Journal of Computational and Applied Mathematics 226 (2009) 345–357

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Multirate Runge–Kutta schemes for advection equations

Martin Schlegel [a,*], Oswald Knoth [a], Martin Arnold [b], Ralf Wolke [a]

[a] Institute for Tropospheric Research, Permoserstrasse 53, 04318 Leipzig, Germany

[b] Martin Luther University Halle/Wittenberg, Natural Science Faculty III, Institute of Mathematics, 06099 Halle (Saale), Germany

ARTICLE INFO

MSC:
65L06
34-04

Keywords:
Time integration
Runge–Kutta method
Multirate scheme
Flux splitting

ABSTRACT

Explicit time integration methods can be employed to simulate a broad spectrum of physical phenomena. The wide range of scales encountered lead to the problem that the fastest cell of the simulation dictates the global time step. Multirate time integration methods can be employed to alter the time step locally so that slower components take longer and fewer time steps, resulting in a moderate to substantial reduction of the computational cost, depending on the scenario to simulate [S. Osher, R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, Math. Comput. 41 (1983) 321–336; H. Tang, G. Warnecke, A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and pace grids, SIAM J. Sci. Comput. 26 (4) (2005) 1415–1431; E. Constantinescu, A. Sandu, Multirate timestepping methods for hyperbolic conservation laws, SIAM J. Sci. Comput. 33 (3) (2007) 239–278]. In air pollution modeling the advection part is usually integrated explicitly in time, where the time step is constrained by a locally varying Courant–Friedrichs–Lewy (CFL) number. Multirate schemes are a useful tool to decouple different physical regions so that this constraint becomes a local instead of a global restriction. Therefore it is of major interest to apply multirate schemes to the advection equation. We introduce a generic recursive multirate Runge–Kutta scheme that can be easily adapted to an arbitrary number of refinement levels. It preserves the linear invariants of the system and is of third order accuracy when applied to certain explicit Runge–Kutta methods as base method.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Modern air pollution models can be used to simulate the evolution of concentrations of contaminants in the atmosphere. The relevant processes can be described by partial differential equations (PDE) of advection-diffusion-reaction type which can be efficiently approximated by conservative high order spatial discretization methods with implicit/explicit time integration.

For the advection equation

$$\frac{\partial}{\partial t}c + \frac{\partial}{\partial x}(uc) = 0, \tag{1}$$

describing transport of contaminants in air pollution models, explicit Runge–Kutta (ERK) time integration methods have proven to be very efficient. All of these methods have in common that stability requirements limit the global time step to

* Corresponding author.
E-mail addresses: schlegel@tropos.de (M. Schlegel), knoth@tropos.de (O. Knoth), martin.arnold@numerik.uni-halle.de (M. Arnold), wolke@tropos.de (R. Wolke).

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.08.009

be smaller than some critical value proportional to the ratio of grid size and the magnitude of the wind speed for each cell [2,7,14].

In realistic scenarios there are usually regions of interest such as urban areas which have to be examined more closely than the surrounding area, e.g. woodland or agricultural regions. Therefore the spatial grid in these regions is refined [3,4]. Additionally even for equidistant grids the wind speed may vary considerably across the entire domain. Thus the smallest cell, or more exactly the cell with the smallest characteristic time determines the global time step. Multirate time integration methods can be employed to adapt the time step locally so that slower components take longer and fewer time steps, resulting in a moderate to substantial reduction of the computational cost, depending on the scenario to simulate [5].

The multirate scheme we shall introduce is intended to be implemented in a complex three-dimensional weather model. This model shall also employ local refinement strategies and must allow for strongly varying wind speeds. Cell size ratios up to 1:16 and wind speeds of some m/s at ground level in contrast to approximately 100 m/s in large altitudes may lead to characteristic times per cell varying by several orders of magnitude. Therefore the use of a multirate time integration scheme is crucial for efficient computation.

Multirate approaches have been developed since the early 1980s. Osher and Sanders [11] presented a scheme allowing multirate Euler steps. More current approaches allow for a wide variety of multirate schemes by proposing generic multirate methods based on traditional ERKs. In 2005 Tang and Warnecke [12] proposed different multirate schemes which can be generalized to support arbitrary ERKs as base method. The resulting schemes however are not mass preserving. The approach of Constantinescu and Sandu [2] yields multirate ERK schemes which are mass preserving and at most of second order accuracy.

The approach discussed in this paper borrows ideas from an implicit–explicit splitting scheme introduced in [10], where explicit Runge–Kutta methods are combined with an arbitrary implicit time integrator. In contrast to the methods mentioned above this new method is based on a right-hand side splitting and not on a splitting by components. Applied to the discretized advection equation this means that it is based on a splitting of the fluxes instead of flux differences per cell. The new method is called *Recursive Flux Splitting Multirate* (RFSMR).

The main part of this paper, Section 2, is dedicated to the introduction and order analysis of RFSMR. In particular we shall show that the scheme is second order accurate for second order base methods. Furthermore we prove that the third order methods from Knoth and Wolke [10] can be extended to a third order method in the new context. The order analysis will be carried out in the context of partitioned Runge–Kutta methods (PRK). Afterwards, in Section 3 we will present spatial discretizations of the advection equation. A decomposition approach beneficial for the new schemes will be outlined in the context of block structured grids. In Section 4 we will perform numerical tests affirming both stability and accuracy as well as efficiency of this novel class of multirate schemes.

2. Explicit Runge–Kutta multirate scheme based on a right-hand side splitting

The multirate scheme is derived from a splitting of the right-hand side of the differential equation in two parts as follows

$$w' = F(w) + G(w), \qquad w(0) = w_0, \quad w \in \mathbb{R}^N. \tag{2}$$

Starting point is an implicit–explicit (IMEX) integration method introduced in [10] for the efficient solution of advection-diffusion-reaction equations in air pollution. For air pollution models the G term then represents the non-stiff advection part which can be solved using explicit methods. Opposed to this the F term represents the stiff diffusion-reaction part which must be solved using implicit methods. In the cited IMEX scheme the non-stiff, explicit part is integrated by an explicit Runge–Kutta method, the integration method for the stiff part is undetermined. For the theoretical analysis of the method it is assumed that this integration is carried out exactly. In [10] the IMEX method applied to (2) is presented in a generalized Butcher-like tableau. For a given time step Δt the method reads as follows:

