dots, s$, and in the case of ESDIRK methods, by assuming $a_{1,1} = 0$. This will allow to simplify the following stability analysis, which will be carried out explicitly for DIRK methods only, but whose results are also valid for ERK and ESDIRK methods again with appropriate assumptions on the Butcher tableaux.

For the corresponding multi-rate methods, denoting again by R the amplification function of the single-rate method, for the linear problem $y' = Ly$ the slow variables can be computed as discussed previously by $u_{n+1}^s = P_s R(h_s L) P_s^T u_n^s + P_s R(h_s L) P_f^T u_n^f$. For the fast variables, assuming that

$$U_s^{(i,l)} \approx P_s Q((l + c_i)/M) u_n$$

for $i = 1, \dots, s$, and $l = 0, \dots, M - 1$, and setting $Q^{(i,l)} = Q((l + c_i)/M)$ for brevity, one has

$$\begin{aligned} U_f^{(i,l)} &= u_{n,l}^f + h_f \sum_{j=1}^s a_{i,j} L_{ff} U_f^{(j,l)} + h_f \sum_{j=1}^s a_{i,j} L_{fs} P_s Q^{(j,l)} u_n \\ u_{n,l+1}^f &= u_{n,l}^f + h_f \sum_{i=1}^s b_i L_{ff} U_f^{(i,l)} + h_f \sum_{i=1}^s b_i L_{fs} P_s Q^{(i,l)} u_n. \end{aligned} \tag{17}$$

In order to derive an explicit expression for the amplification matrix of the multi-rate method, we will now consider the specific case of DIRK methods, which as explained before allows to derive formulae that are also valid for ERK and ESDIRK methods. From the basic definitions it follows that

$$\begin{aligned} U_f^{(1,l)} &= u_{n,l}^f + h_f a_{1,1} L_{ff} U_f^{(1,l)} + h_f a_{1,1} L_{fs} P_s Q^{(1,l)} u_n \\ U_f^{(2,l)} &= u_{n,l}^f + h_f a_{2,1} L_{ff} U_f^{(1,l)} + h_f a_{2,1} L_{fs} P_s Q^{(1,l)} u_n \\ &\quad + h_f a_{2,2} L_{ff} U_f^{(2,l)} + h_f a_{2,2} L_{fs} P_s Q^{(2,l)} u_n \\ &\quad \vdots \\ U_f^{(s,l)} &= u_{n,l}^f + h_f \sum_{j=1}^s a_{s,j} L_{ff} U_f^{(j,l)} + h_f \sum_{j=1}^s a_{s,j} L_{fs} P_s Q^{(j,l)} u_n \\ u_{n,l+1}^f &= u_{n,l}^f + h_f \sum_{i=1}^s b_i L_{ff} U_f^{(i,l)} + h_f \sum_{i=1}^s b_i L_{fs} P_s Q^{(i,l)} u_n. \end{aligned} \tag{18}$$

This can be rewritten as

$$\begin{aligned}
 U_f^{(1,l)} &= (\mathbb{I}_d - h_f a_{1,1} L_{ff})^{-1} \left(u_{n,l}^f + h_f a_{1,1} L_{fs} P_s Q^{(1,l)} u_n \right) \\
 U_f^{(2,l)} &= (\mathbb{I}_d - h_f a_{2,2} L_{ff})^{-1} \times \left[u_{n,l}^f + h_f a_{2,1} \left(L_{ff} U_f^{(1,l)} + L_{fs} P_s Q^{(1,l)} u_n \right) \right. \\
 &\quad \left. + h_f a_{2,2} L_{fs} P_s Q^{(2,l)} u_n \right] \\
 &\vdots \\
 U_f^{(s,l)} &= (\mathbb{I}_d - h_f a_{s,s} L_{ff})^{-1} \times \left[u_{n,l}^f + h_f \sum_{j=1}^{s-1} a_{s,j} L_{ff} U_f^{(j,l)} \right. \\
 &\quad \left. + h_f \sum_{j=1}^s a_{s,j} L_{fs} P_s Q^{(j,l)} u_n \right] \\
 u_{n,l+1}^f &= u_{n,l}^f + h_f \sum_{i=1}^s b_i L_{ff} U_f^{(i,l)} + h_f \sum_{i=1}^s b_i L_{fs} P_s Q^{(i,l)} u_n. \tag{19}
 \end{aligned}$$

We then define

$$R_{ff}^{(k)} = (\mathbb{I}_d - h_f a_{k,k} L_{ff})^{-1} \left[\mathbb{I}_d + h_f \sum_{j=1}^{k-1} a_{k,j} L_{ff} R_{ff}^{(j)} \right] \quad k = 1, \dots, s \tag{20}$$

$$B^{(1,l)} = (\mathbb{I}_d - h_f a_{1,1} L_{ff})^{-1} h_f a_{1,1} L_{fs} P_s Q^{(1,l)} \text{ and for } k = 2, \dots, s$$

$$B^{(k,l)} = (\mathbb{I}_d - h_f a_{k,k} L_{ff})^{-1} \left[h_f \sum_{j=1}^k a_{k,j} L_{fs} P_s Q^{(j,l)} + h_f \sum_{j=1}^{k-1} a_{k,j} L_{ff} B^{(j,l)} \right].$$

Notice that in the ERK and ESDIRK cases this yields $B^{1,l} = \mathbb{O}_{d,N}$. Equations (19) can then be rewritten as

$$\begin{aligned}
 U_f^{(k,l)} &= R_{ff}^{(k)} u_{n,l}^f + B^{(k,l)} u_n \quad k = 1, \dots, s \\
 u_{n,l+1}^f &= \left[\mathbb{I}_d + h_f \sum_{i=1}^s b_i L_{ff} R_{ff}^{(i)} \right] u_{n,l}^f \\
 &\quad + h_f \sum_{i=1}^s b_i L_{ff} B^{(i,l)} u_n + h_f \sum_{i=1}^s b_i L_{fs} P_s Q^{(i,l)} u_n \tag{21}
 \end{aligned}$$

Setting then for $l = 0, \dots, M - 1$

$$\begin{aligned}
 C_{ff} &= \mathbb{I}_d + h_f \sum_{i=1}^s b_i L_{ff} R_{ff}^{(i)} \\
 D^{(l)} &= \sum_{i=1}^s b_i \left[L_{ff} B^{(i,l)} + L_{fs} P_s Q^{(i,l)} \right] \tag{22}
 \end{aligned}$$

and using the identity $P_s^T P_s + P_f^T P_f = \mathbb{I}_N$, one has for the generic step

$$\begin{aligned}
 u_{n+1}^f &= C_{ff}^M u_n^f + h_f \sum_{k=1}^M C_{ff}^{M-k} D^{(k-1)} P_f^T u_n^f \\
 &\quad + h_f \sum_{k=1}^M C_{ff}^{M-k} D^{(k-1)} P_s^T u_n^s.
 \end{aligned}
 \tag{23}$$

As a consequence, the blocks R_{ff} and R_{fs} of the multirate amplification matrix R_{mr} in (10) have the form:

$$\begin{aligned}
 R_{ff} &= C_{ff}^M + h_f \sum_{k=1}^M C_{ff}^{M-k} D^{(k-1)} P_f^T \\
 R_{fs} &= h_f \sum_{k=1}^M C_{ff}^{M-k} D^{(k-1)} P_s^T.
 \end{aligned}
 \tag{24}$$

Notice that, from (16), one has $C_{ff} = R(h_f L_{ff})$.

5 Examples of Stability Analysis on Specific Model Problems

The amplification matrices derived in Section 4 can be in principle computed analytically for a generic (small) linear system. Unfortunately, the resulting expressions are extremely cumbersome and not very illuminating, even in the simplest case of problems with 2 DOF. Furthermore, using these analytic expressions, it is quite difficult to study in detail the case of large number of fast substeps, which is one of the main results of our analysis. For these reasons, we will compute the spectral radii of the stability matrices numerically.

In previous works devoted to the analysis of multi-rate methods [21, 27, 28, 41], a simple two DOF system has been considered as model problem for multi-rate stability analysis. This system can be written as

$$L = \begin{bmatrix} -1 & 1 \\ -\kappa\alpha & -\alpha \end{bmatrix}.$$

The parameter $\alpha > 0$ determines the ratio of the fast to slow time scale and is therefore an indicator of the stiffness of the system. The parameter κ denotes the strength of the coupling of the fast to the slow component. For $\kappa < 1$, both eigenvalues of L are negative, so this is a suitable model problem for the stability analysis. Since it is not possible to present the results of stability analyses for a whole class of methods, we will only discuss some examples of an explicit and an implicit RK method, noting that similar results have been obtained when performing the same analysis on other methods. The main goal of the analysis is to assess the impact of the multi-rate procedure on the stability properties of the original single-rate method. For this purpose, we will introduce as in [6] the parameter $C = h_s \Lambda$, where $\Lambda = \max(|\lambda_i|)$ and $\lambda_i, i = 1, 2$ denote the eigenvalues of L . This parameter, which in a PDE context would be interpreted as an analog of the Courant number, must satisfy an $O(1)$ A-stability bound for single-rate explicit methods, while essentially all useful single-rate implicit methods are A-stable for arbitrarily large values of C . As a consequence, increased values of C for explicit multi-rate methods will demonstrate their potential increase in efficiency with respect to their single-rate counterparts, while conditional stability for implicit multi-rate methods will demonstrate the robustness limits of the multi-rate approach. As example of

Table 1 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\alpha = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 0.9 \times 10^{-5}$	3	3	3	3	3	3	3
$\kappa = 0.9 \times 10^{-4}$	3	3	3	3	3	3	3
$\kappa = 0.9 \times 10^{-3}$	3	3	3	3	3	3	3
$\kappa = 0.9 \times 10^{-2}$	3	3	3	3	3	3	3
$\kappa = 0.9 \times 10^{-1}$	4	4	4	4	4	4	4
$\kappa = 0.9$	3	3	3	3	3	3	3

Table 2 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\alpha = 10$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 0.9 \times 10^{-5}$	6	12	23	28	28	28	28
$\kappa = 0.9 \times 10^{-4}$	6	12	23	28	28	28	28
$\kappa = 0.9 \times 10^{-3}$	6	12	23	27	26	26	26
$\kappa = 0.9 \times 10^{-2}$	6	12	16	15	15	15	15
$\kappa = 0.9 \times 10^{-1}$	6	11	10	10	10	10	10
$\kappa = 0.9$	5	10	7	7	7	7	7

Table 3 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\alpha = 100$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 0.9 \times 10^{-5}$	6	12	23	45	90	≥ 100	≥ 100
$\kappa = 0.9 \times 10^{-4}$	6	12	23	45	76	74	74
$\kappa = 0.9 \times 10^{-3}$	6	12	23	45	43	43	43
$\kappa = 0.9 \times 10^{-2}$	6	12	23	25	25	25	25
$\kappa = 0.9 \times 10^{-1}$	6	12	16	15	15	15	15
$\kappa = 0.9$	6	10	10	10	10	10	10

explicit RK method, we consider the classical fourth order RK method, for which Hermite interpolation is used in the multi-rate procedure. The Butcher tableaux of the method is reported in Appendix A. It can be observed from Tables 1, 2, 3 and 4 that for the explicit method considered the multi-rate approach effectively extends the stability region as the time scale separation increases. Notice that the maximum value of C for which the single rate method is stable is 3.

