A class of multirate multiderivative schemes

J. Schütz, A. Ishimwe, A. Moradi and A. Thenery Manikantan

UHasselt Computational Mathematics Preprint

July 5th, 2024

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

00 (2024) 1–16

Journal Logo

A class of multirate multiderivative schemes

Jochen Schütz[†], Ange Ishimwe[⋆], Afsane Moradi[∗], Arjun Thenery Manikantan[†]

†Faculty of Sciences and Data Science Institute, Hasselt University, Agoralaan Gebouw D, BE-3590 Diepenbeek, Belgium

⋆Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, Avenue Georges Lemaître 4, BE-1348 Louvain-la-Neuve, Belgium

∗Institute of Analysis and Numerics, Otto von Guericke University Magdeburg, Universitätsplatz 2, DE-39106 Magdeburg, Germany

Abstract

In this work, we treat the numerical resolution of ordinary differential equations (ODEs) that contain both stiff and non-stiff terms, where these terms can be identified and separated, and where the stiff terms are 'easier' to evaluate than the non-stiff terms. In [Wensch, Knoth, Galant, BIT Numer Math 49 (2009), pp. 449–473], a class of so-called multirate schemes has been proposed to efficiently resolve said ODEs. Here, we extend this class of schemes by adding multiple temporal derivatives of the non-stiff part to the formulation. Order conditions and simplified order conditions of up to order four are derived in this work. Through this modification to the original algorithm, we can devise schemes with lesser stages at the same order. In particular, we devise a four-stage fourth-order scheme. The efficacy of the proposed methods is demonstrated through numerical experiments.

Keywords: Multirate, multiderivative, singularly perturbed ODE

1. Introduction

Ordinary differential equations serve as a fundamental tool for comprehending and exploring complex dynamics. Within these dynamics, phenomena can manifest across a spectrum of temporal scales. This implies that the characteristics of the phenomena may vary widely in terms of their time scales, with some exhibiting significantly smaller characteristic times compared to others. Such terms, known as stiff terms, pose unique challenges in computational analysis. To fix the ideas, let us consider the ODE

$$y'(t) = F(y) \equiv f(y) + g(y), \tag{1.1}$$

where g denotes the 'stiff' or fast part and f the 'non-stiff' or slow part. Obviously, the definition of stiffness is non-trivial [1]. For this work, we interpret it in the way that f should be treated with a 'large' timestep Δt, while g should be treated with a 'small' timestep $\Delta\tau < \Delta t$. This type of equations frequently arises in singularly perturbed equations such as, for example, the low-Mach Navier-Stokes equations [2] or relaxation problems [3]. Normally, because of the stiff term, a small time step has to be used for explicit schemes, leading to long simulations. However, in some cases, this stiff term g can be evaluated more easily than the non-stiff terms. For instance, this is the case for physical problems where one spatial dimension is significantly less important than the others. Atmospheric or hydrodynamic models are examples of this. In these models, the horizontal lengths of the domains are much larger than the vertical one. In this configuration, fast phenomena with small characteristics are computed using two dimensional terms, while the slow ones are represented by three dimensional ones. For atmospheric models [4, 5], waves with a characteristic speed typically of 300 m/s are much faster than velocity of the air. For ocean models [6], external gravity waves are the fast processes with a characteristic velocity that can be 100 times the one of others phenomena.

To address these potential challenges, high-order multirate schemes have been introduced in the seminal work [7]. They have been quite heavily extended, for a highly incomplete list of references, see [8, 9, 10, 11, 12, 13, 14, 15, 16] and the references therein. The underlying idea of these schemes is to compute the parts corresponding to f – the non-stiff part – with an explicit Runge-Kutta scheme of timestep Δt, and the parts corresponding to g using another explicit scheme of timestep $\Delta\tau < \Delta t$. Obviously, as for any coupled scheme, not any two schemes can be combined, but they need to fulfill certain coupling conditions to preserve the order of accuracy. Typically, these come in the form of algebraic requirements on the schemes' coefficients.

In this work, we extend the work of [7] to deal with multiderivative Runge-Kutta schemes for the slow part. Multiderivative schemes are schemes that do not only incorporate y', which is $f(y) + g(y)$ as visible from (1.1), but also (parts of) y'' and higher derivatives, see, e.g., [17, 18, 19, 20] and the references therein. In this work, we use those parts of the higher derivatives of y that are associated to f, i.e., we use for example the term $f(y)^{\{1\}} \equiv f'(y)F(y)$, see Rem. 1 for more information. This way, one arrives at higher order schemes with fewer stages than without this addition. In particular, we show a fourth-order multirate scheme with only four stages.

This article is structured as follows. It begins by introducing the multiderivative split-explicit time integrator method in Section 2, along with some illustrative and newly developed examples. The subsequent section presents the order conditions. Sections 4 and 5 explore the properties and numerical results of the presented schemes on various ODEs. Finally, a conclusion is provided at the end of this article.

2. Split-explicit time integration methods

In this section, we give the general formulation of a multiderivative-multirate scheme; and then, in anticipation of the order conditions in Sec. 3, already show some novel schemes particularly developed for this work.

2.1. Formulation

Following [7], we define a multirate scheme in a semi-discrete way. While f is already discretized through an explicit multiderivative Runge-Kutta method, the equation associated to g is left continuous in time. In this work, we use multiderivative Runge-Kutta methods with up to four temporal derivatives in total, see also Rem. 1. In all what follows, $\Delta t > 0$ is a given time-step size that can be either constant throughout the algoritm or adaptive.

Definition 1. *(Multiderivative multirate scheme) For a given number of stages $s \in \mathbb{N}$ and a given number of derivatives $1 \leq m \leq 4$, we assume that the matrices α, γ, $\beta^{\{k\}} \in \mathbb{R}^{(s+1)\times(s+1)}$, $0 \leq k \leq m-1$, are given and are all strictly lower-triangular to obtain an explicit scheme. As in [7, eq. (2.3)], we assume the balancing condition*

$$d_i = \sum_{j=1}^{s+1} \beta_{ij}, \qquad 1 \leq i \leq s+1. \tag{2.1}$$

Then, the semi-discrete multiderivative multirate scheme is given by $(1 \leq i \leq s+1)$

$$\begin{aligned}
Z_{ni}(0;\Delta t) &= y_n + \sum_{j=1}^{i-1} \alpha_{ij}\left(Y_{nj}(\Delta t) - y_n\right), \\
\partial_\tau Z_{ni}(\tau;\Delta t) &= \frac{1}{\Delta t}\sum_{j=1}^{i-1} \gamma_{ij}\left(Y_{nj}(\Delta t) - y_n\right) + d_i g(Z_{ni}(\tau;\Delta t)) + \sum_{k=0}^{m-1}\sum_{j=1}^{i-1} \Delta t^k \beta_{ij}^{\{k\}} f^{\{k\}}(Y_{nj}(\Delta t)), \\
Y_{ni}(\Delta t) &= Z_{ni}(\Delta t;\Delta t);
\end{aligned} \tag{2.2}$$

the update is then given by

$$y_{n+1} = Y_{n,(s+1)}(\Delta t).$$

For this definition, we have used

$$
\begin{aligned}
f^{\{0\}}(y) &:= f(y),\\
f^{\{1\}}(y) &:= f_y F,\\
f^{\{2\}}(y) &:= f_{yy}(F,F) + f_y(f_y + g_y)F,\\
f^{\{3\}}(y) &:= f_{yyy}(F,F,F) + 3f_{yy}((f_y + g_y)F, F) + f_y\left(f_{yy}(F,F) + g_{yy}(F,F)\right) + f_y(f_y + g_y)(f_y + g_y)F.
\end{aligned} \tag{2.3}
$$

$f_y(y)$ denotes the Jacobian of f w.r.t. to y. $f_{yy}(\cdot,\cdot)$ and $f_{yyy}(\cdot,\cdot,\cdot)$ denote the second and third order derivatives of f w.r.t. to y, respectively. For an easier presentation, we have omitted the argument (y). The colors introduced here will also be used in the presentation of the order conditions.

Remark 1. *Let us make some comments here:*

- *Please note that for the exact solution y to (1.1) and $k \in \mathbb{N}^{\geq 0}$, there holds*

$$\frac{\mathrm{d}^{\mathrm{k}}}{\mathrm{d}t^k} f(y) = f^{\{k\}}(y).$$

  If g was identically zero, this would mean $y''(t) = f^{\{1\}}(y(t))$, $y'''(t) = f^{\{2\}}(y(t))$ and so on, hence the term multiderivative. While a two-derivative scheme (i.e., using $f^{\{1\}}(y)$) has been proven to be of relevance in practice, see, e.g, [21], the higher-derivative schemes are included for scientific curiosity. If they are to be used, the higher derivatives should typically be approximated by some sort of finite difference, see [22, 23, 24, 25]

- *The scheme from Def. 1 is still continuous in the pseudo-time τ. In the numerical experiments, this will be discretized using an explicit* one-derivative *Runge-Kutta method with a finer timestep-size $\Delta\tau \leq \Delta t$. The reason that we do not use a multi-derivative Runge-Kutta method lies in the fact that for $g^{\{k\}}$, $k \geq 1$, one would need evaluations of f and its derivatives again. This would be unattractive given that we consider f to be more difficult to evaluate than g.*

- *The magenta terms in Def. 1 distinguish the algorithm that we propose here from the one in [7]. These extra terms will be the reason that we can construct schemes with less stages while keeping the order of convergence.*

As in [7, Thm. 2.1], Def. 1 yields a multiderivative Runge-Kutta method in the case that $g \equiv 0$, with Butcher tableaux

$$A^{\{k\}} := R\beta^{\{k\}}, \qquad 0 \leq k \leq m-1, \tag{2.4}$$

where

$$R := (\mathrm{Id} - \alpha - \gamma)^{-1}. \tag{2.5}$$

Typically, we will indicate schemes by giving A, α and γ rather then the β.