$$w_1 = w(t_n), \tag{3}$$

$$w_i = v_i(\tilde{c}_i \Delta t) \quad \text{with} \tag{4}$$

$$\frac{\mathrm{d}v_i}{\mathrm{d}\tau} = \frac{1}{\tilde{c}_i} r_i + F(v_i), \quad \tau \in [0, \tilde{c}_i \Delta t], \qquad v_i(0) = w_{i-1}, \quad i = 2, \ldots, s+1 \tag{5}$$

$$r_i = \sum_{j=1}^{i-1} \tilde{a}_{ij} G(w_j), \tag{6}$$

$$w(t_n + \Delta t) = w_{s+1}, \tag{7}$$

assuming that in each stage the underlying differential equation (5) with the non-discretized implicit term F is integrated exactly. If the implicit term $F \equiv 0$ we obtain the underlying classic explicit Runge–Kutta method which we will call the *outer* method:

$$w_1 = w(t_n) \tag{8}$$

$$w_i = w_{i-1} + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} G(w_j) \tag{9}$$

$$= \begin{cases} w_0 + \Delta t \sum_{j=1}^{i-1} a_{ij} G\left(w_j\right), & \text{if } i = 1, \ldots, s \\ w_0 + \Delta t \sum_{j=1}^{s} b_j G\left(w_j\right), & \text{if } i = s+1 \end{cases} \tag{10}$$

$$w(t_n + \Delta t) = w_{s+1}, \tag{11}$$

with the relation of the tilde coefficients to the Runge–Kutta parameters in common notation:

$$\tilde{a}_{ij} = a_{i,j} - a_{i-1,j}, \tag{12}$$

$$\tilde{a}_{s+1,j} = b_i - a_{s,j}, \tag{13}$$

$$\tilde{c}_i = c_i - c_{i-1}. \tag{14}$$

Note that the idea of the IMEX method is that the cumulative integration interval for the stiff part equals the explicit time step. There is another class of comparable IMEX schemes where the integration for the stiff part starts in each internal stage at the old time stage t_n and the initial value from the previous integration step and therefore the cumulative interval is usually larger than the explicit time step [13].

If F like G is non-stiff but imposes stricter time step restrictions we can approximate (5) by a further ERK method that we will call the *inner* method. This may also include a repeated application of an ERK with a smaller time step since this repeated application can also be represented by another ERK with more stages. The choice of the inner method depends on the stability requirements with respect to F. In addition to the subscript i, which denotes the ith stage of the outer method, a superscript k is introduced, which indicates the kth stage of the inner method computed in stage i of the outer method. For instance $v_i^{(k)}$ denotes the approximation for v at stage i of the outer method and stage k of the inner method. In the rest of this paper we will denote the parameters of the inner method with a superscript I and the parameters of the outer method with a superscript O.

Approximating (5) by an ERK we obtain

$$\begin{aligned}
v_i^{(1)} &= w_{i-1}, \\
v_i^{(k)} &= v_i^{(k-1)} + \tilde{c}_i^{\mathrm{O}} \Delta t \sum_{j=1}^{k-1} \tilde{a}_{k,j}^{\mathrm{I}} \left(\frac{1}{\tilde{c}_i^{\mathrm{O}}} r_i + F\left(v_i^{(j)}\right) \right) \\
&= v_i^{(k-1)} + \left[\tilde{c}_i^{\mathrm{O}} \Delta t \sum_{j=1}^{k-1} \tilde{a}_{k,j}^{\mathrm{I}} \frac{1}{\tilde{c}_i^{\mathrm{O}}} r_i \right] + \left[\tilde{c}_i^{\mathrm{O}} \Delta t \sum_{j=1}^{k-1} \tilde{a}_{k,j}^{\mathrm{I}} F\left(v_i^{(j)}\right) \right] \\
&= v_i^{(k-1)} + \left[\Delta t \sum_{j=1}^{k-1} \tilde{a}_{k,j}^{\mathrm{I}} \left(\sum_{l=1}^{i-1} \tilde{a}_{i,l}^{\mathrm{O}} G(w_l) \right) \right] + \left[\tilde{c}_i^{\mathrm{O}} \Delta t \sum_{j=1}^{k-1} \tilde{a}_{k,j}^{\mathrm{I}} F\left(v_i^{(j)}\right) \right] \\
&= v_i^{(k-1)} + \left[\Delta t \tilde{c}_k^{\mathrm{I}} \sum_{j=1}^{i-1} \tilde{a}_{i,j}^{\mathrm{O}} G(w_j) \right] + \left[\tilde{c}_i^{\mathrm{O}} \Delta t \sum_{j=1}^{k-1} \tilde{a}_{k,j}^{\mathrm{I}} F\left(v_i^{(j)}\right) \right], \\
k &= 2, \ldots, q+1, \\
w_i &= v_i^{(q+1)}
\end{aligned}$$

with q denoting the number of stages of the inner method. This results in the new multirate scheme, RFSMR. Again $w_i^{(k)}$ denotes the approximation for w at stage i of the outer method and stage k of the inner method.

$$w_1^{(q+1)} = w(t_n) \tag{15}$$

$$w_i^{(1)} = w_{i-1}^{(q+1)}, \quad i = 2, \ldots, s+1, \tag{16}$$

$$w_i^{(k)} = w_i^{(k-1)} + \Delta t \tilde{c}_k^{\mathrm{I}} \sum_{j=1}^{i-1} \tilde{a}_{ij}^{\mathrm{O}} G\left(w_j^{(q+1)}\right) + \Delta t \tilde{c}_i^{\mathrm{O}} \sum_{j=1}^{k-1} \tilde{a}_{kj}^{\mathrm{I}} F\left(w_i^{(j)}\right), \quad i = 2, \ldots, s+1;\ k = 2, \ldots, q+1 \tag{17}$$

$$w(t_n + \Delta t) = w_{s+1}^{(q+1)}. \tag{18}$$

For the application we have in mind let us assume that for a given ERK method the slow system

$$y' = G(y)$$

and the complete equation (2) can be integrated stable with time steps Δt and $\Delta t/2$ respectively. A simple combination of the same ERK-method as the outer and inner method therefore does not generally result in a stable multirate method and

Table 1
Explicit second order Runge–Kutta methods

0		
1	1	
	1/2	1/2

(RK2a)

0			
1/2	1/2		
1	1	0	
	1/2	0	1/2

(RK32a)

0				
1/2	1/2			
1/2	1/4	1/4		
1	1/4	1/4	1/2	
	1/4	1/4	1/4	1/4

(RK42a)

0		
1/2	1/2	
	0	1

(RK2b)

depends on the distribution of the outer nodes c_i^{O}. Whereas for the classical two-stage method (RK2a) (cp. Table 1) only one inner integration has to be performed with a time interval of length Δt, two inner integrations are necessary for the other classical two-stage method (RK2b) with a time interval of length $\Delta t/2$ for both stages. To get a stable multirate method also for (RK2a) the base method has to be carried out twice with time step $\Delta t/2$ as the inner integration. The same overall method can also be obtained by combining two different methods for the outer and inner method, (RK32a) for the outer method and (RK2a) for the inner method or (RK2a) for the outer method and (RK42a) for the inner method. Note that the method (RK42a) executed once with a time step of Δt is equivalent to (RK2a) executed twice with a time step of $\Delta t/2$.