As an example of implicit RK methods, we consider the fourth order ESDIRK method denoted as ESDIRK4(3)6L[2]SA in [24], Section 7.1.1, for which the associated continuous output interpolator was used in the multi-rate procedure. Again, the Butcher tableaux of the method and the coefficients required to build the associated continuous output interpolation are reported in Appendix A. Repeating the previous analysis for this method, the stability properties are not affected by the multi-rate extension, as long as the coupling coefficient is

Table 4 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\alpha = 1000$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 0.9 \times 10^{-5}$	6	12	23	45	90	≥ 100	≥ 100
$\kappa = 0.9 \times 10^{-4}$	6	12	23	45	90	≥ 100	≥ 100
$\kappa = 0.9 \times 10^{-3}$	6	12	23	45	76	74	74
$\kappa = 0.9 \times 10^{-2}$	6	12	23	45	43	43	43
$\kappa = 0.9 \times 10^{-1}$	6	12	23	25	25	25	25
$\kappa = 0.9$	6	12	16	15	15	15	15

bounded by 1. The same seems to hold more generally for all the unconditionally L-stable single rate methods we have studied. No results are reported in this case since the parameter C was found to be larger than the reference value 100 in all cases.

System (25), however, does not have in our opinion a clear physical interpretation and does not allow to study the stability for problems with eigenvalues located close or on the imaginary axis, which are also very important in many applications.

We therefore consider a system with four DOF that has a clear physical interpretation and allows to study explicitly the dependence of the multi-rate stability on the intensity of the coupling of fast and slow variables and on the separation of their time scales. This system is an extension of a similar system considered in [6], in which however only a partial coupling of the slow and fast variables was considered. More specifically, consider the second order system

$$\begin{aligned}
 m_1 u_1'' &= -k_1 u_1 - k_2(u_1 - u_2) - c_1 u_1' \\
 m_2 u_2'' &= k_2(u_1 - u_2) - c_2 u_2'.
 \end{aligned}
 \tag{25}$$

It is easy to see that the system describes the dynamics of two point masses m_1, m_2 , the first of which is subject to elastic forces due to two springs of elastic constants k_1, k_2 , respectively, the first of which is attached to a wall, while the second ties the two masses. Both masses are also subject to frictional forces defined by the coefficients c_1, c_2 . Assuming that $y_1 = u_1, y_2 = u_1', y_3 = u_2, y_4 = u_2'$, one can rewrite the system in first order form as

$$\begin{aligned}
 y_1' &= y_2 \\
 y_2' &= -\frac{k_1}{m_1} y_1 - \frac{k_2}{m_1} (y_1 - y_3) - \frac{c_1}{m_1} y_2 \\
 y_3' &= y_4 \\
 y_4' &= \frac{k_2}{m_2} (y_1 - y_3) - \frac{c_2}{m_2} y_4.
 \end{aligned}
 \tag{26}$$

The system can be further rewritten to highlight the intensity of the coupling and the time scale separation between the slow and fast variables. For $i = 1, 2$ we define the natural periods of the two masses as $\omega_i = \sqrt{k_i/m_i}$ and the quantities $\gamma_i = c_i/m_i$ and we set

$$\alpha = \frac{\omega_2}{\omega_1} \quad \beta = \frac{\gamma_2}{\gamma_1} \quad \kappa = \frac{m_2}{m_1}.
 \tag{27}$$

Notice that α, β determine the ratios of the proper periods of the two (decoupled) masses and of the effective frictional forces, so that in order for y_1, y_2 to represent slow variables either

α , or β , or both must be significantly larger than one. On the other hand, κ represents the intensity of the coupling, so that for $\kappa \rightarrow 0$ the slow dynamics tends to be decoupled from the fast dynamics, while stronger coupling arises if κ is of order 1. With these definitions, the system can be rewritten as

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -\omega_1^2(1 + \alpha^2\kappa)y_1 - \gamma_1 y_2 + \kappa\alpha^2\omega_1^2 y_3 \\ y_3' &= y_4 \\ y_4' &= \alpha^2\omega_1^2 y_1 - \alpha^2\omega_1^2 y_3 - \beta\gamma_1 y_4. \end{aligned} \tag{28}$$

In matrix form, the system can be written as $y' = Ly$ with system matrix given by

$$L = \begin{bmatrix} L_{ss} & L_{sf} \\ L_{fs} & L_{ff} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2(1 + \alpha^2\kappa) & -\gamma_1 & \kappa\alpha^2\omega_1^2 & 0 \\ 0 & 0 & 0 & 1 \\ \alpha^2\omega_1^2 & 0 & -\alpha^2\omega_1^2 & -\beta\gamma_1 \end{bmatrix}.$$

As for the previous model problem, we will perform the stability analysis fixing the time scale ratios and considering different values of the coupling parameter κ and of the number of sub-steps M . More specifically, we consider time scale ratios α, β of order 10 and 100 and for each value of α, β , we consider a range of values of $\kappa \in [0, 1]$ and a range of values of the number of sub-steps for the fast variables. It should be remarked that the range for which the application of multi-rate methods is meaningful is $\kappa < 1$. Indeed, for values $\kappa \approx 1$ both degrees of freedom exhibit fast dynamics, so that no actual time scale separation exists and the application of multi-rate methods would not be advantageous. Furthermore, it is of interest to assess how the time resolution used for the fast variables affects the overall stability of the multi-rate method.

We consider the same examples of explicit and implicit RK methods as for the 2 DOF case. We first consider the purely oscillatory case $\gamma_1 = 0$. In this setting, the multi-rate extension does not seem able to increase substantially the stability region of the single rate scheme. Indeed, in the case of implicit methods, the multi-rate version becomes only conditionally stable with values of the parameter C close to 1.

If instead the real part of the system eigenvalues is non zero, the multi-rate extension of the RK method has a significantly wider stability range. As in the previous case, the results are reported in Tables 5, 6, 7 and 8 for $\gamma = 0.01, \beta = 1$ and α ranging from 1 to 1000. It can be observed that stability is achieved for much larger values of the parameter C than those of the single-rate method, which is approximately 3. As expected, the stability is improved in the weak coupling limit and if a larger number of substeps is employed. However, it can be observed that the maximum value of C for which stability is maintained does not grow monotonically as a function of the number of substeps.

Similar results, not shown, are also obtained for the smaller value $\gamma_1 = 0.001$ and for the cases $\alpha = 100, \beta = 10$ and $\alpha = 100, \beta = 100$. Instead, when the time scale separation is only due to the real part of the eigenvalues, such as for example in the cases $\gamma_1 = 0.01, \alpha = 1$ and $\beta = 10$ or $\beta = 100$, the multi-rate extension of the explicit RK method has the same range of stable C values as its single-rate counterpart.

We then consider again the ESDIRK4(3)6L[2]SA method, with the associated continuous output interpolator. As for explicit methods, if the real part of the system eigenvalues is non zero, the multi-rate extension of the ESIDRK method maintains unconditional stability for a wider range of values for the coupling parameter κ , as it can be seen from Tables 9, 10, 11 and 12 for the case $\gamma_1 = 0.01, \beta = 1$ and α ranging from 1 to 1000. On the other hand,

Table 5 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\gamma_1 = 0.01$, $\alpha = 1$, $\beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	3	3	3	3	3	3	3
$\kappa = 10^{-4}$	3	3	3	3	3	3	3
$\kappa = 10^{-3}$	3	3	3	3	3	3	3
$\kappa = 10^{-2}$	3	3	3	3	3	3	3
$\kappa = 10^{-1}$	4	4	4	4	4	4	4
$\kappa = 1$	3	3	3	3	3	3	3

Table 6 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\gamma_1 = 0.01$, $\alpha = 10$, $\beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	6	12	23	29	29	26	26
$\kappa = 10^{-4}$	6	12	23	14	13	13	13
$\kappa = 10^{-3}$	6	10	7	6	6	6	6
$\kappa = 10^{-2}$	5	6	6	6	6	6	6
$\kappa = 10^{-1}$	4	5	5	5	5	5	5
$\kappa = 1$	4	4	4	4	4	4	4

Table 7 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\gamma_1 = 0.01$, $\alpha = 100$, $\beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	6	12	23	41	62	13	13
$\kappa = 10^{-4}$	6	12	22	7	6	6	6
$\kappa = 10^{-3}$	6	12	7	6	6	6	6
$\kappa = 10^{-2}$	6	6	6	6	6	6	6
$\kappa = 10^{-1}$	5	5	5	5	5	5	5
$\kappa = 1$	4	4	4	4	4	4	4

Table 8 Maximum value of C for stability of the classical fourth order ERK method with Hermite interpolator, $\gamma_1 = 0.01$, $\alpha = 1000$, $\beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	6	12	23	46	6	6	6
$\kappa = 10^{-4}$	6	12	23	6	6	6	6
$\kappa = 10^{-3}$	6	12	7	6	6	6	6
$\kappa = 10^{-2}$	6	6	6	6	6	6	6
$\kappa = 10^{-1}$	5	5	5	5	5	5	5
$\kappa = 1$	4	4	4	4	4	4	4

Table 9 Maximum value of C for stability of fourth order ESDIRK method with continuous output interpolator, $\gamma_1 = 0.01, \alpha = 1, \beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-4}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-3}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-2}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-1}$	≥ 100	7	7	7	7	7	7
$\kappa = 1$	4	4	4	4	4	4	4