2.2. Schemes

Based on the order conditions to be shown in Sec. 3, we have developed a couple of novel schemes. In particular, here, we present

- a class of two-stage, third-order two-derivative schemes depending on a parameter ξ, called Mul3s2m2(ξ)
- a four-stage, fourth-order two-derivative scheme, called Mul4s4m2,
- a three-stage, fourth-order three-derivative scheme, called Mul4s3m3.

Third order two-stage two-derivative scheme Mul3s2m2(ξ). Let $\xi \in \mathbb{R}^{\neq \frac{-1}{6}}$ be a free parameter and define

$$c_1 := 2\xi + \frac{1}{3}, \qquad \dot{b}_1 = \frac{3\xi}{6\xi + 1}, \qquad \dot{b}_2 = \frac{1/2}{6\xi + 1}.$$

Then, the Butcher-tableaux of the schemes are defined by

$$A^{\{0\}} = \begin{pmatrix} 0 & & \\ c_1 & 0 & \\ 1 & 0 & 0 \end{pmatrix}, \qquad A^{\{1\}} = \begin{pmatrix} 0 & & \\ \xi & 0 & \\ \dot{b}_1 & \dot{b}_2 & 0 \end{pmatrix}. \tag{2.6}$$

If we set $\alpha = \begin{pmatrix} 0 & & \\ 0 & 0 & \\ 0 & 1 & 0 \end{pmatrix}$, and $\gamma = 0$, then it can be deduced from the order conditions to be presented in Sec. 3 that the multirate scheme with $\beta = R^{-1}A$ and $\dot{\beta} = R^{-1}\dot{A}$ is third-order convergent.

Fourth order four-stage two-derivative scheme Mul4s4m2. This scheme, and also the next scheme, have been developed by solving the simplified order conditions, see Thm. 2, numerically in Matlab with the help of fsolve.

$$A^{\{0\}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.644528962237943 & 0 & 0 & 0 & 0 \\ 0 & 0.793930203564751 & 0 & 0 & 0 \\ 0 & 0.651368938661906 & 0.234630026296709 & 0 & 0 \\ 0.368783295148086 & 0.361990106948867 & 0.147750352586748 & 0.121476245316299 & 0 \end{pmatrix}, \tag{2.7}$$

$$A^{\{1\}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.019204137009700 & 0 & 0 & 0 & 0 \\ 0 & 1.074197913721907 & 0 & 0 & 0 \\ 0 & -0.328894199359934 & -0.868581157332243 & 0 & 0 \\ 0.046047593117438 & -0.004291996212853 & 0 & 0 & 0 \end{pmatrix} \tag{2.8}$$

α is set to $\alpha = \begin{pmatrix} 0 & & & \\ 0 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 \end{pmatrix}$, and $\gamma = 0$.

Fourth-order three-stage three-derivative scheme Mul4s3m3.

$$A^{\{0\}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1.009283680769299 & 0 & 0 & 0 \\ 3.720878355840538 & -2.718495837492225 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$A^{\{1\}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.253296309203584 & 0 & 0 & 0 \\ -3.356309948891324 & -2.584529228478059 & 0 & 0 \\ -0.331202647364177 & 0.855031437707487 & -0.023828790343315 & 0 \end{pmatrix} \tag{2.9}$$

$$A^{\{2\}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.075395834891222 & 0 & 0 & 0 \\ 0.989887257282753 & 1.428802815206199 & 0 & 0 \\ -0.297547643762234 & -0.882455016628254 & 0.507585613307806 & 0 \end{pmatrix}$$

α and γ are set in exactly the same way as for the scheme before. For comparison, please note that the fourth-order one-derivative scheme presented in [26] has five stages and uses a γ that is different from zero.

3. Order conditions

In this section, we investigate the order of consistency of the scheme presented in Def. 1. Although other approaches are possible as well, e.g., via generalized additive Runge-Kutta methods [8], we rely very heavily on the approach presented in [7] and later extended in [26]. As only the magenta terms in Def. 1 change in comparison to [7, 26], it is clear that the order conditions also look rather similar, with the addition of the multiderivative order conditions. For comparison, order conditions of up to order four for explicit two derivative schemes are, e.g., given in [19], for three derivative schemes in [27]; and for one-derivative multirate schemes in [26].

Theorem 1 (Order conditions for scheme in Def. 1)**.** *Define D as a diagonal matrix with $D_{ii} = d_i$; and $b^{\{k\}} \in \mathbb{R}^{1\times(s+1)}$ as the last row of $A^{\{k\}}$. $\mathbb{1} \in \mathbb{R}^{(s+1)\times 1}$ denotes a columvector filled with ones. As usual for Runge-Kutta schemes, define*

$$c_i^{\{k\}} = \sum_j A_{ij}^{\{k\}}.$$

For notational simplicity, we will set $b \equiv b^{\{0\}}$ and $c \equiv c^{\{0\}}$. Further, we define

$$\tilde{c} = \alpha c, \qquad \tilde{b} = (RD)_{(s+1),-} \in \mathbb{R}^{1\times(s+1)},$$

i.e., $\tilde{b}$ denotes the last row of the matrix RD.
Then, the scheme in Def. 1 is of order $1 \le q \le 4$, if it fulfills the following order conditions up to order q:

Order	*Algebraic condition*			*El. diff.*[1]
1	$b\mathbb{1}$	$=$	1	F
2	$bc + b^{\{1\}}\mathbb{1}$	$=$	$\frac{1}{2}$	f_yF
	$\tilde{b}(c+\tilde{c})$	$=$	1	g_yF
3	$bc^2 + 2b^{\{1\}}c + 2b^{\{2\}}\mathbb{1}$	$=$	$\frac{1}{3}$	$f_{yy}(F,F)$
	$bAc + b^{\{1\}}c + bc^{\{1\}} + b^{\{2\}}\mathbb{1}$	$=$	$\frac{1}{6}$	f_yf_yF
	$\tilde{b}(Id+\alpha)(Ac + c^{\{1\}})$	$=$	$\frac{1}{3}$	g_yf_yF
	$3\tilde{b}(\alpha+\gamma/2)RD(c+\tilde{c}) + \tilde{b}D(c+2\tilde{c})$	$=$	1	g_yg_yF
	$bRD(c+\tilde{c}) + 2b^{\{1\}}c + 2b^{\{2\}}\mathbb{1}$	$=$	$\frac{1}{3}$	f_yg_yF
	$\tilde{b}(c^2 + \tilde{c}^2 + c\cdot\tilde{c})$	$=$	1	$g_{yy}(F,F)$
4	$4bc^3 + 12b^{\{1\}}c^2 + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_{yyy}(F,F,F)$
	$12bRD(c+\tilde{c})c + 12b^{\{1\}}(2c^2 + RD(c+\tilde{c})) + 72b^{\{2\}}c + 72b^{\{3\}}\mathbb{1}$	$=$	*3*	$f_{yy}(g_yF,F)$
	$24bcAc + 24bcc^{\{1\}} + 24b^{\{1\}}Ac + 24b^{\{1\}}c^{\{1\}} + 24b^{\{1\}}c^2 + 72b^{\{2\}}c + 72b^{\{3\}}\mathbb{1}$	$=$	*3*	$f_{yy}(F,f_yF)$
	$12bAc^2 + 24bA^{\{1\}}c + 12b^{\{1\}}c^2 + 24b^{\{2\}}c + 24bc^{\{2\}} + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_yf_{yy}(F,F)$
	$24bA^2c + 24bAc^{\{1\}} + 24bA^{\{1\}}c + 24b^{\{1\}}Ac + 24b^{\{1\}}c^{\{1\}}$			
	$+ 24bc^{\{2\}} + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_yf_yf_yF$
	$12bARD(c+\tilde{c}) + 24bA^{\{1\}}c + 12b^{\{1\}}RD(c+\tilde{c}) + 24bc^{\{2\}} + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_yf_y(g_y)F$
	$4bRD(c^2 + \tilde{c}^2 + c\tilde{c}) + 12b^{\{1\}}c^2 + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_yg_{yy}(F,F)$
	$12bRD(Id+\alpha)(Ac + c^{\{1\}}) + 24b^{\{1\}}(Ac + c^{\{1\}}) + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_yg_yf_yF$
	$4bRD\left(3(\alpha+\frac{\gamma}{2})RD(c+\tilde{c}) + D(c+2\tilde{c})\right) + 12b^{\{1\}}RD(c+\tilde{c}) + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1}$	$=$	*1*	$f_yg_yg_yF$
	$\tilde{b}(c^3 + c^2\tilde{c} + c\tilde{c}^2 + \tilde{c}^3)$	$=$	*1*	$g_{yyy}(F,F,F)$
	$\tilde{b}\left((4+8\alpha)\tilde{c} + (8+4\alpha)c\right)\odot(Ac + c^{\{1\}})$	$=$	*3*	$g_{yy}(F,f_yF)$
	$6\tilde{b}(Id+\alpha)(Ac^2 + 2A^{\{1\}}c + 2c^{\{2\}})$	$=$	*1*	$g_yf_{yy}(F,F)$
	$6\tilde{b}(Id+\alpha)(2AAc + 2Ac^{\{1\}} + 2A^{\{1\}}c + 2c^{\{2\}})$	$=$	*1*	$g_yf_yf_yF$
	$\tilde{b}\left(6(Id+\alpha)(ARD(c+\tilde{c}) + 2A^{\{1\}}c + 2c^{\{2\}})\right)$	$=$	*1*	$g_yf_yg_yF$
	$\tilde{b}\left(6\alpha RD(c+\tilde{c})^2 + \gamma RD(c+\tilde{c})(4c+2\tilde{c}) + 6D\tilde{c}c + 3Dc^2 + 3D\tilde{c}^2\right)$	$=$	*3*	$g_{yy}(F,g_yF)$
	$\tilde{b}\left(4(\alpha+\frac{\gamma}{2})RD(c^2 + \tilde{c}^2 + c\tilde{c}) + 3D\tilde{c}^2 + 2D\tilde{c}c + Dc^2\right)$	$=$	*1*	$g_yg_{yy}(F,F)$
	$\tilde{b}\left(12(\alpha+\frac{\gamma}{2})RD(Id+\alpha)(Ac + c^{\{1\}}) + D(4Id+8\alpha)(Ac + c^{\{1\}})\right)$	$=$	*1*	$g_yg_yf_yF$
	$\tilde{b}\left(4(\alpha+\frac{\gamma}{2})RD(3(\alpha+\frac{\gamma}{2})RD(c+\tilde{c}) + D(c+2\tilde{c}))\right.$			