In the implementation this problem is solved by applying the inner method $n = \lceil 2\tilde{c}_i^{\mathrm{O}} \rceil$ times with a time step of $\Delta t^{(\mathrm{outer})}/n$ for stage i of the outer method. Consequently the fast method advances in time with half the step size of the slow method if for the outer method the following holds:

$$\forall i\colon c_i^{\mathrm{O}} \in \{0, 1/2, 1\}.$$

Other time step ratios R can be obtained by choosing $n = \lceil R\tilde{c}_i \rceil$ if

$$\forall i\colon R\tilde{c}_i^{\mathrm{O}} \in \mathbb{N}_0.$$

For instance if the fast method is to advance with a third of the slow step size (i.e. $R = 3$), employing (RK2b) as outer method will result in four inner integrations corresponding to a time step ratio of 4, whereas the usage of (RK2a) as outer method will actually result in the desired number of three inner integrations.

Additionally we can apply the RFSMR splitting recursively an arbitrary number of times so that any number of temporal refinement levels can be implemented. It is straightforward to implement RFSMR for an arbitrary number of temporal refinement levels via recursive calls of the time integration routine. The formal equivalent is to recursively continue the splitting (2) to

$$\begin{aligned} w' &= F(w) + G(w) \\ &= [F_1(w) + F_2(w)] + G(w) \end{aligned}$$

and integrate the emerging system

$$\frac{\mathrm{d}\hat{v}_i^{(k)}}{\mathrm{d}\tau} = F_1(w) + \frac{1}{\tilde{c}_k^{\mathrm{I}}} \sum_{j=1}^{k-1} \tilde{a}_{kj}^{\mathrm{I}} \left(F_2(w) + \frac{1}{\tilde{c}_i^{\mathrm{O}}} \sum_{j=1}^{k-1} \tilde{a}_{ij}^{\mathrm{O}} G(w) \right),$$

$$\tau \in [0, \tilde{c}_i^{\mathrm{O}} \tilde{c}_k^{\mathrm{I}} \Delta t],\ \hat{v}_i^{(k)} = w_i^{k-1},\ i = 2, \ldots, s+1,\ k = 2, \ldots, q+1,$$

in the same way as (5) using the algorithm (15)–(18).

For order analysis it is convenient to cast (15)–(18) in PRK form. In general an explicit s-stage PRK method for the ODE system (2) can be written in the form:

$$w_{n+1} = w_n + \Delta t \sum_{i=1}^{s} b_i^{\mathrm{F}} F(w_{n,i}) + \Delta t \sum_{i=1}^{s} b_i^{\mathrm{G}} G(w_{n,i}), \tag{19}$$

$$w_{n,i} = w_n + \Delta t \sum_{j=1}^{i-1} a_{ij}^{\mathrm{F}} F(w_{n,j}) + \Delta t \sum_{j=1}^{i-1} a_{ij}^{\mathrm{G}} G(w_{n,j}), \quad i = 1, \ldots, s \tag{20}$$

where $(a_{ij}^{\mathrm{F}}), b_j^{\mathrm{F}}$ and $(a_{ij}^{\mathrm{G}}), b_j^{\mathrm{G}}$ are parameters of Runge–Kutta methods. Usually partitioned Runge–Kutta methods are introduced for a componentwise splitting of ODE systems. According to Ascher et al. [1] the right-hand side splitting can also be transformed into a splitting by components and vice versa.

As for explicit Runge–Kutta methods, formal order conditions can be derived [8]. These conditions consist of the order conditions for the individual Runge–Kutta methods and additional *coupling conditions*. For internally consistent PRKs (i.e. $c^{\mathrm{F}} = c^{\mathrm{G}}$) the order conditions up to order three are listed below.

Order 1: $\sum_{i=1}^{s} b_i^{\mathrm{F}} = \sum_{i=1}^{s} b_i^{\mathrm{G}} = 1$
Order 2: $\sum_{i=1}^{s} b_i^{\mathrm{F}} c_i^{\mathrm{F}} = \sum_{i=1}^{s} b_i^{\mathrm{G}} c_i^{\mathrm{G}} = 1/2$
$\sum_{i=1}^{s} b_i^{\mathrm{F}} c_i^{\mathrm{G}} = \sum_{i=1}^{s} b_i^{\mathrm{G}} c_i^{\mathrm{F}} = 1/2$

Order 3: $\sum_{i=1}^{s} b_i^{\mathrm{F}} \left(c_i^{\mathrm{F}}\right)^2 = \sum_{i=1}^{s} b_i^{\mathrm{G}} \left(c_i^{\mathrm{G}}\right)^2 = \frac{1}{3}$
$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i^{\mathrm{F}} a_{i,j}^{\mathrm{F}} c_j^{\mathrm{F}} = \sum_{i=1}^{s} \sum_{j=1}^{s} b_i^{\mathrm{G}} a_{i,j}^{\mathrm{G}} c_j^{\mathrm{G}} = \frac{1}{6}$
$\sum_{i=1}^{s} b_i^{\mathrm{G}} \left(c_i^{\mathrm{F}}\right)^2 = \sum_{i=1}^{s} b_i^{\mathrm{F}} \left(c_i^{\mathrm{G}}\right)^2 = \frac{1}{3}$
$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i^{\mathrm{G}} a_{i,j}^{\mathrm{F}} c_j^{\mathrm{F}} = \sum_{i=1}^{s} \sum_{j=1}^{s} b_i^{\mathrm{F}} a_{i,j}^{\mathrm{G}} c_j^{\mathrm{G}} = \frac{1}{6}$.