Table 10 Maximum value of C for stability of fourth order ESDIRK method with continuous output interpolator, $\gamma_1 = 0.01, \alpha = 10, \beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-4}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-3}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-2}$	≥ 100	5.0	5.0	5.0	5.0	5.0	5.0
$\kappa = 10^{-1}$	3.0	3.0	3.0	3.0	3.0	3.0	3.0
$\kappa = 1$	2.0	2.0	2.0	2.0	2.0	2.0	2.0

Table 11 Maximum value of C for stability of fourth order ESDIRK method with continuous output interpolator, $\gamma_1 = 0.01, \alpha = 100, \beta = 1$

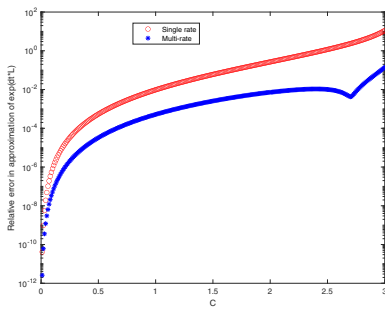
	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-4}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-3}$	≥ 100	≥ 100	5.0	5.0	5.0	5.0	5.0
$\kappa = 10^{-2}$	≥ 100	3.0	3.0	3.0	3.0	3.0	3.0
$\kappa = 10^{-1}$	2.0	2.0	2.0	2.0	2.0	2.0	2.0
$\kappa = 1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0

values of $C = O(1)$ are obtained for $\kappa \geq 10^{-3}$ in the $\alpha = 10, 100$ cases and for $\kappa \geq 10^{-5}$ in the $\alpha = 1000$ case. It is also to be remarked that, when unconditional stability is lost, the number of sub-steps does not have a clear impact on the maximum value of C for which stability is guaranteed. Similar results, not shown, are obtained with $\gamma_1 = 0.001$, If instead $\gamma_1 = 0.01, \alpha = 1, \beta = 10$ or $\gamma_1 = 0.01, \alpha = 1, \beta = 100$, so that time scale separation only depends on the real part of the eigenvalues, unconditional stability is maintained for all values of $\kappa < 10^{-1}$.

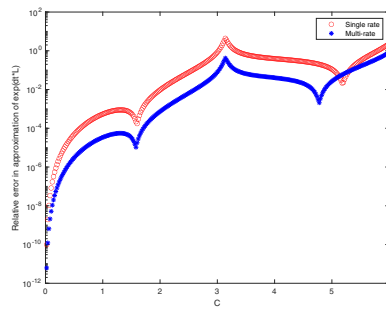
The previously introduced expression for the multi-rate amplification matrix also allows to assess the accuracy gain achieved by the use of smaller time steps on the fast components. For example, we consider the case of a system with $\gamma = 0.01, \alpha = 50, \beta = 1$ and weak coupling $\kappa = 10^{-3}$. We compute the exact evolution matrix $\exp(Lt)$ of problem $y' = Ly$ and compare

Table 12 Maximum value of C for stability of fourth order ESDIRK method with continuous output interpolator, $\gamma_1 = 0.01$, $\alpha = 1000$, $\beta = 1$

	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$	$M = 64$	$M = 128$
$\kappa = 10^{-5}$	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100
$\kappa = 10^{-4}$	≥ 100	≥ 100	5	5	5	5	5
$\kappa = 10^{-3}$	3	3	3	3	3	3	3
$\kappa = 10^{-2}$	2	2	2	2	2	2	2
$\kappa = 10^{-1}$	1	1	1	1	1	1	1
$\kappa = 1$	1	1	1	1	1	1	1



a)



b)

Fig. 1 Errors with respect of exact solution at increasing values of C for a) explicit fourth order RK method, b) ESDIRK fourth order method

it to appropriate powers of the single-rate and multi-rate amplification matrix for different values of the time step. For the multi-rate methods, $M = 10$ substeps were employed. The relative errors in the l^2 operator norm of the approximations of $\exp(Lt)$ obtained by the explicit fourth order RK method and fourth order ESDIRK method considered before are reported in Figure 1, for values of the time step corresponding to increasing values of the parameter C . It can be observed that significant improvements in the accuracy of the approximation obtained with a given time step can be achieved.

6 Implementation in the OpenModelica Environment

The multi-rate Runge-Kutta methods presented in the previous Sections were implemented in the GBODE solver, which is part of the open-source OpenModelica simulation environment described in [11].

OpenModelica allows to model and simulate dynamical systems described by differential-algebraic equations using the high-level, equation-based, object-oriented modelling language Modelica [29]. OpenModelica applies structural analysis, symbolic manipulation and numerical solvers to reduce the (possibly high-index) differential-algebraic formulation of the system model to a set of explicit ordinary differential equations, producing efficient C code to compute the right-hand-side of Equation (1) and its Jacobian $\partial f / \partial y$. This code is then linked to ODE solvers such as GBODE or state-of-the-art solvers such as IDA or CVODE from SUNDIALS [19]. In order to provide an efficient implementation of the multi-rate solver, the

generated C code was augmented with a selection algorithm that, based on structural dependencies, only runs the parts of the code that are strictly necessary to compute the derivatives of the fast states $f_f(y_s, y_f, t)$ of Equation (1), as well as the Jacobian $\partial f_f / \partial y_f$, for the computation of local refinement steps.

The implementation of the multi-rate method in GBODE is quite efficient from a computational point of view: it is written using the C language, it uses the KINSOL solver of SUNDIALS to solve the nonlinear equations of implicit methods such as ESDIRK, relying on sparse linear algebra methods. It also supports dense output to provide the solution on a regular time grid, as well as to compute accurate high-order interpolation of slow variables during refinement steps.

7 Numerical Results

In this Section we will present three numerical test cases, reporting results with the multi-rate versions of the methods fully described in Appendix A. The numerical results demonstrate how the proposed multi-rate method works when simulating non-trivial systems. Although OpenModelica is capable of handling extremely complex dynamical system models, for this paper we have selected test cases whose model equations can be written explicitly in compact form, so that the results are in principle reproducible with other implementations of the multi-rate method or different numerical approaches. All tests were carried out on a laptop with an i7-1365U CPU running Windows 11.

7.1 Inverter Chain

The first test case consists of the model of an inverter chain, which is an important test problem for electrical circuits that has already been considered in the literature on multi-rate methods, e.g., in [42, 49]. The system of equations is given by

$$y_1'(t) = U_{op} - y_1(t) - \Gamma g(u(t), y_1(t)) \quad (29)$$

$$y_j'(t) = U_{op} - y_j(t) - \Gamma g(y_{j-1}(t), y_j(t)), \quad j = 2, 3, \dots, N, \quad (30)$$

where y_j is the output voltage of the j -th inverter, $u(t)$ is the input voltage of the first inverter, U_{op} is the operating voltage corresponding to the logical value 1, and Γ is a stiffness parameter. The function g is defined as

$$g(y, z) = \max(y - U_\tau, 0)^2 - \max(y - z - U_\tau, 0)^2, \quad (31)$$

where U_τ is a switching threshold. The system has a stable equilibrium if the input is zero, the odd-numbered inverters have output zero and the even-numbered inverters have output one. If the input is increased up to U_{op} , the first inverter output switches from zero to one, triggering a switching cascade that propagates through the entire chain at finite speed. If the first inverter input is then switched back to zero, a second switching cascade propagates again through the system, bringing it back to the original equilibrium. This problem is obviously a good candidate for adaptive multi-rate integration, since only a small fraction of inverter outputs are changing at any given time. It is also representative of the simulation of real-life electronic digital circuit models, where only a small fraction of gates are active at any given point in time, while most other gates sit idle in a stable state.

The model was set up with $N = 1000$ inverters, $U_{op} = 5$, $U_\tau = 1$, and $\Gamma = 500$. The integration interval is $t \in [0, 200]$, with initial conditions $y_j(0) = 6.247 \times 10^{-3}$ for even j and

Fig. 2 Reference solution for selected even inverter outputs

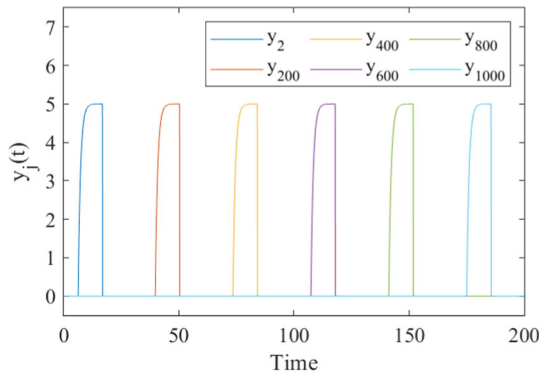
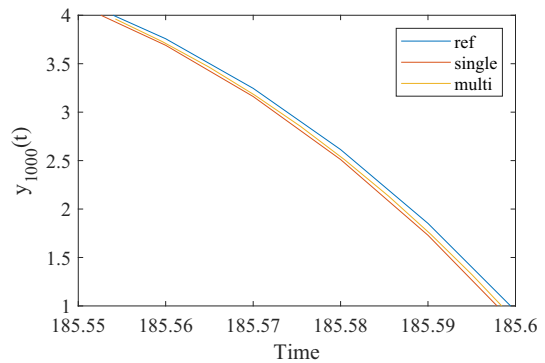


Fig. 3 Comparison of reference, single-rate, and multi-rate ESDIRK3(2)4L[2]SA solutions in the inverter test case



$y_j(0) = 1$ for odd j ; $u(t)$ is a continuous piecewise-linear function which is zero until $t = 5$, then increases to 5 until $t = 10$, remains constant until $t = 15$, decreases to zero until $t = 20$ and then remains zero for $t > 20$. We have used the third order ESDIRK method denoted as ESDIRK3(2)4L[2]SA in [24] to compute a reference single-rate solution with relative and absolute tolerance 10^{-9} , a single-rate solution with relative and absolute tolerance 10^{-5} and multi-rate solution with relative and absolute tolerance 10^{-5} . For the multi-rate algorithms, the parameter values $\alpha_{max} = 1.2$, $\alpha_{min} = 0.5$, $\alpha = 0.9$, $\phi = 0.05$, and $\beta = 1$ were used; the associated dense output reconstruction was used for the interpolation operator $Q(\tau)$, both in the multi-rate procedure to interpolate the slow variables when integrating the fast variables, and when re-sampling the solution over a regular time grid with 0.01 s intervals.