[1] Elementary differential

	$+D(6\alpha+2\gamma)RD(c+\tilde{c})+D^2(c+3\tilde{c})\Big) \;=\; 1$	$g_y g_y g_y F$

Proof. The proof of these order conditions is in big parts very similar to the proof in [7]; it can be found in the appendix. □

The order conditions presented here are rather lengthy; it is 27 conditions for order four. It has been realized in [7] that some older methods can be cast into the framework of Def. 1 with a particular definition of α and γ, that is called 'Property A' in [7] and (slightly specialized) 'MIS-KW' in [26]. This property A is also very helpful in our setting:

Theorem 2. *For meaning of the variables, consult Thm. 1. Let $\gamma = 0$, and let α be such that $\alpha_{ij} \in \{0,1\}$, and $\sum_{j=1}^{s+1}\alpha_{ij} \in \{0,1\}$, see [7, Def. 5.1]. Then, the scheme in Def. 1 is of order $1 \le q \le 4$, if it fulfills the following order conditions up to order q:*

Order	Algebraic condition	Elem. differ.
1	$b\mathbb{1} = 1$	F
2	$bc + b^{\{1\}}\mathbb{1} = \frac{1}{2}$	f_yF
3	$bc^2 + 2b^{\{1\}}c + 2b^{\{2\}}\mathbb{1} = \frac{1}{3}$	$f_{yy}(F,F)$
	$bAc + b^{\{1\}}c + bc^{\{1\}} + b^{\{2\}}\mathbb{1} = \frac{1}{6}$	f_yf_yF
	$\tilde{b}(Id+\alpha)(Ac+c^{\{1\}}) = \frac{1}{3}$	g_yf_yF
4	$4bc^3 + 12b^{\{1\}}c^2 + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1} = 1$	$f_{yyy}(F,F,F)$
	$24bcAc + 24bcc^{\{1\}} + 24b^{\{1\}}Ac + 24b^{\{1\}}c^{\{1\}} + 24b^{\{1\}}c^2 + 72b^{\{2\}}c + 72b^{\{3\}}\mathbb{1} = 3$	$f_{yy}(F,f_yF)$
	$12bAc^2 + 24bA^{\{1\}}c + 12b^{\{1\}}c^2 + 24b^{\{2\}}c + 24bc^{\{2\}} + 24b^{\{3\}}\mathbb{1} = 1$	$f_yf_{yy}(F,F)$
	$24bA^2c + 24bAc^{\{1\}} + 24bA^{\{1\}}c + 24b^{\{1\}}Ac + 24b^{\{1\}}c^{\{1\}}$ $+24b^{\{2\}}c + 24bc^{\{2\}} + 24b^{\{3\}}\mathbb{1} = 1$	$f_yf_yf_yF$
	$12bRD(Id+\alpha)(Ac+c^{\{1\}}) + 24b^{\{1\}}(Ac+c^{\{1\}}) + 24b^{\{2\}}c + 24b^{\{3\}}\mathbb{1} = 1$	$f_yg_yf_yF$
	$\tilde{b}\,((4Id+8\alpha)\tilde{c} + (8Id+4\alpha)c) \odot (Ac+c^{\{1\}}) = 3$	$g_{yy}(F,f_yF)$
	$6\tilde{b}(Id-\alpha)(Ac^2 + 2A^{\{1\}}c + 2c^{\{2\}}) = 1$	$g_yf_{yy}(F,F)$
	$6\tilde{b}\left((Id+\alpha)(2AAc + 2Ac^{\{1\}} + 2A^{\{1\}}c + 2c^{\{2\}}\right) = 1$	$g_yf_yf_yF$
	$\tilde{b}\left(12\alpha RD(I+\alpha)(Ac+c^{\{1\}}) + D(4+8\alpha)(Ac+c^{\{1\}})\right) = 1$	$g_yg_yf_yF$

Proof. The proof is very similar to the one of [7, Thm. 5.1]. In particular, one can use the same identities

$$\tilde{c}^k = \alpha c^k, \quad k \ge 1, \qquad D = C - \tilde{C},$$

where C and $\tilde{C}$ are diagonal matrices with c and $\tilde{c}$, respectively, on the diagonals. For the conditions up to order three, the procedure is the same as in [7], with the obvious incorporation of the multiderivative parts. For order four then, it is straightforward to see that the conditions corresponding to the elementary differentials $g_{yyy}(F,F,F)$, $g_yg_{yy}(F,F)$, $g_{yy}(g_yF,F)$ and $g_yg_yg_yF$ are equivalent to $c_{s+1} = 1$ which, then again, is equivalent to $b\mathbb{1} = 1$. The conditions corresponding to the elementary differentials $f_{yy}g_yF$, $f_yg_{yy}(F,F)$ and $f_yg_yg_yF$ are equivalent to the one corresponding to $f_{yyy}(F,F,F)$. The ones corresponding to $f_yf_yg_yF$ are equivalent to the ones of f_yf_{yy}; and the ones of $g_yf_yg_yF$ are equivalent to the ones of $g_yf_{yy}(F,F)$. □

Lemma 1. *Under the assumptions of Thm. 2, there is no two stage scheme ($s = 2$) of order four, even not with higher derivatives.*

Proof. The conditions corresponding to $g_{yy}(f_yF,F)$ and $g_yg_yf_yF$ cannot be fulfilled simultaneously under these assumptions, as they will lead to

$$A_{31}^{\{1\}} + A_{32}^{\{1\}} - A_{21}^{\{0\}}(A_{31}^{\{0\}} - 1) = \frac{3}{8},$$
$$A_{31}^{\{1\}} + A_{32}^{\{1\}} - A_{21}^{\{0\}}(A_{31}^{\{0\}} - 1) = \frac{1}{4}$$

which obviously cannot be fulfilled simultaneously. □

4. Stability regions

To analyze the stability of the methods, we apply, as is customary for IMEX schemes [28], our developed schemes to the model equation

$$y'(t) = i\mu y(t) + \lambda y(t),$$

where i denotes the imaginary unit and $\mu \in \mathbb{R}$, $\lambda \in \mathbb{R}$ are assumed to be constants. This equation is a prototype of a convection-diffusion equation, as the eigenvalues of a pure convection operator lie on the imaginary axes, and those of a pure diffusion operator on the (negative) real axis. In [7, Sec. 5.1], a similar problem has been considered.