We will now cast RFSMR as a generic PRK method. First we rewrite (17):

$$w_i^{(k)} = w_i^{(k-1)} + \Delta t \sum_{j=1}^{i-1} \underbrace{\tilde{c}_k^{\mathrm{I}} \tilde{a}_{ij}^{\mathrm{O}}}_{=\tilde{a}_{i,j}^{(k),\mathrm{G}}} G\left(w_j^{(s+1)}\right) + \Delta t \sum_{j=1}^{k-1} \underbrace{\tilde{c}_i^{\mathrm{O}} \tilde{a}_{kj}^{\mathrm{I}}}_{=\tilde{a}_{i,j}^{(k),\mathrm{F}}} F\left(w_i^{(j)}\right). \tag{21}$$

Using (12)–(14) we can transform the parameters $\tilde{a}_{i,j}^{(k),\mathrm{F}}$, $\tilde{a}_{i,j}^{(k),\mathrm{G}}$ to the parameters in common notation. The complete method can be represented using

$$A^{\mathrm{I}} = \begin{pmatrix} a_{1,1}^{\mathrm{I}} & & \\ \vdots & \ddots & \\ a_{q,1}^{\mathrm{I}} & \cdots & a_{q,q}^{\mathrm{I}} \end{pmatrix}, \qquad B^{\mathrm{I}} = \begin{pmatrix} b_1^{\mathrm{I}} & \cdots & b_q^{\mathrm{I}} \\ \vdots & \ddots & \vdots \\ b_1^{\mathrm{I}} & \cdots & b_q^{\mathrm{I}} \end{pmatrix},$$

$$C^{\mathrm{I}} = \begin{pmatrix} c_1^{\mathrm{I}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_q^{\mathrm{I}} & 0 & \cdots & 0 \end{pmatrix}, \qquad M = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}, \quad \mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$$

with $C^{\mathrm{I}} \in \mathbb{R}^{q\times q}$, $M \in \mathbb{R}^{q\times q}$, $\mathbf{1} \in \mathbb{R}^{q}$ to read

fast

$c_1^{\mathrm{O}}\mathbf{1} + \tilde{c}_2^{\mathrm{O}} c^{\mathrm{I}}$	$\tilde{c}_2^{\mathrm{O}} A^{\mathrm{I}}$			
$c_2^{\mathrm{O}}\mathbf{1} + \tilde{c}_3^{\mathrm{O}} c^{\mathrm{I}}$	$\tilde{c}_2^{\mathrm{O}} B^{\mathrm{I}}$	$\tilde{c}_3^{\mathrm{O}} A^{\mathrm{I}}$		
$\vdots$	$\vdots$	$\ddots$	$\ddots$	
$c_s^{\mathrm{O}}\mathbf{1} + \tilde{c}_{s+1}^{\mathrm{O}} c^{\mathrm{I}}$	$\tilde{c}_2^{\mathrm{O}} B^{\mathrm{I}}$	$\cdots$	$\tilde{c}_s^{\mathrm{O}} B^{\mathrm{I}}$	$\tilde{c}_{s+1}^{\mathrm{O}} A^{\mathrm{I}}$
	$\tilde{c}_2^{\mathrm{O}} b^{\mathrm{I}}$	$\cdots$	$\tilde{c}_s^{\mathrm{O}} b^{\mathrm{I}}$	$\tilde{c}_{s+1}^{\mathrm{O}} b^{\mathrm{I}}$

slow

$c_1^{\mathrm{O}}\mathbf{1} + \tilde{c}_2^{\mathrm{O}} c^{\mathrm{I}}$	$a_{2,1}^{\mathrm{O}} C^{\mathrm{I}}$			
$c_2^{\mathrm{O}}\mathbf{1} + \tilde{c}_3^{\mathrm{O}} c^{\mathrm{I}}$	$a_{2,1}^{\mathrm{O}} M + \tilde{a}_{3,1}^{\mathrm{O}} C^{\mathrm{I}}$	$a_{3,2}^{\mathrm{O}} C^{\mathrm{I}}$		
$\vdots$	$\vdots$	$\ddots$	$\ddots$	
$c_s^{\mathrm{O}}\mathbf{1} + \tilde{c}_{s+1}^{\mathrm{O}} c^{\mathrm{I}}$	$a_{s,1}^{\mathrm{O}} M + \tilde{a}_{s+1,1}^{\mathrm{O}} C^{\mathrm{I}}$	$\cdots$	$a_{s,s-1}^{\mathrm{O}} M + \tilde{a}_{s+1,s-1}^{\mathrm{O}} C^{\mathrm{I}}$	$a_{s+1,s}^{\mathrm{O}} C^{\mathrm{I}}$
	$b_1^{\mathrm{O}} \mathbf{e}_1^{\mathrm{T}}$	$\cdots$	$b_{s-1}^{\mathrm{O}} \mathbf{e}_1^{\mathrm{T}}$	$b_s^{\mathrm{O}} \mathbf{e}_1^{\mathrm{T}}$

in the common Butcher notation. Starting from this notation it can easily be shown that both fast and slow method as well as the coupled method are second order accurate if the underlying base method is at least second order accurate.

$$(b^{\mathrm{F}})^{\mathrm{T}}\mathbf{1} = \sum_{k=2}^{s+1} \tilde{c}_k^{\mathrm{O}} (b^{\mathrm{I}})^{\mathrm{T}}\mathbf{1} = 1$$

$$(b^{\mathrm{F}})^{\mathrm{T}} c^{\mathrm{F}} = \sum_{k=2}^{s+1} \tilde{c}_k^{\mathrm{O}} (b^{\mathrm{I}})^{\mathrm{T}} \left(c_{k-1}^{\mathrm{O}}\mathbf{1} + \tilde{c}_k^{\mathrm{O}} c^{\mathrm{I}}\right) = \frac{1}{2}$$

$$(b^{\mathrm{G}})^{\mathrm{T}}\mathbf{1} = \sum_{k=2}^{s+1} b_{k-1}^{\mathrm{O}} \mathbf{e}_1^{\mathrm{T}}\mathbf{1} = 1$$

$$(b^{\mathrm{G}})^{\mathrm{T}} c^{\mathrm{G}} = \sum_{k=2}^{s+1} b_{k-1}^{\mathrm{O}} \mathbf{e}_1^{\mathrm{T}} \left(c_{k-1}^{\mathrm{O}}\mathbf{1} + \tilde{c}_k^{\mathrm{O}} c^{\mathrm{I}}\right) = \frac{1}{2}$$

$$c^{\mathrm{G}} = c^{\mathrm{F}} \Rightarrow (b^{\mathrm{F}})^{\mathrm{T}} c^{\mathrm{G}} = (b^{\mathrm{G}})^{\mathrm{T}} c^{\mathrm{F}} = \frac{1}{2}.$$