Figure 2 shows the output of the even inverters y_{200} , y_{400} , y_{600} , y_{800} , y_{1000} in the reference solution, which follow the rising and falling edges of the first even inverter output y_2 with increasing delay, corresponding to about 150 time units for the last inverter output y_{1000} . Figure 3 shows a detail of the falling edge of the last inverter output y_{1000} , comparing the reference, single-rate, and multi-rate solutions. It is apparent how the multi-rate and single-rate solutions are very close to each other and that both are shifted in time with respect to the reference one by about 0.0015 time units, corresponding to a relative error of about 10^{-5} , which is compatible with the tolerances employed.

Figure 4 shows the activity diagram of the multi-rate algorithm between $t = 25$ and $t = 35$. Here, the rows correspond to the variables in vector y , while the columns correspond to time steps; blue points correspond to variables that are active in a given time step. The full blue vertical lines correspond to global time steps, where all variables are involved in

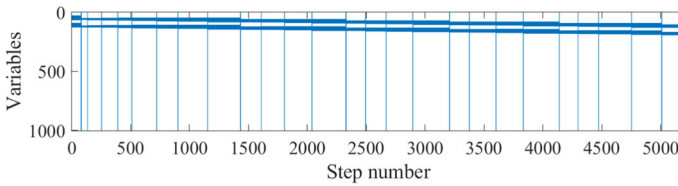


Fig. 4 Activity diagram of the ESDIRK3(2)4L[2]SA multi-rate solution in the inverter test case

the computation of the time step, while the slightly slanted pairs of horizontal blue segments correspond to the rising and falling inverter output waves, slowly moving along the inverter chain. Only the inverters currently undergoing the transition are automatically selected as active variables during refinement steps. The activity plot clearly highlights the ability of the self-adjusting multi-rate approach to identify automatically the set of fast evolving variables, which is essential if this set is time-varying.

In this specific example, the activity of fast states is quite localized: the typical size of the fast variable subspace is about 5, i.e., $1/200$ of the global system size N . This means that, as long as ϕ is significantly less than one (otherwise most steps would be chosen as global ones), and larger than $1/200$, the same number of fast states will be selected. This was experimentally confirmed by running the simulations with different values of ϕ in the range $0.4 < \phi < 0.01$, which led to very similar results in terms of number of global and refinement steps. In general, the performance of the proposed multi-rate algorithm will not be too sensitive to the specific value of ϕ for those cases where the multi-rate strategy is more advantageous, i.e., when there is strongly localized activity most of the time, so that only a tiny fraction of variables shows significant integration errors for most of the time steps.

Experimenting with this test case showed an important fact, that can have a substantial impact on the solver performance. The adoption of a multi-rate method on a strongly nonlinear model such as the one considered in this example has consequences on the convergence of Newton's method when solving the implicit equation for global steps. When single-rate integration is performed, the length of the steps is severely limited by the accuracy requirement; as a consequence, the extrapolation of the previous solution provides a good initial guess for Newton's method, that converges in a limited number of iterations in most cases, with a few convergence failures. When multi-rate integration is performed, global steps are much longer (100 times more on average, in this case), so the convergence of Newton's method in such steps can be more problematic.

The comparison of some performance indicators of the single and multi-rate algorithms is shown in Table 13. The first column refers to the single-rate method, while the second and the third refer to two variants of the multi-rate method. The Jac A variant uses the same Jacobian recomputation strategy of the single-rate method, i.e., the Jacobian is only recomputed by KINSOL according to its heuristic criteria, that try to strike a balance between the added computational effort of re-computing and re-factorizing the Jacobian and the added computational effort of additional iterations due to the use of outdated Jacobians. As a result, the Jacobian is pre-emptively re-computed by GBODE every 10 integration steps. In the Jac B variant, instead, GBODE pre-emptively re-computes the Jacobian at the beginning of each new time step.

Even though the total number of steps of the single-rate and multi-rate method is comparable, the multi-rate method requires less than 1% global steps than the single-rate method, while 99% of the steps are local refinement steps, involving only a small number of variables; thus, there is a potential for a speed-up factor up to 100.

Table 13 Performance of single- and multi-rate ESDIRK3(2)4L2JSA methods and of IDA solver in the inverter test case

	Single-Rate	Multirate Jac A	Multirate Jac B	IDA
Accepted global steps	65316	495	510	55064
Rejected global steps (error test failure)	12081		0	8468
Rejected global steps (convergence failure)	1	103	107	18
Accepted fast steps	//	66915	66954	//
Rejected fast steps (error test failure)	//	11654	11641	//
Global RHS calls	1221140	593591	261995	104003
Global Jacobian computations	6943	24339	18266	15741
Local RHS calls	//	1143414	566482	//
Local Jacobian computations	//	12889	235762	//
Simulation time in s	66.1	37.8	14.6	12.5

Unfortunately, the considered test case is quite stiff and strongly non-linear, so the aforementioned issue with the convergence of the implicit stage computations is particularly severe. Whereas the average number of calls to the right hand side of Eq. (1) per step is about 20 for the single-rate method, which is reasonable for a third-order implicit single-step method, it turns out to be over 1000 for the global steps of the multi-rate method, due to a massively increased number of iterations per step to achieve successful convergence. As a consequence, the speed-up factor turns out to be less than 2. In this case, adopting strategy Jac B is very beneficial, as it leads to a speed-up factor of about 4, due to the more than halved number of global and local right-hand-side computations that stems from the reduced number of iterations, as well as to a significant reduction of Jacobian computations for the global steps. The increased number of Jacobian computations for the local steps is probably not too relevant, due to their very small size.

The last column of Table 13 allows to compare the performance of the state-of-the-art IDA solver to single-rate and multi-rate ESDIRK3 as implemented in GBODE. IDA turns out to be significantly faster than single-rate ESDIRK and slightly faster than the multi-rate ESDIRK for two main reasons. The first is that IDA is a multi-step method that requires to solve only one implicit system of equations per step, whereas ESDIRK3 is a three-stage Runge-Kutta method that requires to solve an implicit system three times per step. The second is that IDA is a highly optimized state-of-the-art solver, whereas GBODE is not yet fully optimized from the point of view of Jacobian recomputation strategies. Such an optimization, though, goes beyond the scope of the present paper.

7.2 Finite difference discretization of the Burgers equation

The second test case consists of an application to a PDE problem. We consider the viscous Burgers equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (32)$$

as a simple model of computational fluid dynamics applications. The equation is considered on the domain $[0, 25]$ and time interval $[0, 5]$. We assume $\nu = 10^{-2}$ and an initial datum given by

$$u_0(x) = \exp \left\{ - \left(\frac{x - L/2}{L/50} \right)^2 \right\}. \quad (33)$$

Equation (32) is semi-discretized in space by a standard centered finite difference approximation on a uniform mesh on $[0, 25]$ with $N = 1000$ nodes. We used the third order ESDIRK method denoted as ESDIRK3(2)4L[2]SA in [24] to compute a reference single-rate solution with relative and absolute tolerance 10^{-9} , a single-rate solution with relative and absolute tolerance 10^{-5} , two multi-rate solutions with relative and absolute tolerance 10^{-5} , one with $\phi = 0.2$ and one with $\phi = 0.04$, and another two multi-rate solutions with relative and absolute tolerance 10^{-6} , one with $\phi = 0.2$ and one with $\phi = 0.04$. The associated dense output reconstruction was used for the interpolation operator $Q(\tau)$, both in the multi-rate procedure to interpolate the slow variables when integrating the fast variables, and when re-sampling the solution over a regular time grid with 0.1 s time intervals. For the multi-rate algorithms, the parameter values $\alpha_{max} = 1.2$, $\alpha_{min} = 0.5$, $\alpha = 0.9$, and $\beta = 1$ were used.

Fig. 5 shows the reference solution $u(x, t)$ along the spatial axis at five subsequent time instants. As time progresses, a sharp shock wave is formed on the right boundary of the active region, only slightly smoothed by the diffusion term, moving to the right at finite

Fig. 5 Reference solution for Burgers equation test case

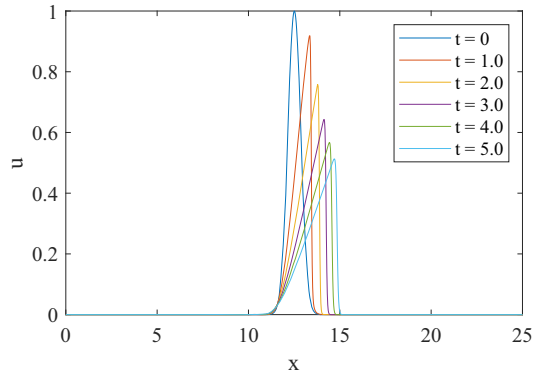
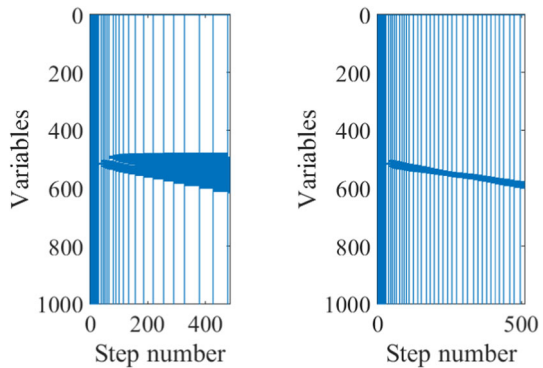


Fig. 6 Activity diagram of Burgers' equation test case, tolerance 10^{-5} , left $\phi = 0.2$, right $\phi = 0.04$



speed, followed by a trailing region where the values of u slowly get back to zero. After $t = 1$, for each time instant the variable vector can be partitioned in three subsets. One, involving a few percent of the node variables, corresponds to the fast-changing shock wave. The second, involving about 15% of the node variables, corresponds to the slow trailing wave, while the remaining part of the node vector contains values that always remain very close to zero.

The activity diagram in the case $\phi = 0.2$ is shown in the left half of Fig. 6. For this computation, relative and absolute tolerance values were set to 10^{-5} . The first 30 steps are all global steps, where the solver increases the step size from the initial value of $1.4 \cdot 10^{-4}$ to $2.3 \cdot 10^{-2}$, when the error tolerance threshold is first violated. Then, multi-rate integration kicks in, with the fast variable subsets \mathcal{V}_n^f encompassing the first two above-mentioned subsets, where basically all the action takes place in the solution, corresponding to the central blue part of the diagram. After $t = 1$, which corresponds to the 130th step, only a very few global steps, about one every 30 fast refinement steps, are taken, essentially to capture the progressive enlarging of the fast variable subsets \mathcal{V}_n^f as the shock-wave moves to the right of the spatial domain.