The term $f(y) := i\mu y$ is assumed to be the slow part; the second term $g(y) := \lambda y$ the fast part. We follow the steps of [7] and observe that for some given constant c, the solution to $y' = c + \lambda y$ is given by

$$y(t) = e^{\lambda t} y_0 + t\phi(\lambda t)c,$$

with $\phi(z) := \frac{e^z - 1}{z}$. Applying this to (2.2), one finds that

$$\begin{aligned} Y_{ni}(\Delta t) =& e^{d_i \lambda \Delta t}\left(y_n + \sum_{j=1}^{i-1} \alpha_{ij}(Y_{nj}(\Delta t) - y_n)\right) \\ &+ \phi(d_i \lambda \Delta t)\left(\sum_{j=1}^{i-1} \gamma_{ij}\left(Y_{nj}(\Delta t) - y_n\right) + \sum_{k=0}^{m-1}\sum_{j=1}^{i-1} \Delta t^{k+1} \beta_{ij}^{\{k\}} (i\mu)^{k+1} Y_{nj}(\Delta t)\right). \end{aligned}$$

Recursively unfolding this to $Y_{n,s+1}$ yields the stability function $R(\tilde{\lambda}, \tilde{\mu})$ as a function of $\tilde{\mu} := i\Delta t\mu$ and $\tilde{\lambda} := \Delta t\lambda$.

For the schemes presented in this work, we have plotted the stability regions in Fig. 1. The first to notice is that the stability regions encorporate the whole negative real axis. This means that the splitting method is consistent so that if one solved exactly the fast part, it does not give rise to any stability problems. The second observation is that the stability region is smaller for higher-order methods. Lastly, for the same order of accuracy, the stability region is larger for schemes with more stages. This is because having more stages provides more degrees of freedom to select the coefficients, potentially leading to a larger stability region.

Let us note that all the developments in this section are based on the assumption that the fast part is solved exactly or with high order temporal schems combined with a really small time step. In practice, this is not always possible. In this case, the stability regions will also depend on the method used to solved the stiff terms.

5. Numerical results

Please note that in principle, multirate schemes are agnostic to the fine solver. This is also the case for this work here. For the numerical results, the fine solver is either a standard RK4 with timestepsize $\Delta\tau = \frac{\Delta t}{M}$ for some given M; or it is exactly integrated through a highly resolved numerical computation.

5.1. A singularly perturbed ODE: van der Pol equation

In this section, we discuss numerical results for the van der Pol equation

$$y_1'(t) = y_2(t), \quad y_2'(t) = \frac{1}{\varepsilon}\left((1 - y_1(t)^2)y_2(t) - y_1(t)\right), \tag{5.1}$$

$$y_1(0) = 2, \qquad y_2(0) = -\frac{2}{3} + \frac{10}{81}\varepsilon - \frac{292}{2187}\varepsilon^2. \tag{5.2}$$

Van der Pol's equation constitutes a singularly perturbed problem as $\varepsilon \to 0$, making it particularly challenging for a numerical solver to accurately resolve the solution. For $\varepsilon \to 0$, stiffness increases. The initial conditions are chosen in such a way that the solution is asymptotically smooth as $\varepsilon \to 0$ and final time T_{end} small enough, so we do not have to deal with any sharp gradients that spoil the numerical solution [29]. We integrate until time $T_{\text{end}} = 0.5$, which is

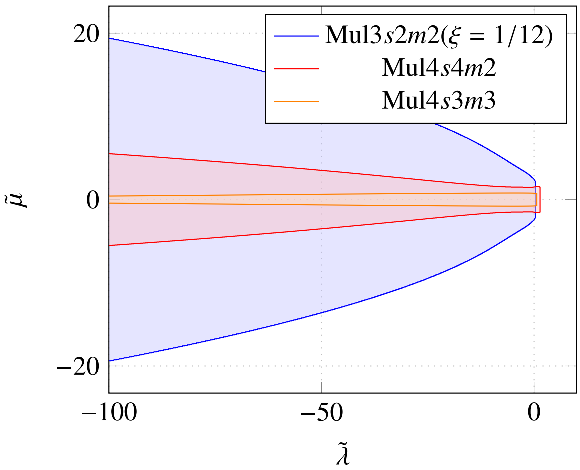

Figure 1. Stability domains for the different methods presented in this work, i.e., parts of the domain where there holds $|R(\tilde{\lambda}, \tilde{\mu})| \leq 1$. For this picture, we treat the part λy as the 'fast' (i.e., stiff) part. It can be seen very well that all the schemes encompass the negative real axis, which is natural, as the λy part is treated exactly.

small enough for having solutions without sharp gradients. The numerical error is then defined as the Eulerian norm of the difference of exact and numerical solution at time T_{end}. As is frequently done in the context of IMEX schemes, see, e.g., [30], we set non-stiff and stiff parts as

$$f(y) = \begin{pmatrix} y_2(t) \\ 0 \end{pmatrix}, \quad g(y) = \frac{1}{\varepsilon}\begin{pmatrix} 0 \\ (1 - y_1(t)^2)y_2(t) - y_1(t) \end{pmatrix}.$$

In Fig. 2, we report on numerical results using the third-order scheme (2.6) with $\xi = \frac{1}{12}$ (Mul3s2m2(1/12) for various ε. First (top picture left), we assess the quality of the outer iteration, and we set the resolution of the pseudo-time τ in (2.2) to be an exact solution (as there is no exact analytical solution to the van der Pol equation, a very highly resolved numerical computation is used). For the other three pictures, we use the classical Runge-Kutta 4 (RK4) scheme for the inner iteration using a $\Delta\tau = \frac{\Delta t}{M}$ with $M = 4$ (top right), $M = 10$ (bottom left) and $M = \frac{1}{\varepsilon}$ (bottom right). Values not plotted are NaN, hence, the method was not stable in this case. For the case with an exact fast solver, it is clearly visible that the method converges with order three for larger values of ε ('non-stiff case'), which is to be expected based on the analysis from Sec. 3. For the smaller ε, order reduction appears due to the stiff nature of the problem. Furthermore, for increasing stiffness, more iterations on the fast solver need to be done for the method to be stable. Obviously, this is not unexpected for a fully explicit scheme. In particular, only for $\Delta\tau = \varepsilon\Delta t$, so a highly-resolved fast solver, there is no convergence issue at all. Also $\Delta\tau = \sqrt{\varepsilon}\Delta t$ has been checked, this did not lead to a uniformely stable scheme throughout the ε-values considered. The message at this point is pretty clear, it is that the multirate scheme, in combination with a fully explicit interior scheme, is only useful for moderately stiff problems, or in combination with an *implicit* fast solution process.

From now on, we will hence only consider moderately stiff problems, i.e., we make the somewhat arbitrary choice of $10^{-2} \leq \varepsilon \leq 1$. In Fig. 3 on the left, we compute solutions to van der Pol equation (5.1) using the fourth-order two-derivative scheme Mul4s4m2 given in (2.8). The pseudo-time τ is integrated using an RK4-scheme with $\Delta\tau = \frac{\Delta t}{10}$. For ε considered here, this is enough to guarantee stability for all values of Δt considered. It is visible from the numerical results that the error converges with the design order of the scheme, which is four in this case. Fig. 3 on the right shows numerical results for the fourth-order three-derivative scheme Mul4s3m3, see (2.9). For large Δt and $\varepsilon = 10^{-2}$, the scheme has stability issues. This is not surprising given the size of the stability region shown in Sec. 4.