Knoth and Wolke [10] derived one additional third order consistency condition for their IMEX splitting in addition to the classical order conditions:

$$\sum_{i=1}^{s+1} \tilde{c}_i \sum_{j=1}^{s} \left(a_{ij} + a_{i-1,j}\right) c_j = \frac{1}{3} \tag{22}$$

Table 2
Third order explicit Runge–Kutta methods

0				
1/2	1/2			
1/2	−1/6	2/3		
1	1/3	−1/3	1	
	1/6	1/3	1/3	1/6

(RK43)

Table 3
A third order consistent multirate method, based on (RK43)

0										
1/4	1/4									
1/4	1/4	0								
1/2	1/2	0	0							
1/2	1/2	0	0	0						
1/2	−1/6	0	0	0	2/3					
3/4	1/12	0	0	0	1/6	1/2				
3/4	1/12	0	0	0	1/6	1/2	0			
1	1/3	0	0	0	−1/3	1	0	0		
1	1/3	0	0	0	−1/3	1	0	0	0	
	1/6	0	0	0	1/3	1/3	0	0	0	1/6

Slow method

0										
1/4	1/4									
1/4	−1/12	1/3								
1/2	1/6	−1/6	1/2							
1/2	1/12	1/6	1/6	1/12						
1/2	1/12	1/6	1/6	1/12	0					
3/4	1/12	1/6	1/6	1/12	0	1/4				
3/4	1/12	1/6	1/6	1/12	0	−1/12	1/3			
1	1/12	1/6	1/6	1/12	0	1/6	−1/6	1/2		
1	1/12	1/6	1/6	1/12	0	1/12	1/6	1/6	1/12	
	1/12	1/6	1/6	1/12	0	1/12	1/6	1/6	1/12	0

Fast method

with $a_{s+1,j} = b_j$. They constructed two methods satisfying these conditions, a three-stage third order method (RK3c) and a four-stage third order method (RK43), listed in Table 2. The four-stage method has the convenient nodes $c = (0, 1/2, 1/2, 1)$, suitable for the time step ratio of $R = 2$ desired for our application. We employ this method both as inner and outer method and eliminate redundant rows and columns to obtain the multirate PRK listed in Table 3. The redundant lines are due to Runge–Kutta stages i of the base method with $\tilde{c}_i = 0$ which result in redundant columns l of the generated PRK with

$$\forall j\colon a^{\mathrm{F}}_{j,l} = 0 \ \wedge \ a^{\mathrm{G}}_{j,l} = 0,$$

and consequently in redundant rows, correlated to the Runge–Kutta stages $l = k$ with

$$\forall j\colon a^{\mathrm{F}}_{k,j} = a^{\mathrm{F}}_{k-1,j} \ \wedge \quad a^{\mathrm{G}}_{k,j} = a^{\mathrm{G}}_{k-1,j}$$

conditions. It is an open question whether multirate schemes constructed from methods satisfying the additional third order conditions (22) are generally of third order.

3. Spatial discretization

We now approximate the advection equation (1). Apart from t all variables may be vector valued, e.g. if transport of multiple different contaminants in three-dimensional space is simulated. Additionally u may depend on t and x. To keep notation compact, we will describe in the following uniform advection $u = \text{const}$ of a single species in one spatial dimension.

In the course of spatial discretization a partial differential equation (PDE) is transformed to a semidiscrete ordinary differential equation (semidiscrete ODE). The continuous variable c is transformed to a variable w being discrete in space and continuous in time. For our model we choose a finite volume approximation, so the elements of w represent the average values of c across fixed disjoint spatial intervals:

$$w_j(t) = \frac{1}{h_j} \int_{x_{j-1/2}}^{x_{j+1/2}} c(x, t)\mathrm{d}x, \tag{23}$$

$$h_j = x_{j+1/2} - x_{j-1/2}, \tag{24}$$

$$x_j = \frac{x_{j-1/2} + x_{j+1/2}}{2}. \tag{25}$$

The cell edges at $x_{j+1/2}$ define the grid. We call h_j the cell size and x_j the midpoint of cell j. With the above discretization (1) can be transformed to the semidiscrete *flux form* [7]

$$w_j'(t) = -\frac{1}{h_j}\left(f_{j+1/2}\left(t, w(t)\right) - f_{j-1/2}\left(t, w(t)\right)\right), \tag{26}$$

with the fluxes f at the cell boundaries $x_{j+1/2}$

$$f_{j+1/2} = u w_{j+1/2}.$$

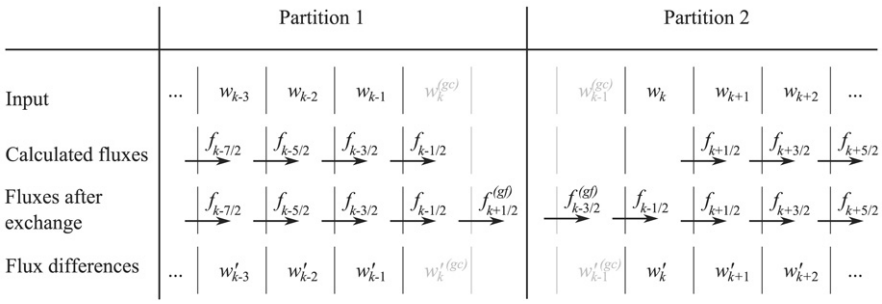


Fig. 1. Flux exchange in RFSMR.

This can be interpreted so that the temporal evolutions w'_j are the differences of incoming and outgoing fluxes $f_{j-1/2}, f_{j+1/2}$, divided by the cell size h_j. The division by the cell volume is correlated to the transformation of a mass flux to a concentration evolution.