If a smaller value $\phi = 0.04$ is chosen, the activity diagram shown in the right half of Fig. 6 is obtained. In this case, the sets \mathcal{V}_n^f cannot be large enough to encompass the whole region around the shock with non constant values. Hence, they can only cover the much narrower subset of variables corresponding to the shock transition, moving from left to right in the spatial domain, while the degrees of freedom corresponding to the trailing region and

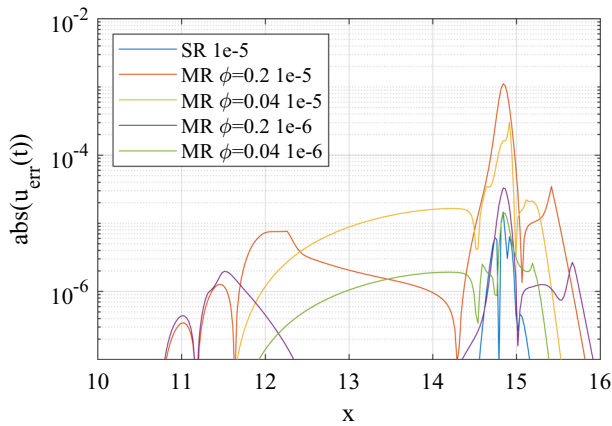


Fig. 7 Absolute error for the solution of Burgers equation

to the essentially constant part of the solution both end up in the subset of slow variables \mathcal{V}_n^s . Therefore, compared to the previous solution, there are more global steps, but on the other hand the fast steps involve a much smaller set of variables.

Fig. 7 shows the spatial distribution of the absolute error over the spatial domain at $t = 5$. The error is computed considering the difference between the single-rate and multi-rate solutions and the reference solution computed with the single rate method employed with tolerance 10^{-8} . It is to be remarked that, in this way, an estimate of the time discretization error is only obtained, since all methods use the same spatial discretization. The interaction of time and space discretization errors is an extremely important point for application to PDE solvers, which is however beyond the scope of this work. For the single-rate solution with tolerance 10^{-5} , the maximum error is about $1.5 \cdot 10^{-5}$, which is consistent with the selected tolerance. The multi-rate solutions with tolerance 10^{-5} show a significantly larger error around $x = 14.8$, where the steepest region of the shock wave is located, with maximum errors of about $3 \cdot 10^{-4}$ for $\phi = 0.04$ and 10^{-3} for $\phi = 0.02$. On the other hand, it is important to consider that the error tolerance of 10^{-5} concerns the local error over one integration step, whereas the errors shown in Fig. 7 are global errors accumulated over the entire integration time interval. Most importantly, when reducing the tolerance to 10^{-6} , the maximum absolute error is dramatically reduced to about 10^{-5} , thus validating the correct behaviour of the multi-rate error control mechanism.

Finally, Table 14 compares the performance of the single- and multi-rate simulations. As in the previous case, the overall number of steps in the multi-rate case (485) is only marginally larger than the number of steps in the single rate case (383), while the number of global steps is one order of magnitude smaller. In principle, a significant performance improvement would be expected. However, the observed simulation speed-up factor was limited, between 1 and 2 depending on the specific settings. Note that in this case it was not necessary to pre-empt the computation of the Jacobian for each global time step to achieve the best simulation time - in fact, such a choice significantly increased it. This behaviour is likely due to the test case being markedly less nonlinear than the previous one, as also clearly indicated by the fact that there were zero convergence failures reported, so that there is a clear advantage at re-using outdated Jacobians for several global time steps.

The reason for the disappointing speed-up of the multi-rate algorithm can probably be found in the overhead of the multi-rate code implementation, which requires to set up the solver and

determine the relevant section of RHS computation code to run for each fast refinement step; given the very simple expressions of the RHS terms, which require very limited computation effort, this overhead likely plays a major role in this case. The comparison of the single-rate ESDIRK3 performance with the IDA performance shows a ratio of 4.5, which can be largely attributed, as in the previous test case, to the larger number of nonlinear systems to be solved at each step (3 for ESDIRK3 vs. 1 for IDA), as well as to the better optimizations of a state-of-the-art solver like IDA. Further optimizations of the GBODE code will be necessary, which are, however, beyond the scope of this paper.

7.3 Thermal model of a large building heating system

The third test case consists of an idealized model of the thermal behaviour of a large building with $N = 100$ heated units. For simplicity, it is assumed that each unit, with temperature $T_{u,j}$, is well-insulated from the others and only interacts with a central heating system with supply temperature T_s and with the external environment at temperature T_e , which is assumed to vary sinusoidally during the day with a peak at 14:00. The thermal supply system has a temperature T_s , which is regulated by a proportional controller with fixed set point T_s^0 , that acts on the heat input Q_s . The total energy consumption of the thermal supply system is tracked by the variable E .

Each unit has a proportional temperature controller with variable set point $T_{u,j}$, acting on the command signal u_j of a heating unit with thermal conductance G_h (e.g., a fan coil with variable fan speed), which responds to the command as a first-order linear system with time constant t_h . The heating unit then enables the heat transfer between the heating fluid of the supply system and the room, with thermal power $Q_{h,j}$. Each unit also exchanges a thermal power $Q_{e,j}$ with the external environment.

The model is described by the following system of differential-algebraic equations:

$$T_e = 278.15 + 8 \cos\left(2\pi \frac{t - 14 \times 3600}{24 \times 3600}\right) \tag{34}$$

$$Q_s = \text{sat}(K_{ps} Q_{max}(T_s^0 - T_s), 0, Q_{max}) \tag{35}$$

$$Q_{ht} = \sum_1^N Q_{h,j} \tag{36}$$

$$C_s \frac{dT_s}{dt} = Q_s - Q_{ht} \tag{37}$$

$$t_h \frac{dG_{h,j}}{dt} = u_j G_{hn} - G_{h,j}, \quad j = 1, \dots, N \tag{38}$$

$$Q_{h,j} = G_{h,j}(T_s - T_{u,j}), \quad j = 1, \dots, N \tag{39}$$

$$Q_{e,j} = G_u(T_{u,j} - T_e), \quad j = 1, \dots, N \tag{40}$$

$$C_{u,j} \frac{dT_{u,j}}{dt} = Q_{h,j} - Q_{e,j}, \quad j = 1, \dots, N \tag{41}$$

$$u_j = \text{sat}(K_{pu}(T_{u,j}^0 - T_{u,j}), 0, 1), \quad j = 1, \dots, N \tag{42}$$

$$T_{u,j}^0 = f_j(t), \quad j = 1, \dots, N \tag{43}$$

$$\frac{dE}{dt} = Q_s \tag{44}$$

Table 14 Performance of single- and multi-rate ESDIRK(2)4L[2]SA methods on Burgers' equation test case

	SR	MR $\phi = 0.2$ $tol = 10^{-5}$	MR $\phi = 0.2$ $tol = 10^{-6}$	MR $\phi = 0.04$ $tol = 10^{-5}$	MR $\phi = 0.04$ $tol = 10^{-6}$	IDA // $tol = 10^{-5}$
Accepted global steps	383	49	56	67	112	393
Rejected global steps	0	0	0	0	0	0
Accepted fast steps	//	436	873	448	1110	//
Rejected fast steps	//	0	0	0	0	//
Global RHS calls	5667	1250	1809	1579	3183	431
Global Jacobian comp.	3	35	28	35	29	41
Local RHS calls	//	5390	8169	5041	9157	//
Local Jacobian comp.	//	43	50	79	162	//
Simulation time in s	0.168	0.084	0.125	0.096	0.188	0.037

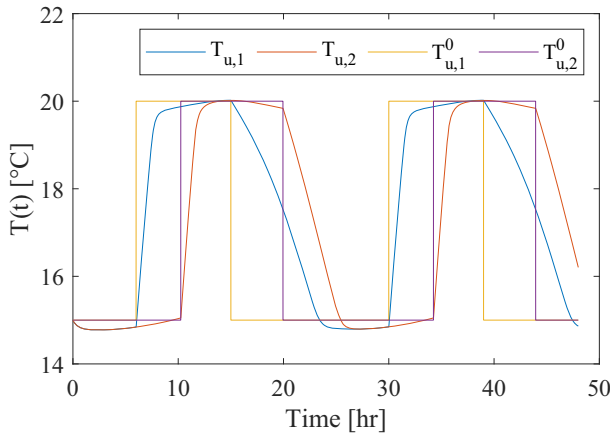


Fig. 8 Temperatures and set points for units 1 and 2

which can be easily reformulated by substitution as a system of $2N + 2$ explicit ordinary differential equations in the variables $T_s, G_{h,j}, T_{u,j}$, and E .

We assume the following values for the system parameters: $K_{ps} = 0.2, T_h = 293.15, T_l = 288.15, T_s^0 = 343.15, G_{hn} = 200, G_u = 150, Q_{max} = 0.7NG_{hn}(T_s^0 - T_h), C_s = 2 \cdot 10^6 N, t_h = 20, C_{u,j} = (1 + 0.348j/N) \times 10^7, K_{pu} = 1$. All the values of physical constants are in SI units.

The functions $f_j(t)$ are periodic with a period of one day, i.e., 86400 s. They start at T_l at midnight, get increased from T_l to T_h at a pseudo-random time between 6 and 12 am, and switch back to T_l at a pseudo-random time between 15 and 22 pm. The transitions are smooth, using the function $\text{smoothStep}(\cdot, \cdot, \cdot)$ defined as

$$\text{smoothStep}(t, t_s, \Delta t) = \frac{1}{2} \left(\tanh \left(\frac{t - t_s}{\Delta t} \right) + 1 \right), \tag{45}$$

with $\Delta t = 1$ s, while the function $\text{sat}(\cdot, \cdot, \cdot)$ is a smooth saturation function defined as

$$\text{sat}(x, x_{min}, x_{max}) = \frac{x_{max} + x_{min}}{2} + \frac{x_{max} - x_{min}}{2} \tanh \left(2 \frac{x - x_{min}}{x_{max} - x_{min}} - 1 \right). \tag{46}$$

The initial conditions of the system at $t = 0$ (corresponding to 00:00, midnight) are $T_s = T_s^0, G_{h,j} = 0, T_{u,j} = 288.15, E = 0$. The integration interval is two days, i.e $t \in [0, 172800]$.