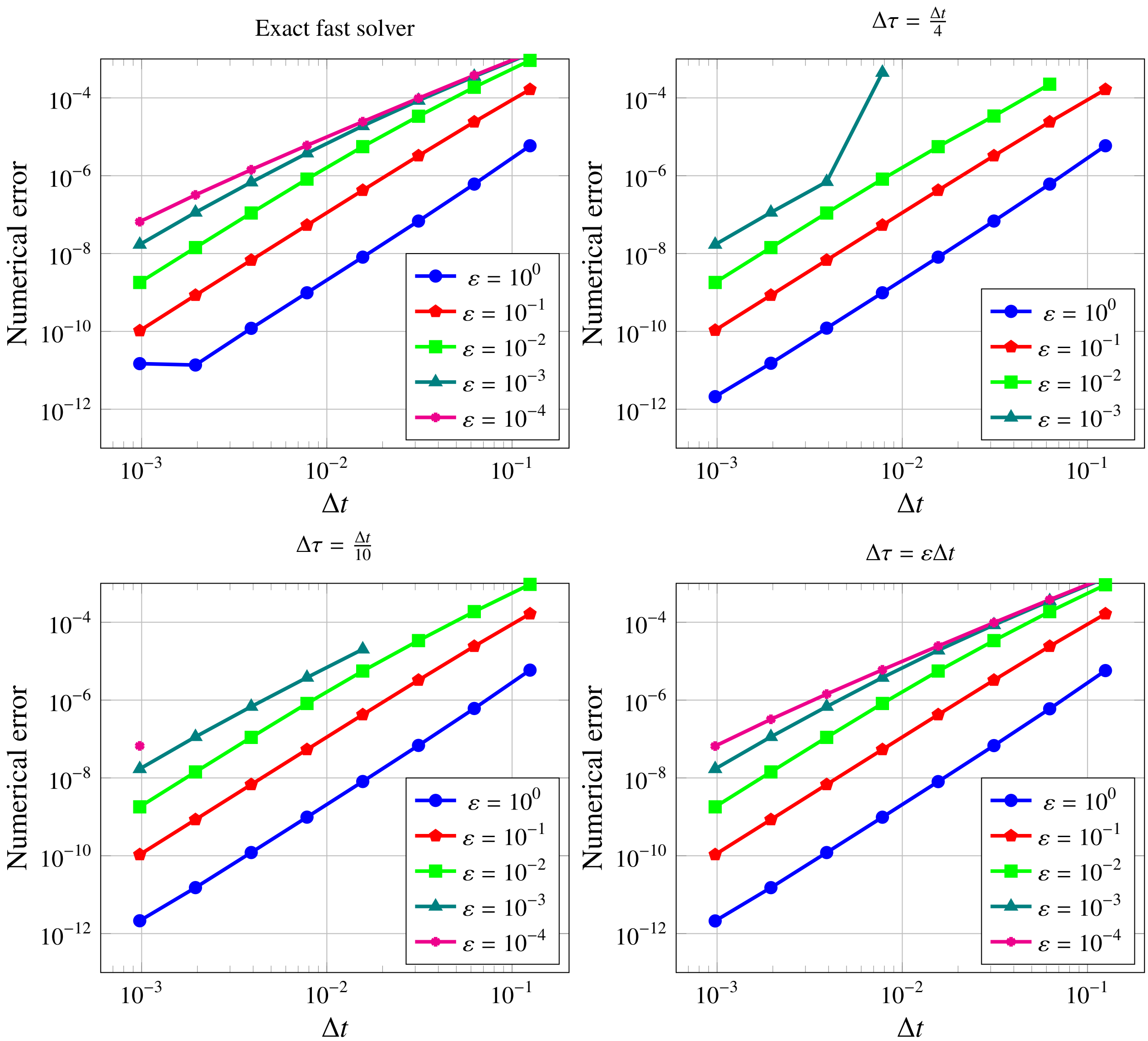

Figure 2. Numerical results for the van der Pol equation (5.1), generated using the third-order two-derivative multirate scheme (2.8) with a value of $\xi = \frac{1}{12}$. Integration of the pseudo-time τ is done using an RK4 scheme with $\Delta\tau = \frac{\Delta t}{M}$ with $M = 4$ (top right), $M = 10$ (bottom left) and $M = \frac{1}{\varepsilon}$ (bottom right). Please note that values not plotted were NaN, i.e., the overall method was not stable there.

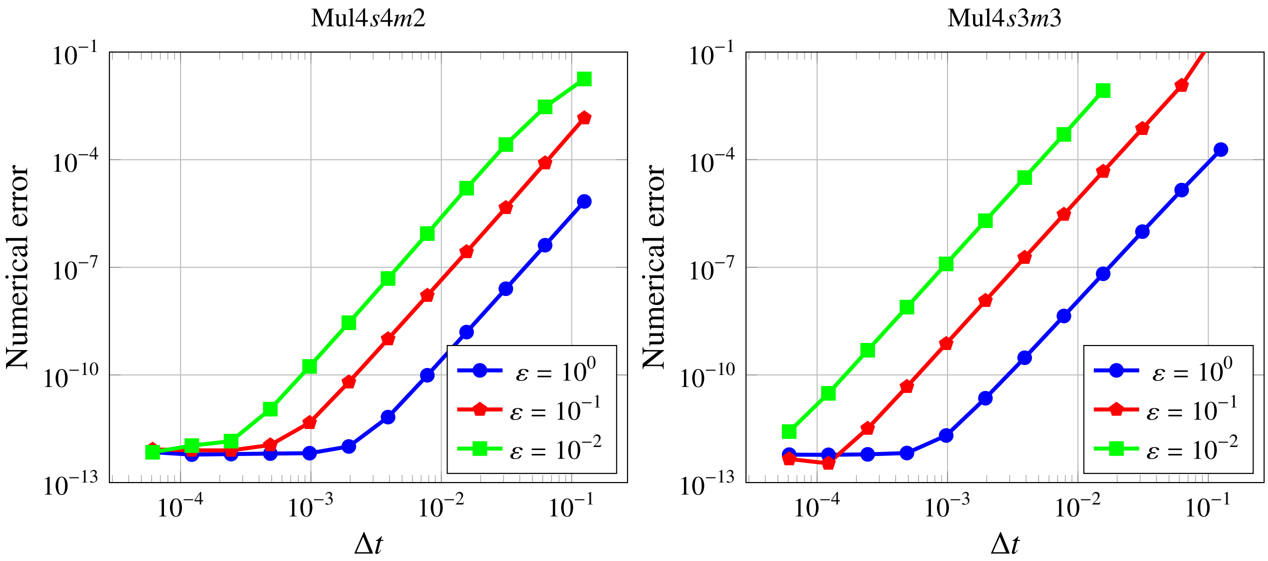

Figure 3. Numerical results for the van der Pol equation (5.1), generated using the fourth-order two-derivative multirate scheme (2.8) (left) and the fourth-order three derivative multirate scheme (2.9) (right). Integration of the pseudo-time τ is done using an RK4 scheme with $\Delta\tau = \frac{\Delta t}{10}$. Please note that values not plotted were NaN, i.e., the method was not stable there.

5.2. *Stiff hyperbolic equation*

To examine the properties of the method more closely, we consider in this section the following prototypical stiff hyperbolic equation, taken from [31]:

$$w_t + Aw_x = 0, \qquad (x,t) \in [0, 2\pi] \times [0, T_{end}] \tag{5.3}$$

with

$$A = \begin{pmatrix} a & 1 & 0 \\ \frac{1}{\varepsilon^2} & a & \frac{1}{\varepsilon^2} \\ 0 & 1 & a \end{pmatrix}$$

for some given parameter $a > 0$ (that we choose $a = 1$ in the numerical experiments). T_{end} is set to 0.5. Boundary conditions are assumed to be periodic; as initial conditions, we choose

$$w(x,0) = \begin{pmatrix} e^{-\sin(x)} \\ e^{-\sin(x)^2} \\ \cos(x) \end{pmatrix}.$$

This hyperbolic system has the three wave speeds a and $a \pm \frac{\sqrt{2}}{\varepsilon}$. For $\varepsilon \ll 1$, there is a slow wave speed (mimicking the 'convective' wave speed of the Euler equations) and two fast wave speeds (the equivalent for Euler equations would be the 'acoustic' wave speeds). Using an explicit time integration scheme on this equation, such as, e.g., explicit Euler, would result in the timestep restriction

$$\Delta t \lesssim \frac{\Delta x}{a + \frac{\sqrt{2}}{\varepsilon}} = O(\varepsilon \Delta x).$$

As in [31, Sec. 4], we split the matrix A into $A = \widehat{A} + \widetilde{A}$, with

$$\widehat{A} = \begin{pmatrix} a & \varepsilon & 0 \\ \frac{1}{\varepsilon} & a & \frac{1}{\varepsilon} \\ 0 & \varepsilon & a \end{pmatrix}, \qquad \widetilde{A} = \begin{pmatrix} 0 & 1-\varepsilon & 0 \\ \frac{1-\varepsilon}{\varepsilon^2} & 0 & \frac{1-\varepsilon}{\varepsilon^2} \\ 0 & 1-\varepsilon & 0 \end{pmatrix}.$$

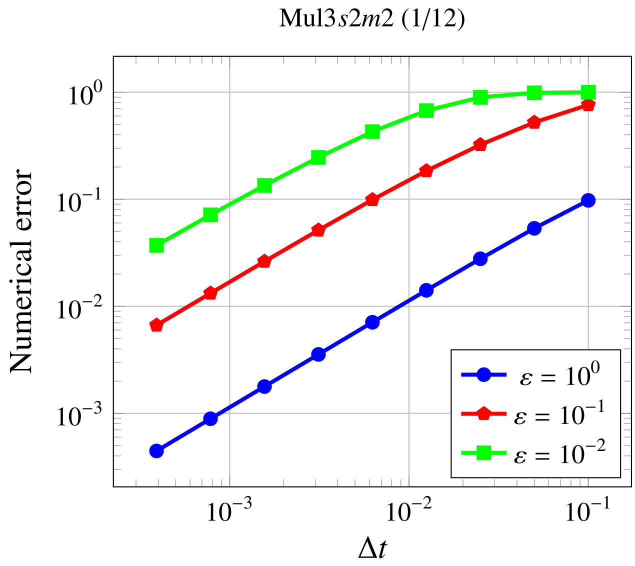

Figure 4. Numerical results to the hyperbolic problem (5.3) using a first-order Finite Volume scheme with a local Lax-Friedrichs/Rusanov flux, hence the first-order convergence. As time integration, Mul3s2m2 is chosen. We use $M = 1$, $M = 5$ and $M = 56$ for $\varepsilon = 1$, $\varepsilon = 0.1$ and $\varepsilon = 0.01$, respectively. The errors are scaled by the norm of the solution, which is ε−dependent.

The wavespeeds of $\widehat{A}$ are a and $a \pm \sqrt{2}$; the wavespeeds of $\widetilde{A}$ are 0 and $\pm\frac{\sqrt{2}(1-\varepsilon)}{\varepsilon}$. In the following, the contribution $\widehat{A}w_x$ is treated as the non-stiff part, and $\widetilde{A}w_x$ as the stiff part. We discretize the equation in space through a standard first-order Finite Volume method with a local Lax-Friedrichs / Rusanov numerical flux.