We will approximate $f_{j+1/2}$ by different upwind methods commonly used for advection problems. The *first order scheme* for uniform advection reads

$$w_{j+1/2} = \begin{cases} w_j, & \text{if } u \geq 0, \\ w_{j+1}, & \text{if } u < 0. \end{cases} \tag{27}$$

The other spatial discretization we will consider is the positive *third order upwind biased* scheme [9]:

$$\begin{aligned} w_{j+1/2} &= w_j + \max\left(0, \min\left(r_{j+1/2}, 2, 2\left(-\alpha_j + \gamma_j r_{j+1/2}\right)\right)\right)(w_j - w_{j-1}), \\ \alpha_j &= -\frac{h_j h_{j+1}}{(h_{j-1} + h_j)(h_{j-1} + h_j + h_{j+1})}, \\ \gamma_j &= \frac{h_j(h_{j-1} + h_j)}{(h_j + h_{j+1})(h_{j-1} + h_j + h_{j+1})} \end{aligned} \tag{28}$$

for nonnegative advection speeds with the slope ratio

$$r_{j+1/2} = \frac{w_{j+1} - w_j}{w_j - w_{j-1}}.$$

For equidistant grids this scheme can be written as member of the κ-family [6] with $\kappa = 1/3$. The more general formulation (28) allows for the accurate calculation of fluxes at the interface between piecewise equidistant grids with different resolutions.

A widespread approach to apply multirate schemes is to partition the cells of the grid according to the characteristic time per cell [2]. Based on this common partitioning we decompose the faces of the grid on which the fluxes are defined so that each partition contains the outgoing fluxes of the corresponding cell partition. As the flux differences form the right-hand side, this decomposition by components leads to a natural decomposition of the right-hand side. Due to the partitioning of the fluxes, mass preservation is guaranteed at any stage of the explicit PRK method. Additionally, as for the calculation of a cell's outflow the most recent intermediate result is employed, the method preserves positivity. For further discussion let us assume that the slow and fast cells correspond to a partitioning of the grid in two contiguous parts with a common interface.

This partitioning is shown schematically in Fig. 1. The first row shows the partitioning of the cells. Assuming positive wind speed the second row shows the partitioning of the fluxes. In practical applications additional ghost cells including their outer face are added at the interface between the two partitions. Thus we can store additional cell values $w^{(gc)}$ and additional face fluxes $f^{(gf)}$. Due to the positive wind speed and assuming a third order upwind spatial discretization the flux at the interface $f_{k-1/2}$ depends only on the cell values w_{k-2}, w_{k-1}, w_k, all of which are present in some form in Partition 1. Partition 2 contains cells from index k as ordinary cells and cell $k-1$ as ghost cell.

The second row of the table shows the fluxes that can be calculated per partition. After the calculation of the fluxes neighboring partitions must exchange fluxes. In Fig. 1 the fluxes received are noted with a superscript *(gf)* for ghost flux.

After the flux exchange each partition can compute the inflow/outflow difference for any cell contained in this partition. Note that for a negative sign of the wind speed the flux $f_{k-1/2}$ at the interface depends on concentrations w_{k-1}, w_k, w_{k+1}, so that the interface flux must be calculated by Partition 2.

For further clarification we will now present one macro time step integrated with a multirate scheme based on forward Euler steps. We will only regard the cells at the interface including ghost cells. The grid is decomposed as in Fig. 1, Partition 1 is integrated with the half macro time step. Data contained in the data structure of Partition 1 is denoted with a hat.

Initialization	$\hat{w}^1_{k-1} = \hat{w}_{k-1}(t)$
	$\hat{w}^{1,(gc)}_k = \hat{w}^{(gc)}_k(t)$
	$w^1_k = w_k(t)$
	$w^{1,(gc)}_{k-1} = w^{(gc)}_{k-1}(t)$
Exchange P1 $\Leftarrow$ P2	$\hat{f}^{1,(gf)}_{k+1/2} = f^1_{k+1/2}$
Time integration P1	$\hat{w}^2_{k-1} = \hat{w}^1_{k-1} + \frac{\Delta t}{2h_{k-1}}\left(\hat{f}^1_{k-1/2} - \hat{f}^1_{k-3/2}\right)$
	$\hat{w}^{2,(gc)}_k = \hat{w}^{1,(gc)}_k + \frac{\Delta t}{2h_k}\left(\hat{f}^{1,(gf)}_{k+1/2} - \hat{f}^1_{k-1/2}\right)$
	$\hat{f}^{2,(gf)}_{k+1/2} = \hat{f}^{1,(gf)}_{k+1/2}$
	$\hat{w}^3_{k-1} = \hat{w}^2_{k-1} + \frac{\Delta t}{2h_{k-1}}\left(\hat{f}^2_{k-1/2} - \hat{f}^2_{k-3/2}\right)$
	$\hat{w}^{3,(gc)}_k = \hat{w}^{2,(gc)}_k + \frac{\Delta t}{2h_k}\left(\hat{f}^{2,(gf)}_{k+1/2} - \hat{f}^2_{k-1/2}\right)$
Exchange P1 $\Rightarrow$ P2	$f^{\mathrm{cum},(gf)}_{k-3/2} = \left(\hat{f}^1_{k-3/2} + \hat{f}^2_{k-3/2}\right)/2$
	$f^{\mathrm{cum}}_{k-1/2} = \left(\hat{f}^1_{k-3/2} + \hat{f}^2_{k-3/2}\right)/2$
Time integration P2	$w^3_k = w^1_k + \frac{\Delta t}{h_k}\left(f^1_{k+1/2} - f^{\mathrm{cum}}_{k-1/2}\right)$
	$w^{3,(gc)}_{k-1} = w^{1,(gc)}_{k-1} + \frac{\Delta t}{h_{k-1}}\left(f^{\mathrm{cum}}_{k-1/2} - f^{\mathrm{cum},(gf)}_{k-3/2}\right)$
Finalization	$\hat{w}_{k-1}(t+\Delta t) = \hat{w}^3_{k-1}$
	$\hat{w}^{(gc)}_k(t+\Delta t) = \hat{w}^{3,(gc)}_k$
	$w_k(t+\Delta t) = w^3_k$
	$w^{(gc)}_{k-1}(t+\Delta t) = w^{3,(gc)}_{k-1}$

One can show that this algorithm yields the same results as the naive application of our multirate scheme. Note that we need only two unidirectional exchanges while in a naive implementation four unidirectional exchanges or two bidirectional exchanges would be necessary.

4. Numerical results

In this section we will examine stability and convergence of multirate schemes generated with RFSMR and compare these results to those obtained employing the underlying base method in classical (i.e. singlerate) form.