We have used the fourth order ESDIRK method denoted as ESDIRK4(3)6L[2]SA in [24] to compute a reference single-rate solution with relative and absolute tolerance 10^{-9} , a single-rate solution with relative and absolute tolerance 10^{-5} and multi-rate solution with relative and absolute tolerance 10^{-5} . For the multi-rate algorithms, the parameter values $\alpha_{max} = 1.2, \alpha_{min} = 0.5, \alpha = 0.9, \phi = 0.05$, and $\beta = 1$ were used; the associated dense output reconstruction was used for the interpolation operator $Q(\tau)$, both in the multi-rate procedure to interpolate the slow variables when integrating the fast variables, and when re-sampling the solution over a regular time grid with 10 s intervals.

Figure 8 shows the temperatures and the respective set points for units 1 and 2. The set point for the first unit $T_{u,1}^0$ is raised around 6:00 and reduced around 14:00, while the set

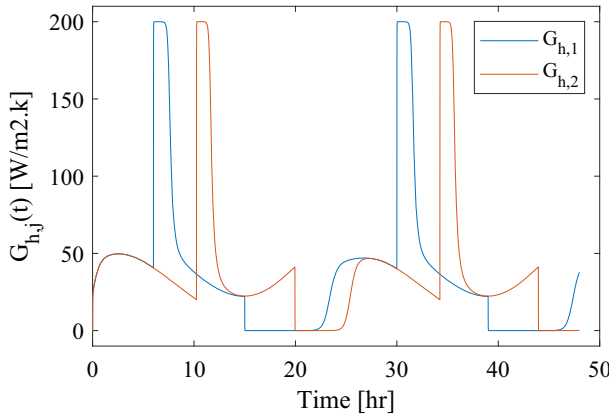
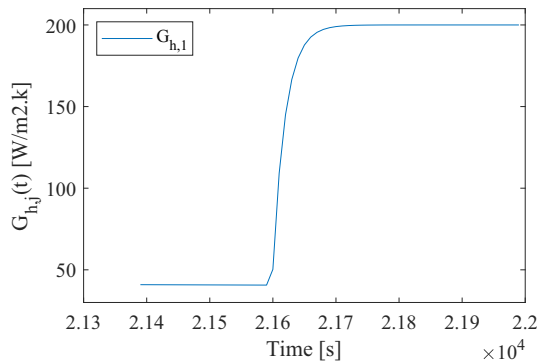


Fig. 9 Conductance of fan-coils of units 1 and 2

Fig. 10 Detail of the first rising front of fan-coil 1



point for the second unit $T_{u,2}^0$ is raised around 10:00 and reduced around 20:00. The unit temperatures $T_{u,1}$ and $T_{u,2}$ follow with some delay. Falling temperature transients are slower because they are only driven by the heat losses to the ambient.

Figure 9 shows the fan-coil conductances $G_{h,1}$ and $G_{h,2}$. Initially, they are quite low, to keep the night temperature around T_l . Subsequently, they very quickly reach the maximum value when the set point is raised, remain at the maximum for a while and then decrease once the unit temperatures approach the new set point value T_h , and drop sharply to zero when the set point is reduced to T_l in the afternoon, increasing again during the night once the temperature has fallen below the set point. The dynamics of $G_{h,1}$ is much faster than the dynamics of the temperatures, see the detail of the first rising front of $G_{h,1}$ shown in Figure 10.

Figure 11 shows the thermal power input of the supply system Q_s and the total thermal power output to the unit fan coils Q_{ht} , with a zoom-in of Q_{ht} between 7:45 and 12:15 shown in Figure 12. The supply temperature T_s is shown in Figure 13.

The reason why this system can benefit from multi-rate integration is twofold: the various occupants change their unit set points asynchronously, and the changes applied to one unit are very weakly coupled to the other units through the large thermal inertia of the supply system. Every time an occupant gets to its unit and raises the set point in the morning, or reduces it in the evening, a local fast transient is triggered on the unit's temperature control

Fig. 11 Input and output thermal power of the supply system

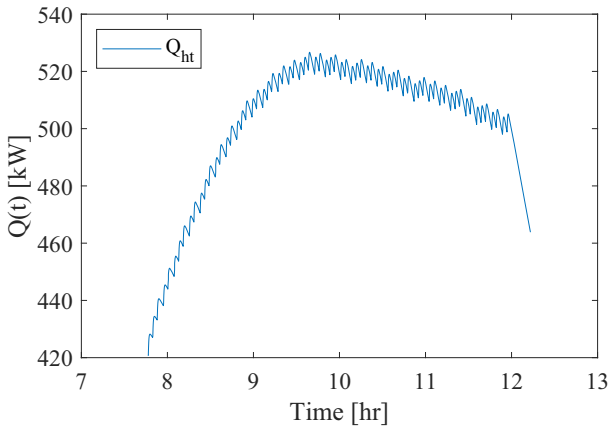
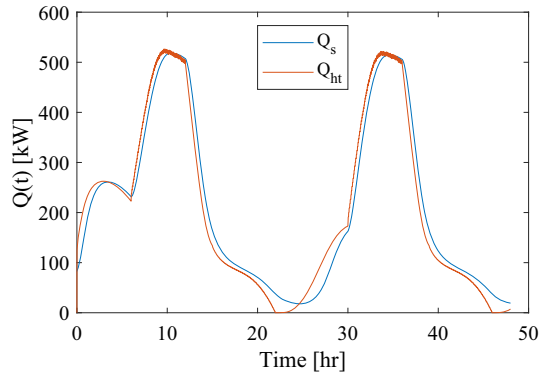
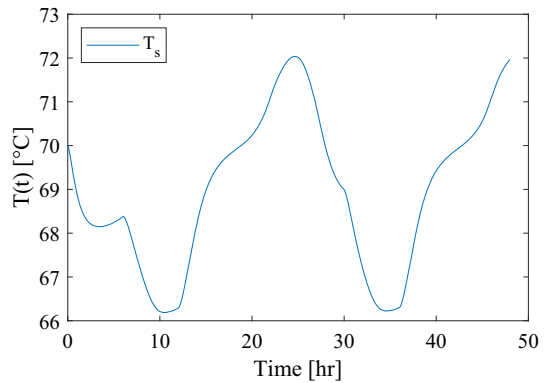


Fig. 12 Zoom-in of the total output thermal power of the supply system between 7:45 and 12:15 on day 1

Fig. 13 Heating supply temperature



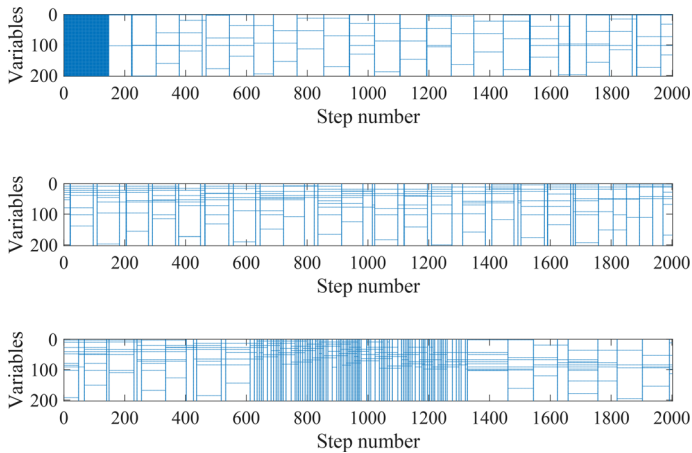


Fig. 14 Activity diagrams of the multi-rate solution

system output u_j , which requires relatively short time steps to simulate the transient of $G_{h,j}$. However, this action does not influence the other units directly, because they are insulated from each other, but rather only through the increased or decreased power consumption Q_{ht} , shown in Figure 11. Even though Q_{ht} shows fast changes, see Fig. 12, corresponding to the individual unit heating systems being turned on or off, the large inertia of the supply system causes the supply temperature T_s to remain quite smooth, as shown in Figure 13, so that the influence of these events on the heat exchanged by the other units becomes relevant only over much larger time intervals, even more so as N grows.

Hence, every set point switching transient can be handled by a set of fast variables that only includes the local heater conductance $G_{h,j}$, the unit temperature $T_{u,j}$, the supply temperature T_s , and the consumed energy E , using interpolated values for the other unit variables, $G_{h,k}, T_{u,k}, k \neq j$, which only start getting influenced by the consequences of switching in other units after hundreds of seconds. This brings down the number of DOF required to manage these transients from $2N + 2$ to $O(1)$.

Figure 14 shows three portions of the activity diagram, with rows corresponding to variables and columns to time steps. The top diagram corresponds to the beginning of the simulation, from 00:00 to 07:30. The first 145 steps to the left of the diagram correspond to the night transient from 00:00 to 06:00. During this period, no set points are changed, so there are no fast local transients taking place in the system, only the response of all the system variables to the slow sinusoidal variation of the external ambient temperature, which is handled by global time steps up to 1200 s long.

After 06:00, the unit temperature set points start being increased from T_l to T_h , in a pseudo-random fashion, triggering many local fast transient. As a consequence, the average step length is dramatically reduced, and the remaining portion of the topmost activity diagram, spanning until 07:30, shows a very sparse pattern, where the two bordering lines at the top and bottom correspond to T_u and E , which are always active, while the other shorter horizontal segments correspond to the transients of $G_{h,j}$ and $T_{u,j}$, which are handled by fast refinement steps; only a few global steps are necessary, shown in the figure as vertical blue bands).

This pattern continues until 12:00, e.g., see the the middle diagram of Figure 14, covering the time interval from 8:45 to 10:00.

Table 15 Total energy consumption computed by the ESDIRK4(3)6L[2]SA method

	Total Consumption [MWh]
Reference	10.16868431299
Single-Rate	10.16868431246
Multi-Rate	10.16822913061

Table 16 Performance of single- and multi-rate ESDIRK4(3)6L[2]SA method

	Single-Rate	Multi-Rate	CVODE
Accepted global steps	27642	1106	42257
Rejected global steps (error test failure)	5918	74	2470
Accepted fast steps	//	31074	//
Rejected fast steps	//	5680	//
Global RHS calls	1022880	73169	55042
Global Jacobian computations	763	193	788
Local RHS calls	//	1056965	//
Local Jacobian computations	//	2546	//
Simulation time in s	14.7	2.62	5.57

The bottom diagram of Figure 14, which covers the interval from 11:30 to 15:30, shows the sparse patterns ending at 12:00, when set points stop being changed, followed by a pattern composed of mostly global steps until between 12:00 and 15:00, when the room temperatures mostly react to the slowly changing external temperature, after which set points start changing again, bringing the sparse pattern in again, and so on.