In Fig. 4, we plot numerical results for the problem integrated with the third-order Mul3s2m2(1/12) scheme. As basis for our investigations, we use the *non-stiff* CFL condition

$$\Delta t \lesssim \frac{\Delta x}{a + \sqrt{2}}.$$

This leads to an initial setup of 20 spatial cells and 5 time steps; which is then multiplied by two in each subsequent iteration. To choose $\Delta\tau$, we need the *stiff* CFL condition

$$\Delta\tau \lesssim \Delta x \frac{\varepsilon}{\sqrt{2}(1-\varepsilon)}.$$

This leads to $M = 1$, $M = 5$ and $M = 56$ for $\varepsilon = 1$, $\varepsilon = 0.1$ and $\varepsilon = 0.01$, respectively. $\Delta\tau$ is then defined as $\Delta\tau = \frac{\Delta t}{M}$. It can be clearly seen from the numerical results that the resulting algorithm is stable.

6. Conclusion and outlook

In this paper, we focused on ordinary differential equations containing both stiff and non-stiff terms. These equations are challenging because they typically require a very small time step for the entire equation, even though only some terms necessitate it. To address this issue, we presented a multiderivative scheme that incorporates multiple derivatives of the stiff part. The idea behind this scheme is to solve the non-stiff terms with a large time step while solving the stiff part exactly or with a very small time step. This approach is particularly relevant when the term with a small time characteristic is less computationally intensive than the other terms.

This class of temporal schemes has been previously studied, but only for single derivatives. We proposed schemes that include up to three derivatives of the non-stiff term and derived the order conditions to develop temporal methods up to order four. We presented three schemes that vary in the number of stages and derivatives. The stability regions were demonstrated using a classical convection-diffusion problem, where the eigenvalues of the stiff terms are purely real, while others are imaginary. Additionally, these schemes were tested on two benchmarks, confirming their order and precision.

Future work is on extending the framework to more realistic settings, e.g., for the compressible Navier-Stokes equations at low Mach numbers, or the shallow water equations at low Froude numbers. In particular, it seems challenging to choose fast-scale solvers that are efficient and keep the stability region as close to the 'exact' stability region as possible.

Acknowledgements

Arjun Thenery Manikantan was funded by the "Bijzonder Onderzoeksfonds" (BOF) from UHasselt - project no. BOF21KP12. During the preparation of this work, A. Ishimwe used Deepl in order to improve some few sentences w.r.t. language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] G. Söderlind, L. Jay, M. Calvo, Stiffness 1952–2012: sixty years in search of a definition, BIT. Numerical Mathematics 55 (2) (2015) 531–558. doi:10.1007/s10543-014-0503-3.

[2] S. Klainerman, A. Majda, Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids, Communications on Pure and Applied Mathematics 34 (1981) 481–524.

[3] S. Jin, Asymptotic preserving (AP) schemes for multiscale kinetic and hyperbolic equations: A review, Rivista di Matematica della Universita Parma 3 (2012) 177–216.

[4] D. R. Durran, P. N. Blossey, Implicit–explicit multistep methods for fast-wave–slow-wave problems, Monthly Weather Review 140 (4) (2012) 1307 – 1325. doi:10.1175/MWR-D-11-00088.1.

[5] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, K. K. Droegemeier, Numerical solutions of a non-linear density current: A benchmark solution and comparisons, International Journal for Numerical Methods in Fluids 17 (1) (1993) 1–22. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650170103, doi:https://doi.org/10.1002/fld.1650170103.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650170103`

[6] A. Ishimwe, E. Deleersnijder, V. Legat, J. Lambrechts, A split-explicit second order Runge–Kutta method for solving 3D hydrodynamic equations, Ocean Modeling (186) (2023) 102273.

[7] J. Wensch, O. Knoth, A. Galant, Multirate infinitesimal step methods for atmospheric flow simulation, BIT Numerical Mathematics 49 (2009) 449–473.

[8] M. Günther, A. Sandu, Multirate generalized additive Runge Kutta methods, Numerische Mathematik 133 (2016) 497–524.

[9] V. T. Luan, R. Chinomona, D. R. Reynolds, A new class of high-order methods for multirate differential equations, SIAM Journal on Scientific Computing 42 (2) (2020) A1245–A1268. doi:10.1137/19M125621X.

[10] A. Naumann, J. Wensch, Multirate finite step methods, Numerical Algorithms 81 (4) (2019) 1547–1571. doi:10.1007/s11075-019-00763-1.

[11] A. Sandu, A class of multirate infinitesimal GARK methods, SIAM Journal on Numerical Analysis 57 (5) (2019) 2300–2327. doi:10.1137/18M1205492.

[12] R. Chinomona, D. R. Reynolds, Implicit-explicit multirate infinitesimal GARK methods, SIAM Journal on Scientific Computing 43 (5) (2021) A3082–A3113. doi:10.1137/20M1354349.

[13] A. Abdulle, M. J. Grote, G. Rosilho de Souza, Explicit stabilized multirate method for stiff differential equations, Mathematics of Computation 91 (338) (2022) 2681–2714. doi:10.1090/mcom/3753.

[14] A. C. Fish, D. R. Reynolds, Adaptive time step control for multirate infinitesimal methods, SIAM Journal on Scientific Computing 45 (2) (2023) A958–A984. doi:10.1137/22M1479798.

[15] A. C. Fish, D. R. Reynolds, S. B. Roberts, Implicit-explicit multirate infinitesimal stage-restart methods, Journal of Computational and Applied Mathematics 438 (2024) Paper No. 115534, 23. doi:10.1016/j.cam.2023.115534.

[16] S. Kang, A. Dener, A. Hamilton, H. Zhang, E. M. Constantinescu, R. L. Jacob, Multirate partitioned Runge-Kutta methods for coupled Navier-Stokes equations, Computers & Fluids. An International Journal 264 (2023) Paper No. 105964, 15. doi:10.1016/j.compfluid.2023.105964.

[17] E. Hairer, G. Wanner, Multistep-multistage-multiderivative methods for ordinary differential equations, Computing (Arch. Elektron. Rechnen) 11 (3) (1973) 287–303.

[18] J. C. Butcher, G. Hojjati, Second derivative methods with RK stability, Numerical Algorithms 40 (4) (2005) 415–429.

[19] R. Chan, A. Tsai, On explicit two-derivative Runge-Kutta methods, Numerical Algorithms 53 (2010) 171–194.

[20] D. C. Seal, Y. Güçlü, A. Christlieb, High-order multiderivative time integrators for hyperbolic conservation laws, Journal of Scientific Computing 60 (2014) 101–140.

[21] J. Zeifang, J. Schütz, Implicit two-derivative deferred correction time discretization for the discontinuous Galerkin method, Journal of Computational Physics 464 (2022) 111353.

[22] A. Baeza, S. Boscarino, P. Mulet, G. Russo, D. Zorío, On the stability of Approximate Taylor methods for ODE and their relationship with Runge-Kutta schemes, arXiv preprint arXiv:1804.03627 (2018). arXiv:1804.03627.

[23] H. Carrillo, C. Parés, Compact approximate Taylor methods for systems of conservation laws, Journal of Scientific Computing 80 (3) (2019) 1832–1866.

[24] D. Zorío, A. Baeza, P. Mulet, An approximate Lax–Wendroff-type procedure for high order accurate schemes for hyperbolic conservation laws, Journal of Scientific Computing 71 (2017) 246–273.
[25] J. Chouchoulis, J. Schütz, J. Zeifang, Jacobian-free explicit multiderivative Runge-Kutta methods for hyperbolic conservation laws, Journal of Scientific Computing 90 (96) (2022).
[26] T. P. Bauer, O. Knoth, Extended multirate infinitesimal step methods: Derivation of order conditions, Journal of Computational and Applied Mathematics 387 (2021) 112541.
[27] M. Ökten Turaci, T. Öziş, Derivation of three-derivative Runge-Kutta methods, Numerical Algorithms 74 (1) (2017) 247–265.
[28] U. M. Ascher, S. Ruuth, R. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Applied Numerical Mathematics 25 (1997) 151–167.
[29] E. Hairer, G. Wanner, Solving ordinary differential equations II, Springer Series in Computational Mathematics, 1991.
[30] S. Boscarino, On an accurate third order implicit-explicit Runge-Kutta method for stiff problems, Applied Numerical Mathematics 59 (2009) 1515–1528.
[31] J. Schütz, S. Noelle, Flux splitting for stiff equations: A notion on stability, Journal of Scientific Computing 64 (2) (2015) 522–540.