For numerical tests we will employ a one-dimensional uniform advection equation as test equation

$$\frac{\partial c}{\partial t} = -\frac{\partial c}{\partial x}$$

with periodic boundary conditions

$$c(t, 0) = c(t, 1).$$

We use two different grid structures to discretize the continuous interval $[0, 1]$: an equidistant grid with grid spacing $h = 0.01$ resulting in 100 cells and a nonequidistant grid consisting of three equidistant sub-grids with $h = 0.02$ on the intervals $[0, 0.26)$, $(0.74, 1]$ and $h = 0.01$ on $[0.26, 0.74]$. This results in two coarse partitions of 13 cells each corresponding to the first two sub-intervals and one fine partition of 48 cells corresponding to the third sub-interval. Note that due to stability considerations the maximum time step on the finer cells is only half as large as on the coarser cells.

4.1. Stability analysis

For the stability analysis we will employ a triangle pulse as initial profile:

$$c(x, t = 0) = \begin{cases} 10x - 4 & \text{if } x \in [0.4, 0.5) \\ -10x + 6 & \text{if } x \in [0.5, 0.6] \\ 0 & \text{otherwise.} \end{cases}$$

Our measure for stability is the *total variation* (TV) seminorm:

$$TV(t) = \sum_j |w_j(t) - w_{j-1}(t)|, \quad t > 0.$$

A spatially and temporally discrete system is called *total variation diminishing* (TVD) if

$$\forall t, \Delta t > 0 : TV(t + \Delta t) \leq TV(t).$$

We compared base methods (RK2a) and (RK43) using the positive third order upwind biased spatial discretization (28), accounting for the relative grid spacing of the correlated cells. Fig. 2 shows distinct differences between the TV developing

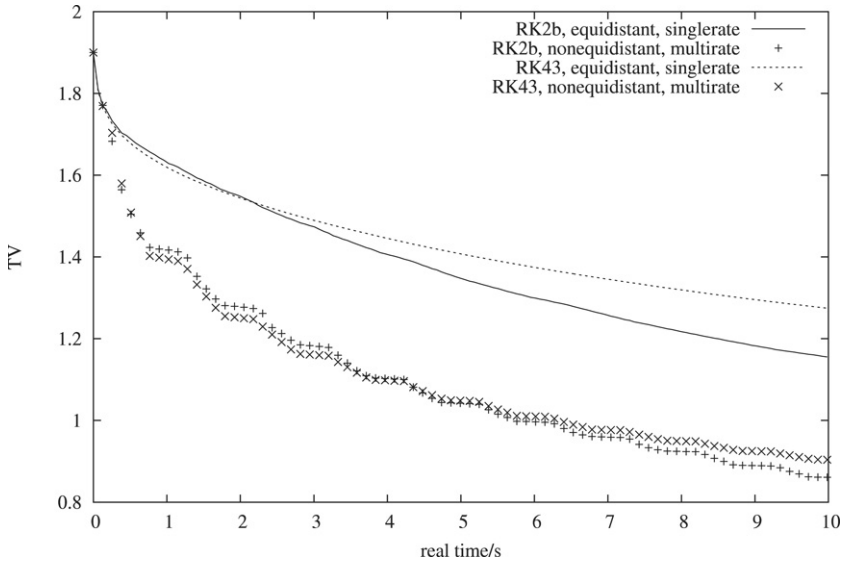


Fig. 2. Comparison of different base methods. Positive third order upwind biased spatial discretization, base methods (RK2a) ($\Delta t = 0.9$) and (RK43) ($\Delta t = 0.8$).

Table 4
A multirate scheme based on (RK32a)/(RK2a)

0					
1/2	1/2				
1/2	1/2	0			
1	1	0	0		
1	1	0	0	0	
1	1/2	0	0	0	1/2

Slow method

0					
1/2	1/2				
1/2	1/4	1/4			
1	1/4	1/4	1/2		
1	1/4	1/4	1/4	1/4	
1	1/4	1/4	1/4	1/4	0

Fast method

for (RK2a) and (RK43). Moreover the graphs for equidistant and nonequidistant grids are not only differently steep but also show clear qualitative differences. The graphs obtained with equidistant grids are very steady while for nonequidistant grids segments with steep and shallow gradient alternate periodically. Again this is a consequence of the diffusive effects of the larger cells: while the cone passes the coarse cells it is smeared, passing the fine cells its form is preserved. The periodic boundary conditions combined with the wind speed of $u = 1$ account for the period of $\Delta t = 1$. Partially these effects may result from the fact that due to the use of limiters the employed spatial discretization is nonlinear. We found similar results for all tested combinations of discretizations.

Fig. 3 shows profiles on a nonequidistant grid developed with (RK2a) using different time steps and the multirate method based on (RK32a)/(RK2a), cp. Table 4. A possible interpretation is that the multirate method locally switches between the smaller time step in the fine central region and the larger time step in the remaining intervals. Within the fine region the profile obtained using multirate time integration is identical to the profile obtained using the classical method with the smaller time step. A deviation is visible starting from the first coarse and therefore slow cell indicated by the dashed vertical line, see Fig. 3(b).

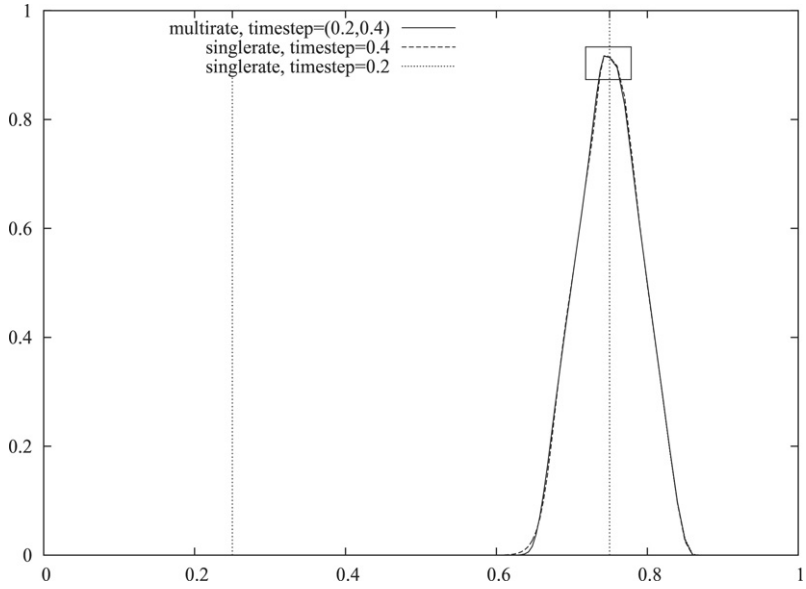
All of the above results confirm that the multirate schemes constructed via RFSMR are total variation diminishing for TVD base methods. This holds for any tested full discretization. The time step restrictions of the RFSMR schemes equal the restrictions of the underlying base method.