The most important result of this simulation is the final value of the energy consumption E , reported in Table 15 for the three solutions. The multi-rate result is affected by a larger error than the single-rate one, but it still has 5 correct significant digits, which is in agreement with the set relative and absolute tolerances 10^{-5} , despite them being set on the local integration error and not on the cumulated error that affects the total energy consumption.

The performance comparison between the single-rate and the multi-rate ESDIRK4(3)6L[2]SA method is shown in Table 16. As in the previous two cases, the number of total time steps required by the multi-rate method is only slightly larger than the number of steps of the single-rate method; in this case, the number of global steps is reduced by a factor 25 compared to the single-rate case. The speed-up factor of the multi-rate method over the single rate one is around 5.6.

In this case, unfortunately, the IDA solver did not produce satisfactory results in OpenModelica, due to some issues in the computation of the Jacobians. Therefore, the stiff CVODE solver was used instead, which is also implemented in SUNDIALS and is based on a BDF method. As shown in the last column of Table 16, CVODE is still faster than the single-rate implementation of ESDIRK4 in the GBODE solver, mainly due to the reduced number of right-hand-side computations. On the other hand, it turns out to be slower than the multi-rate ESDIRK4 method.

This already good result can be improved in the future by further optimizations of our code, both regarding the efficient solution of the global time step implicit equations and the efficient selective computation of the required parts of the right hand side for fast refinement states.

8 Conclusions and Future Work

We have presented a novel approach to self-adjusting multi-rate methods, that allows to obtain an effective multi-rate implementation of a generic Runge-Kutta (RK) method by combining a self-adjusting multi-rate technique with standard time step adaptation methods. Only a small percentage of the variables, associated with the largest values of an error estimator are marked as fast variables. When the global step is sufficient to guarantee a given error tolerance for the slow variables, but not for the fast ones, the multi-rate procedure is employed to achieve uniform accuracy at reduced computational cost.

We have also derived a general linear stability analysis, valid for explicit RK, DIRK and ESDIRK multi-rate methods with an arbitrary number of sub-steps for the active components. We have presented a thorough review of the literature on multi-rate stability analyses, which highlights the advantages of our approach. Furthermore, we have introduced a novel, physically motivated model problem that allows to assess the stability of multi-rate approaches in a context more relevant for realistic applications. The stability analysis has been performed on some examples of ERK and ESDIRK methods, highlighting the potential gains in efficiency and accuracy that can be achieved by the multi-rate approach in the regime of weak coupling between fast and slow variables.

The proposed multi-rate methods have been implemented efficiently in the framework of the OpenModelica software [11]. Three numerical benchmarks have been considered, representing two idealized engineering systems and a basic discretization of a nonlinear PDE typical of Computational Fluid Dynamics applications. The results obtained demonstrate the significant cost reductions with respect to the corresponding single-rate methods implemented the same framework and a computational efficiency that starts to be comparable to that of much more mature single-rate solvers.

In future work, we will further pursue the optimization of the present OpenModelica implementation. In particular, the need for different optimization strategies in the multi-rate and single-rate case, highlighted by our numerical experiments, will be further investigated, in order to take full advantage of the multi-rate DOF reduction. This will allow the effective application of multi-rate methods to larger scale PDE problems as well as to more realistic engineering problems.

A Butcher Tableaux of RK Methods

For completeness, we report in this Appendix the Butcher tableaux of the methods considered in the stability analysis, along with the coefficients of their corresponding dense output interpolators. Table 17 contains the coefficients defining the classical fourth order explicit RK method. Table 18 contains the coefficients defining the fourth order RK method with optimal continuous output introduced in [32]. The matrix B^* containing the coefficients for

Table 17 Butcher tableaux of the classical fourth order ERK method

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

Table 18 Butcher tableaux of optimal explicit RK method with continuous output

0	0	0	0	0	0	0
$\frac{1}{6}$	$\frac{1}{6}$	0	0	0	0	0
$\frac{11}{37}$	$\frac{44}{1369}$	$\frac{363}{1369}$	0	0	0	0
$\frac{11}{17}$	$\frac{3388}{4913}$	$-\frac{8349}{4913}$	$-\frac{8140}{4913}$	0	0	0
$\frac{13}{15}$	$-\frac{36764}{408375}$	$\frac{767}{1125}$	$-\frac{32708}{136125}$	$\frac{210392}{408375}$	0	0
1	$\frac{1697}{18876}$	0	$\frac{50653}{116160}$	$\frac{299693}{1626240}$	$\frac{3375}{11648}$	0
	$\frac{101}{363}$	0	$-\frac{1369}{14520}$	$\frac{11849}{14520}$	0	0

Table 19 Matrix of continuous output coefficients b_{ij}^* for explicit RK method of [32]

b_{ij}^*	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 1$	1	$-\frac{104217}{37466}$	$\frac{1806901}{618189}$	$-\frac{866577}{824252}$
$i = 2$	0	0	0	0
$i = 3$	0	$\frac{861101}{230560}$	$-\frac{2178079}{380424}$	$\frac{12308679}{5072320}$
$i = 4$	0	$-\frac{63869}{293440}$	$\frac{6244423}{5325936}$	$-\frac{7816583}{10144640}$
$i = 5$	0	$-\frac{1522125}{762944}$	$\frac{982125}{190736}$	$-\frac{624375}{217984}$
$i = 6$	0	$\frac{165}{131}$	$-\frac{461}{131}$	$\frac{296}{131}$

the dense output formula (13) associated to this method are given in Table 19. Setting then

$$\begin{aligned}
 \gamma &= 0.43586652150845899941601945, \\
 c_3 &= \frac{3}{5}, \quad \bar{a} = 1 - 6\gamma + 6\gamma^2 \\
 a_{32} &= c_3 \frac{c_3 - 2\gamma}{4\gamma} \quad a_{31} = c_3 - a_{32} - \gamma \\
 b_2 &= \frac{-2 + 3c_3 + 6\gamma(1 - c_3)}{12\gamma(c_3 - 2\gamma)} \quad b_3 = \frac{\bar{a}}{3c_3(c_3 - 2\gamma)} \\
 b_1 &= 1 - b_2 - b_3 - \gamma,
 \end{aligned} \tag{47}$$

we report in Table 20 the coefficients of the third order ESDIRK method denoted as ESDIRK3(2)4L[2]SA in [24]. The corresponding matrix B^* containing the coefficients for the dense output formula (13) are given in Table 21.

Table 22 contains instead the coefficients defining the fourth order ESDIRK method denoted as ESDIRK4(3)6L[2]SA in [24], where we have set instead

$$\gamma = 1/4c_3 = \frac{2 - \sqrt{2}}{4} \quad c_4 = \frac{5}{8} \quad c_5 = \frac{26}{25}$$

Table 20 Butcher tableaux of the ESDIRK3(2)4L[2]SA method

0	0	0	0	0
2γ	γ	γ	0	0
$\frac{3}{5}$	a_{31}	a_{32}	γ	0
1	b_1	b_2	b_3	γ
	b_1	b_2	b_3	γ

Table 21 Matrix of continuous output coefficients b_{ij}^* for the ESDIRK3(2)4L[2]SA method

b_{ij}^*	$j = 1$	$j = 2$	$j = 3$
$i = 1$	$\frac{6071615849858}{5506968783323}$	$-\frac{9135504192562}{5563158936341}$	$\frac{5884850621193}{8091909798020}$
$i = 2$	$\frac{24823866123060}{14064067831369}$	$-\frac{184358657789355}{34679930461469}$	$\frac{40093531604824}{13565043189019}$
$i = 3$	$-\frac{4639021340861}{5641321412596}$	$-\frac{36951656213070}{8103384546449}$	$-\frac{9445293799577}{3414897167914}$
$i = 4$	$-\frac{4782987747279}{4575882152666}$	$\frac{22547150295437}{9402010570133}$	$-\frac{8621837051676}{9402290144509}$

Table 22 Butcher tableaux of the ESDIRK4(3)6L[2]SA method

0	0	0	0	0	0	0
2γ	γ	γ	0	0	0	0
c_3	a_{31}	a_{32}	γ	0	0	0
c_4	a_{41}	a_{42}	a_{43}	γ	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	γ	0
1	b_1	b_2	b_3	b_4	b_5	γ
	b_1	b_2	b_3	b_4	b_5	γ

$$\begin{aligned}
 a_{32} &= \frac{1 - \sqrt{2}}{8} & a_{31} &= c_3 - a_{32} - \gamma \\
 a_{42} &= \frac{5 - 7\sqrt{2}}{64} & a_{43} &= 7\frac{1 + \sqrt{2}}{32} & a_{41} &= c_4 - a_{42} - a_{43} - \gamma \\
 a_{52} &= \frac{-13796 - 54539\sqrt{2}}{125000} & a_{53} &= \frac{506605 + 132109\sqrt{2}}{437500} \\
 a_{54} &= 166\frac{-97 + 376\sqrt{2}}{109375} & a_{51} &= c_5 - a_{52} - a_{53} - a_{54} - \gamma \\
 b_2 &= \frac{1181 - 987\sqrt{2}}{13782} & b_3 &= 47\frac{-267 + 1783\sqrt{2}}{273343} \\
 b_4 &= -16\frac{-22922 + 3525\sqrt{2}}{571953} & b_5 &= -15625\frac{97 + 376\sqrt{2}}{90749876} \\
 b_1 &= 1 - b_2 - b_3 - b_4 - b_5 - \gamma.
 \end{aligned} \tag{48}$$

The corresponding matrix B^* containing the coefficients for the dense output formula (13) are given in Table 23.