7. Appendix: Proof of the order conditions

In this appendix section, we prove Thm. 1. The proof goes very much along the lines of [7, Sec. 3], most of equations are very similar. However, some of the details vary, which is why we provide it here. The notation used is the notation from [7]. We restrict ourselves to four derivatives at most, i.e., $m \leq 4$. The function Z_{ni} from (2.2) is a function of τ and Δt, obviously. As in [7], we define $Z^{(k,l)}$ to be the k−th derivative of Z in direction τ, and the l−th derivative of Z in direction of Δt. $Y^{(l)}$ denotes the l−th derivative of Y in direction Δt. Evaluation is in both cases at $\tau = \Delta t = 0$. By $(f^{\{k\}})^l$, we denote the l−th derivative of $f^{\{k\}}(Y)$ in direction Δt. The understanding is that $(f^{\{k\}})^l$ is zero if $l < 0$. If $l = 0$, we assume $(f^{\{k\}})^0 = f^{\{k\}}$; consult (2.3) for the definition.

With these defintions, there holds

$$\begin{aligned}
Z_{ni}^{(0,l)} &= \sum_j \alpha_{ij} Y_{nj}^{(l)}, \qquad l \geq 1 \\
Z_{ni}^{(1,l)} &= \frac{1}{l+1} \sum_j \gamma_{ij} Y_{nj}^{(l+1)} + d_i g(Z_{ni})^{(0,l)} + \sum_{k=0}^{m-1} \frac{l!}{(l-k)!} \sum_{j=1}^{i-1} \beta_{ij}^{\{k\}} f^{\{k\}}(Y_{nj})^{(l-k)} \\
Z_{ni}^{(k,l)} &= d_i g(Z_{ni})^{(k-1,l)}, \qquad k \geq 2
\end{aligned}$$

In the matrix notation of [7], this amounts to

$$\begin{aligned}
Z^{(0,l)} &= \alpha Y^{(l)}, \qquad l \geq 1 \\
Z^{(1,l)} &= \frac{1}{l+1} (\gamma \otimes I) Y^{(l+1)} + (D \otimes I) g(Z)^{(0,l)} + \sum_{k=0}^{m-1} \frac{l!}{(l-k)!} (\beta^{\{k\}} \otimes I) f^{\{k\}}(Y)^{(l-k)} \\
Z^{(k,l)} &= (D \otimes I) g(Z)^{(k-1,l)}, \qquad k \geq 2
\end{aligned}$$

Here, Y denotes the upright vector $Y = (Y_{n1}, \ldots, Y_{n,s+1})^T$, and similarly for Z. We understand $f^{\{k\}}(Y)$ and similar expressions as the vector $(f^{\{k\}}(Y_{n1}), \ldots, f^{\{k\}}(Y_{n,s+1}))^T$. Using the chain rule, we obtain that for $Y^{(\kappa)}$, there holds

$$\begin{aligned}
Y^{(\kappa)} &= \sum_{l=0}^{\kappa} \binom{\kappa}{l} Z^{(l,\kappa-l)} \\
&= \sum_{l=1}^{\kappa} \binom{\kappa}{l} (D \otimes I) g(Z)^{(l-1,\kappa-l)} + ((\gamma + \alpha) \otimes I) Y^{(\kappa)} + \sum_{k=0}^{m-1} \frac{\kappa!}{(\kappa-k-1)!} (\beta^{\{k\}} \otimes I) f^{\{k\}}(Y)^{\kappa-k-1}.
\end{aligned}$$

Upon using the definitions of R and $A^{\{k\}}$ as in (2.4) and (2.5), we obtain

$$Y^{(\kappa)} = \sum_{l=1}^{\kappa} \binom{\kappa}{l} (RD \otimes I) g(Z)^{(l-1,\kappa-l)} + \sum_{k=0}^{m-1} \frac{\kappa!}{(\kappa-k-1)!} (A^{\{k\}} \otimes I) f^{\{k\}}(Y)^{\kappa-k-1}.$$

In the following, we define

$$c^{\{k\}} := A^{\{k\}}\mathbb{1}.$$

For $k = 0$, this is the usual definition of the time instances of the stages for Runge-Kutta schemes. For easier notation, we write $c \equiv c^{\{0\}}$. We now go through the terms recursively as in [7].

Zeroth and first order. To zeroth and first order, there holds

$$\begin{aligned}
Y^{(0)} &= Z^{(0,0)} = \mathbb{1} \otimes y_n, \\
Y^{(1)} &= (RD \otimes I)g(Z)^{(0,0)} + (A^{\{0\}} \otimes I)f^{\{0\}}(Y)^{(0)} = (RD\mathbb{1}) \otimes g + (A^{\{0\}}\mathbb{1}) \otimes f^{\{0\}} \\
&= c \otimes F.
\end{aligned}$$

The last step is true because $D\mathbb{1} = \beta^{\{0\}}\mathbb{1}$ due to the balancing condition.

Second order. Further, there holds

$$\begin{aligned}
Z^{(0,1)} &= \alpha Y^{(1)} = (\alpha c \otimes I)F =: \tilde{c} \otimes F, \\
Z^{(1,0)} &= (\gamma \otimes I)Y^{(1)} + (\beta^{\{0\}} \otimes I)f^{\{0\}}(Y)^{(0)} + (D \otimes I)g(Z)^{(0,0)} \\
&= (\gamma c \otimes I)F + (\beta^{\{0\}}\mathbb{1} \otimes I)f^{\{0\}} + (D\mathbb{1} \otimes I)g \\
&= ((\gamma c + \beta^{\{0\}}\mathbb{1}) \otimes I)F.
\end{aligned}$$

Now we have

$$\beta\mathbb{1} = R^{-1}A^{\{0\}}\mathbb{1} = R^{-1}c,$$

and hence

$$Z^{(1,0)} = ((\gamma + R^{-1})c \otimes I)F = ((I - \alpha)c) \otimes I)F = (c - \tilde{c}) \otimes F.$$

Now, to second order:

$$\begin{aligned}
Y^{(2)} &= \sum_{l=1}^{2} \binom{2}{l} (RD \otimes I)g(Z)^{(l-1,2-l)} + 2(A^{\{0\}} \otimes I)f^{\{0\}}(Y)^{(1)} + 2(A^{\{1\}} \otimes I)f^{\{1\}}(Y)^{(0)} \\
&= 2(RD \otimes I)g(Z)^{(0,1)} + (RD \otimes I)g(Z)^{(1,0)} + 2(A^{\{0\}} \otimes I)f_y Y^{(1)} + 2(A^{\{1\}} \otimes I)f^{\{1\}}(Y)^{(0)} \\
&= 2(RD \otimes I)g_y Z^{(0,1)} + (RD \otimes I)g_y Z^{(1,0)} + 2A^{\{0\}}c \otimes f_y F + 2A^{\{1\}}\mathbb{1} \otimes f_y F \\
&= 2RD\tilde{c} \otimes g_y F + RD(c - \tilde{c}) \otimes g_y F + 2A^{\{0\}}c \otimes f_y F + 2c^{\{1\}} \otimes f_y F \\
&= RD(c + \tilde{c}) \otimes g_y F + (2A^{\{0\}}c + 2c^{\{1\}}) \otimes f_y F.
\end{aligned}$$

Third order. Continuing with higher-order terms, we obtain

$$\begin{aligned}
Z^{(0,2)} &= \alpha Y^{(2)} = \alpha RD(c + \tilde{c}) \otimes g_y F + (2\alpha A^{\{0\}}c + 2\alpha c^{\{1\}}) \otimes f_y F \\
Z^{(1,1)} &= \frac{1}{2}(\gamma \otimes I)Y^{(2)} + (\beta^{\{0\}} \otimes I)f^{\{0\}}(Y)^{(1)} + (\beta^{\{1\}} \otimes I)f^{\{1\}}(Y)^{(0)} + (D \otimes I)g(Z)^{(0,1)} \\
&= \frac{1}{2}(\gamma \otimes I)Y^{(2)} + (\beta^{\{0\}} \otimes I)f_y Y^{(1)} + (\beta^{\{1\}}\mathbb{1}) \otimes f_y F + (D \otimes I)g_y Z^{(0,1)} \\
&= \frac{1}{2}(\gamma \otimes I)Y^{(2)} + \beta c \otimes f_y F + (R^{-1}c^{\{1\}}) \otimes f_y F + (D\tilde{c}) \otimes g_y F. \\
Z^{(2,0)} &= (D \otimes I)g(Z)^{(1,0)} = (D \otimes I)g_y Z^{(1,0)} = D(c - \tilde{c}) \otimes g_y F.
\end{aligned}$$

Now finally, we end up with the third derivative of Y:

$$\begin{aligned} Y^{(3)} &= \sum_{l=1}^{3} \binom{3}{l} (RD \otimes I) g(Z)^{(l-1,3-l)} + 3(A^{\{0\}} \otimes I) f^{\{0\}}(Y)^{(2)} + 6(A^{\{1\}} \otimes I) f^{\{1\}}(Y)^{(1)} + 6(A^{\{2\}} \otimes I) f^{\{2\}}(Y)^{(0)} \\ &= (RD \otimes I) \sum_{l=1}^{3} \binom{3}{l} g(Z)^{(l-1,3-l)} + 3(A^{\{0\}} \otimes I) f(Y)^{(2)} + 6(A^{\{1\}} \otimes I) f^{\{1\}}(Y)^{(1)} + 6(A^{\{2\}} \otimes I) f^{\{2\}}(Y)^{(0)}. \end{aligned} \tag{7.1}$$