Acknowledgements We thank the reviewer and the Associate Editor for their constructive remarks, which greatly helped in improving the first version of this paper. B.B. carried out part of this research during a sabbatical at Politecnico di Milano in 2022, which was partially supported by the Visiting Faculty Program of

Table 23 Matrix of continuous output coefficients b_{ij}^* for the ESDIRK4(3)6L[2]SA method

b_{ij}^*	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 1$	11963910384665 12483345430363	– 69996760330788 18526599551455	32473635429419 7030701510665	– 14668528638623 8083464301755
$i = 2$	11963910384665 12483345430363	– 69996760330788 18526599551455	32473635429419 7030701510665	– 14668528638623 8083464301755
$i = 3$	– 28603264624 1970169629981	102610171905103 26266659717953	– 38866317253841 6249835826165	21103455885091 7774428730952
$i = 4$	– 3524425447183 2683177070205	74957623907620 12279805097313	– 26705717223886 4265677133337	30155591475533 15293695940061
$i = 5$	– 17173522440186 10195024317061	113853199235633 9983266320290	– 121105382143155 6658412667527	119853375102088 14336240079991
$i = 6$	27308879169709 13030500014233	– 84229392543950 6077740599399	1102028547503824 51424476870755	– 63602213973224 6753880425717

Politecnico di Milano. S.F.G, M.G.M and L.B. have also received support through the PID2021-123153OB-C21 project of the Science and Innovation Ministry of the Spanish government.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Declarations

Conflicts of Interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Andrus, J.: Numerical solution of systems of ordinary differential equations separated into subsystems. *SIAM Journal of Numerical Analysis* **16**, 605–611 (1979)
- Arndt, D., Bangerth, W., Feder, M., Fehling, M., Gassmüller, R., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.P., Sticks, S., Turcksin, B., Wells, D.: The deal II library, version 9.4. *J. Numer. Math.* **30**(3), 231–246 (2022)
- Bangerth, W., Hartmann, R., Kanschat, G.: deal II: a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software* **33**, 24–51 (2007)
- Bank, R., Coughran, W., Fichtner, W., Grosse, E., Rose, D., Smith, R.: Transient simulation of silicon devices and circuits. *IEEE Trans. Electron Devices* **32**, 1992–2007 (1985)
- Belytschko, T., Smolinski, P., Liu, W.: Stability of multi-time step partitioned integrators for first-order finite element systems. *Comput. Methods Appl. Mech. Eng.* **49**, 281–297 (1985)
- Bonaventura, L., Casella, F., Carciopolo, L.D., Ranade, A.: A self adjusting multirate algorithm for robust time discretization of partial differential equations. *Comput. Math. Appl.* **79**, 2086–2098 (2020)
- Constantinescu, E., Sandu, A.: Multirate timestepping methods for hyperbolic conservation laws. *J. Sci. Comput.* **33**, 239–278 (2007)
- Daniel, W.: The subcycled Newmark algorithm. *Comput. Mech.* **20**, 272–281 (1997)
- Daniel, W.: A study of the stability of subcycling algorithms in structural dynamics. *Comput. Methods Appl. Mech. Eng.* **156**, 1–13 (1998)
- Fong, W., Darve, E., Lew, A.: Stability of asynchronous variational integrators. *J. Comput. Phys.* **227**, 8367–8394 (2008)

11. Fritzon, P., Pop, A., Abdelhak, K., Ashgar, A., Bachmann, B., Braun, W., Bouskela, D., Braun, R., Buffoni, L., Casella, F., Castro, R., Franke, R., Fritzon, D., Gebremedhin, M., Heuermann, A., Lie, B., Mengist, A., Mikelsens, L., Moudgalya, K., Ochel, L., Palanisamy, A., Ruge, V., Schamai, W., Sjölund, M., Thiele, B., Tinnerholm, J., Östlund, P.: The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *MIC—Model. Identif. Control* **41**, 241–295 (2020)
12. Gear, C., Wells, D.: Multirate linear multistep methods. *BIT Numer. Math.* **24**, 484–502 (1984)
13. Gravoil, A., Combescure, A., Brun, M.: Heterogeneous asynchronous time integrators for computational structural dynamics. *Int. J. Numer. Meth. Eng.* **102**, 202–232 (2015)
14. Grote, M., Mehlin, M., Mitkova, T.: Runge-Kutta-based explicit local time-stepping methods for wave propagation. *SIAM J. Sci. Comput.* **37**, A747–A775 (2015)
15. Günther, M., Rentrop, P.: Multirate row methods and latency of electric circuits. *Appl. Numer. Math.* **13**, 83–102 (1993)
16. Günther, M., Sandu, A.: Multirate generalized additive Runge Kutta methods. *Numer. Math.* **133**, 497–524 (2016)
17. Günther, M., Sandu, A.: Multirate linearly-implicit GARK schemes. *BIT Numer. Math.* **62**, 869–901 (2022)
18. Hairer, E., Nørsett, S., Wanner, G.: *Solving Ordinary Differential Equations I Nonstiff problems*, 2nd edn. Springer, Berlin (2000)
19. Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)* **31**(3), 363–396 (2005). <https://doi.org/10.1145/1089014.1089020>
20. Hosea, M., Shampine, L.: Analysis and implementation of TR-BDF2. *Appl. Numer. Math.* **20**, 21–37 (1996)
21. Hundsdoerfer, W., Savcenko, V.: Analysis of a multirate theta-method for stiff ODEs. *Appl. Numer. Math.* **59**, 693–706 (2009)
22. Kang, S., Constantinescu, E.: Entropy-Preserving and Entropy-Stable Relaxation IMEX and Multirate Time-Stepping Methods. *J. Sci. Comput.* **93**, 23 (2022)
23. Kang, S., Dener, A., Hamilton, A., Zhang, H., Constantinescu, E., Jacob, R.: Multirate partitioned Runge-Kutta methods for coupled Navier-Stokes equations. *Computers & Fluids* **264**, 105964 (2023)
24. Kennedy, C.A., Carpenter, M.: Diagonally implicit Runge-Kutta methods for ordinary differential equations, a review. Tech. Rep. TM-2016-219173, NASA (2016)
25. Kennedy, C.A., Carpenter, M.: Diagonally implicit Runge-Kutta methods for stiff ODEs. *Appl. Numer. Math.* **146**, 221–244 (2019)
26. Krivodonova, L.: An efficient local time-stepping scheme for solution of nonlinear conservation laws. *J. Comput. Phys.* **229**, 8537–8551 (2010)
27. Kuhn, K., Lang, J.: Comparison of the asymptotic stability for multirate Rosenbrock methods. *J. Comput. Appl. Math.* **262**, 139–149 (2014)
28. Kværnø, A.: Stability of multirate Runge-Kutta schemes. *International Journal of Differential Equations and Applications* **1**, 97–105 (2000)
29. Mattsson, S.E., Elmqvist, H., Otter, M.: Physical system modeling with Modelica. *Control. Eng. Pract.* **6**(4), 501–510 (1998)
30. Orlando, G., Barbante, P.F., Bonaventura, L.: An efficient IMEX-DG solver for the compressible Navier-Stokes equations for non-ideal gases. *J. Comput. Phys.* **471**, 111653 (2022)
31. Orlando, G., Rocca, A.D., Barbante, P.F., Bonaventura, L., Parolini, N.: An efficient and accurate implicit DG solver for the incompressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **94**, 1484–1516 (2022)
32. Owren, B., Zennaro, M.: Derivation of efficient, continuous, explicit Runge-Kutta methods. *SIAM Journal of Scientific and Statistical Computing* **13**, 1488–1501 (1992)
33. Quarteroni, A., Valli, A.: *Domain decomposition methods for partial differential equations*. Oxford University Press, Oxford (1999)
34. Rice, J.: Split Runge-Kutta methods for simultaneous equations. *Journal of Research of the National Institute of Standards and Technology* **60**, (1960)
35. Roberts, S., Sarshar, A., Sandu, A.: Coupled multirate infinitesimal GARK schemes for stiff systems with multiple time scales. *SIAM J. Sci. Comput.* **42**, A1609–A1638 (2020)
36. Rodríguez-Gómez, G., González-Casanova, P., Martínez-Carballido, J.: Computing general companion matrices and stability regions of multirate methods. *Int. J. Numer. Meth. Eng.* **61**, 255–273 (2004)
37. Sand, J., Skelboe, S.: Stability of backward Euler multirate methods and convergence of waveform relaxation. *BIT Numer. Math.* **32**, 350–366 (1992)
38. Sandu, A.: A class of multirate infinitesimal gark methods. *SIAM J. Numer. Anal.* **57**, 2300–2327 (2019)

39. Sandu, A., Günther, M., Roberts, S.: Linearly implicit GARK schemes. *Appl. Numer. Math.* **161**, 286–310 (2021)
40. Sarshar, A., Roberts, S., Sandu, A.: Design of high-order decoupled multirate GARK schemes. *SIAM J. Sci. Comput.* **41**, A816–A847 (2019)
41. Savcenco, V.: Comparison of the asymptotic stability properties for two multirate strategies. *J. Comput. Appl. Math.* **220**, 508–524 (2008)
42. Savcenco, V., Hundsdorfer, W., Verwer, J.: A multirate time stepping strategy for stiff ordinary differential equations. *BIT Numer. Math.* **47**, 137–155 (2007)
43. Schlegel, M., Knoth, O., Arnold, M., Wolke, R.: Multirate Runge-Kutta schemes for advection equations. *J. Comput. Appl. Math.* **226**, 345–357 (2009)
44. Seny, B., Lambrechts, J., Toulorge, T., Legat, V., Remacle, J.: An efficient parallel implementation of explicit multirate Runge-Kutta schemes for discontinuous Galerkin computations. *J. Comput. Phys.* **256**, 135–160 (2014)
45. Skelboe, S.: Stability properties of backward differentiation multirate formulas. *Appl. Numer. Math.* **5**, 151–160 (1989)
46. Skelboe, S., Andersen, P.: Stability properties of backward Euler multirate formulas. *SIAM J. Sci. Stat. Comput.* **10**, 1000–1009 (1989)
47. Smolinski, P., Sleith, S., Belytschko, T.: Stability of an explicit multi-time step integration algorithm for linear structural dynamics equations. *Comput. Mech.* **18**, 236–244 (1996)
48. Trahan, C., Dawson, C.: Local time-stepping in Runge-Kutta discontinuous Galerkin finite element methods applied to the shallow-water equations. *Comput. Methods Appl. Mech. Eng.* **217**, 139–152 (2012)
49. Verhoeven, A., Tasić, B., Beelen, T., ter Maten, E., Mattheij, R.: Automatic partitioning for multirate methods. In: *Scientific Computing in Electrical Engineering*, pp. 229–236. Springer (2007)
50. Verhoeven, A., Maten, E.T., Mattheij, R., Tasić, B.: Stability analysis of the BDF slowest-first multirate methods. *Int. J. Comput. Math.* **84**, 895–923 (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.