First, the sum is treated. If one realizes that $\beta = R^{-1}A$ and $R^{-1} = I - \alpha - \gamma$, then in a straightforward way, one obtains

$$\begin{aligned} \sum_{l=1}^{3} \binom{3}{l} g(Z)^{(l-1,3-l)} = &\left(c^2 + \tilde{c}^2 + c \odot \tilde{c}\right) \otimes g_{yy}(F, F) \\ &+ \left(3(I + \alpha)(A^{\{0\}} c + c^{\{1\}})\right) g_y f_y F \\ &+ \left(3(\alpha + \frac{\gamma}{2}) RD(c + \tilde{c}) + D(c + 2\tilde{c})\right) g_y g_y F. \end{aligned}$$

Now we treat the remainder of (7.1). Before doing this, we make the following computation about the Jacobian of the first total derivative:

$$f_y^{\{1\}} = (f_y F)_y = (f_y (f + g))_y = f_y f_y + f_y g_y + f_{yy} F,$$

Furthermore, there holds

$$f^{\{2\}} = f_{yy}(F, F) + f_y F_y F.$$

Hence,

$$\begin{aligned} &3(A^{\{0\}} \otimes I) f^{\{0\}}(Y)^{(2)} + 6(A^{\{1\}} \otimes I) f^{\{1\}}(Y)^{(1)} + 6(A^{\{2\}} \otimes I) f^{\{2\}}(Y)^{(0)} \\ &= 3(A^{\{0\}} \otimes I) \left(f_{yy}(Y^{(1)}, Y^{(1)}) + f_y Y^{(2)}\right) + 6(A^{\{1\}} \otimes I) f_y^{\{1\}} Y^{(1)} + 6(A^{\{2\}} \otimes I) f^{\{2\}}(Y)^{(0)} \\ &= 3(A^{\{0\}} c^2) \otimes f_{yy}(F, F) + 3(A^{\{0\}} \otimes I) f_y Y^{(2)} + 6(A^{\{1\}} c) \otimes (f_y f_y F + f_y g_y F + f_{yy}(F, F)) \\ &\quad + 6A^{\{2\}} \mathbb{1} \otimes \left(f_{yy}(F, F) + f_y F_y F\right). \end{aligned}$$

Now, we can collect all terms in $Y^{(3)}$ to conclude:

$$\begin{aligned} Y^{(3)} = &RD \left(c^2 + \tilde{c}^2 + c \odot \tilde{c}\right) \otimes g_{yy}(F, F) \\ &+ RD \left(3(I + \alpha)(A^{\{0\}} c + c^{\{1\}})\right) \otimes g_y f_y F \\ &+ RD \left(3(\alpha + \frac{\gamma}{2}) RD(c + \tilde{c}) + D(c + 2\tilde{c})\right) \otimes g_y g_y F \\ &+ 3(A^{\{0\}} c^2 + 2A^{\{1\}} c + 2A^{\{2\}} \mathbb{1}) \otimes f_{yy}(F, F) \\ &+ 3 \left(A^{\{0\}} RD(c + \tilde{c}) + 2A^{\{1\}} c + 2A^{\{2\}} \mathbb{1}\right) \otimes f_y g_y F \\ &+ 3 \left(2A^{\{0\}} A^{\{0\}} c + 2A^{\{0\}} c^{\{1\}} + 2A^{\{1\}} c + 2A^{\{2\}} \mathbb{1}\right) \otimes f_y f_y F. \end{aligned}$$

Fourth order. We conclude this investigation with the conditions for fourth order. The computations are tedious but straightforward. As we are not aiming for fifth order, we do not go into all the details here.

$$\begin{aligned} Z^{(0,3)} &= \alpha Y^{(3)} \\ Z^{(1,2)} &= \frac{1}{3}(\gamma \otimes I) Y^{(3)} + (D \otimes I) g(Z)^{(0,2)} + (\beta^{\{0\}} \otimes I) f^{\{0\}}(Y)^{(2)} + 2(\beta^{\{1\}} \otimes I) f^{\{1\}}(Y)^{(1)} + 2(\beta^{\{2\}} \otimes I) f^{\{2\}}(Y)^{(0)} \\ &= \frac{1}{3}(\gamma \otimes I) Y^{(3)} + (D\tilde{c}^2 \otimes I) g_{yy}(F, F) + (D \otimes I) g_y Z^{(0,2)} + (\beta^{\{0\}} c^2 \otimes I) f_{yy}(F, F) + (\beta^{\{0\}} \otimes I) f_y Y^{(2)} \end{aligned}$$

$$
\begin{aligned}
&+ 2(\beta^{\{1\}} \otimes I) f_y^{\{1\}} Y^{(1)} + 2(\beta^{\{2\}} \otimes I) f^{\{2\}}(Y) \\
Z^{(2,1)} &= (D \otimes I) g(Z)^{(1,1)} = (D \otimes I)\left(g_{yy}(Z^{(0,1)}, Z^{(1,0)}) + g_y Z^{(1,1)} \right) \\
&= (D\tilde{c} \odot (c - \tilde{c}) \otimes I) g_{yy}(F, F) + (D \otimes I) g_y Z^{(1,1)} \\
Z^{(3,0)} &= (D \otimes I) g(Z)^{(2,0)} = (D \otimes I)\left(g_{yy}(Z^{(1,0)}, Z^{(1,0)}) + g_y Z^{(2,0)} \right) \\
&= (D(c - \tilde{c})^2 \otimes g_{yy}(F, F) + DD(c - \tilde{c}) \otimes g_y g_y F
\end{aligned}
$$

Finally, with this, one can derive the fourth-order conditions to be

$$
\begin{aligned}
Y^{(4)} = &\sum_{l=1}^{4} \binom{4}{l} (RD \otimes I) g(Z)^{(l-1, 4-l)} \\
&+ 4(A^{\{0\}} \otimes I) f^{\{0\}}(Y)^{(3)} + 12(A^{\{1\}} \otimes I) f^{\{1\}}(Y)^{(2)} + 24(A^{\{2\}} \otimes I) f^{\{2\}}(Y)^{(1)} + 24(A^{\{3\}} \otimes I) f^{\{3\}}(Y)^{(0)}.
\end{aligned}
$$

Please note that there holds

$$
\begin{aligned}
f^{\{0\}}(Y)^{(3)} &= f_{yyy}(Y^{(1)}, Y^{(1)}, Y^{(1)}) + 3 f_{yy}(Y^{(2)}, Y^{(1)}) + f_y Y^{(3)}, \\
f^{\{1\}}(Y)^{(2)} &= f_{yyy}(Y^{(1)}, Y^{(1)}, F) + f_{yy}(Y^{(2)}, F) + 2 f_{yy}(Y^{(1)}, F_y Y^{(1)}) + f_y F_{yy}(Y^{(1)}, Y^{(1)}) + f_y F_y Y^{(2)} \\
f^{\{2\}}(Y)^{(1)} &= f_{yyy}(Y^{(1)}, F, F) + 2 f_{yy}(F_y Y^{(1)}, F) + f_{yy}(Y^{(1)}, f_y F) + f_y f_{yy}(Y^{(1)}, F) + f_y f_y F_y Y^{(1)} \\
&\quad + f_{yy}(Y^{(1)}, g_y F) + f_y g_{yy}(Y^{(1)}, F) + f_y g_y F_y Y^{(1)}.
\end{aligned}
$$

Now, upon observing that

$$
\begin{aligned}
&\sum_{l=1}^{4} \binom{4}{l} (RD \otimes I) g(Z)^{(l-1, 4-l)} \\
=&4(RD \otimes I) g(Z)^{(0,3)} + 6(RD \otimes I) g(Z)^{(1,2)} + 4(RD \otimes I) g(Z)^{(2,1)} + (RD \otimes I) g(Z)^{(3,0)}
\end{aligned}
$$

and

$$
\begin{aligned}
g(Z)^{(0,3)} &= g_{yyy}(Z^{(0,1)}, Z^{(0,1)}, Z^{(0,1)}) + 3 g_{yy}(Z^{(0,2)}, Z^{(0,1)}) + g_y Z^{(0,3)} \\
g(Z)^{(1,2)} &= g_{yyy}(Z^{(1,0)}, Z^{(0,1)}, Z^{(0,1)}) + 2 g_{yy}(Z^{(0,1)}, Z^{(1,1)}) + g_{yy}(Z^{(1,0)}, Z^{(0,2)}) + g_y Z^{(1,2)} \\
g(Z)^{(2,1)} &= g_{yyy}(Z^{(1,0)}, Z^{(1,0)}, Z^{(0,1)}) + 2 g_{yy}(Z^{(1,1)}, Z^{(1,0)}) + g_{yy}(Z^{(2,0)}, Z^{(0,1)}) + g_y Z^{(2,1)} \\
g(Z)^{(3,0)} &= g_{yyy}(Z^{(1,0)}, Z^{(1,0)}, Z^{(1,0)}) + 3 g_{yy}(Z^{(2,0)}, Z^{(1,0)}) + g_y Z^{(3,0)},
\end{aligned}
$$

one can in a straightforward way obtain the order conditions as in Thm. 1.