4.2. Order analysis

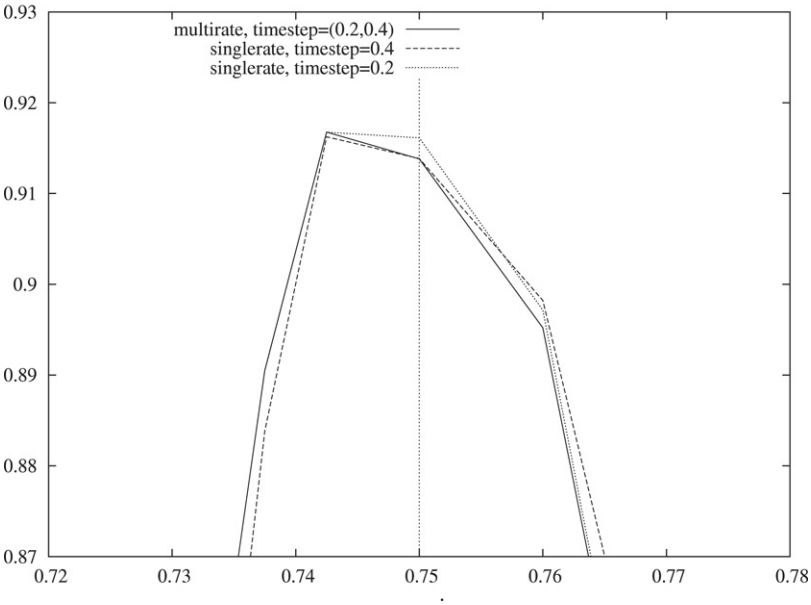
To get an outline of the methods' convergence behavior we will compute numerical solutions for different time steps $\Delta t^{(k)}$ and examine the deviation of $w^{(k)}(t = 1)$ from a reference solution $\tilde{w}$ in a weighted L^1-Norm

$$\|w - \tilde{w}\| = \sum_j h_j |w_j - \tilde{w}_j| \approx \int_0^1 |c(x) - \tilde{c}(x)| \mathrm{d}x.$$

Our reference is obtained by applying a four-stage fourth order singlerate method with $\Delta t = 10^{-5}$. This time step is one order of magnitude below the smallest time step of the remaining samples. All computations in this subsection were done



(a) Total interval.



(b) Detail.

Fig. 3. Profile leaving fine grid.

using the first order upwind spatial discretization and a relatively soft initial profile

$$c(x, t = 0) = \sin^{10}(\pi x).$$

We compare these results to those obtained by applying the underlying base method. For additional comparison we listed the results for the correlated schemes based on the generic multirate scheme of Constantinescu and Sandu [2], see Tables 5 and 6.

Results are shown in Fig. 4(a). Fitting a function of the family $\mathrm{Err}(\nu) = a \cdot \nu^b$ we obtain the results shown in Table 7. As already shown analytically we can obtain third order accuracy by applying RFSMR to (RK43) as base method, even though the error factor is five times as high as for singlerate (RK43). This difference is due to the larger time steps of the multirate methods performed in the coarse region of the grid. As mentioned in [2] their schemes are maximum second order accurate.

Table 5
Multirate scheme according to Constantinescu and Sandu [2], based on (RK2a)

0				
1	1			
0	0	0		
1	0	0	1	
	1/4	1/4	1/4	1/4

Slow method

0				
1/2	1/2			
1/2	1/4	1/4		
1	1/4	1/4	1/2	
	1/4	1/4	1/4	1/4

Fast method

Table 6
Multirate scheme according to Constantinescu and Sandu [2], based on (RK43)

0								
1/2	1/2							
1/2	−1/6	2/3						
1	1/3	−1/3	1					
0	0	0	0	0				
1/2	0	0	0	0	1/2			
1/2	0	0	0	0	−1/6	2/3		
1	0	0	0	0	1/3	−1/3	1	
	1/12	1/6	1/6	1/12	1/12	1/6	1/6	1/12

Slow method

0								
1/4	1/4							
1/4	−1/12	1/3						
1/2	1/6	−1/6	1/2					
1/2	1/12	1/6	1/6	1/12				
3/4	1/12	1/6	1/6	1/12	1/4			
3/4	1/12	1/6	1/6	1/12	−1/12	1/3		
1	1/12	1/6	1/6	1/12	1/6	−1/6	1/2	
1	1/12	1/6	1/6	1/12	1/12	1/6	1/6	1/12

Fast method

Table 7
Empirical order for time integration methods

Method	Error	Speedup (emp.) (%)	Speedup (theor.) (%)
singlerate RK2a	$\approx 0.00625\Delta t^{2.0001}$	–	–
Constantinescu and Sandu RK2a	$\approx 0.01588\Delta t^{2.0000}$	−3.05	12.9
RFSMR RK2a	$\approx 0.01580\Delta t^{2.0001}$	22.51	20.78
singlerate RK43	$\approx 0.00009\Delta t^{3.0006}$	–	–
Constantinescu and Sandu RK43	$\approx 0.00035\Delta t^{2.0041}$	0.18	10.39
RFSMR RK43	$\approx 0.00045\Delta t^{3.0332}$	14.62	20.78

This is confirmed by our results, even though the errors for large time steps are nearly identical with the errors obtained using RFSMR (RK43).

Also listed in Table 7 is the speedup, which is calculated via

$$\mathrm{SU}_{\mathrm{method}} = 1 - \frac{T_{\mathrm{method}}}{T_{\mathrm{singlerate}}},$$

where T is the measured computational time for the empirical speedup and the number of elementary flux calculations for the theoretical speedup respectively. Note that due to the splitting by fluxes employed in the RFSMR algorithm the slow/fast interface is strictly localized on one single edge. Opposed to this for the schemes constructed according to Constantinescu and Sandu [2] the influence of the faster partition grows into the slower partition in course of each Runge–Kutta stage. The relative effect of this issue decreases with a growing number of cells per interface. All in all the simplicity of the RFSMR algorithm leads both to superior empirical and theoretical speedup.

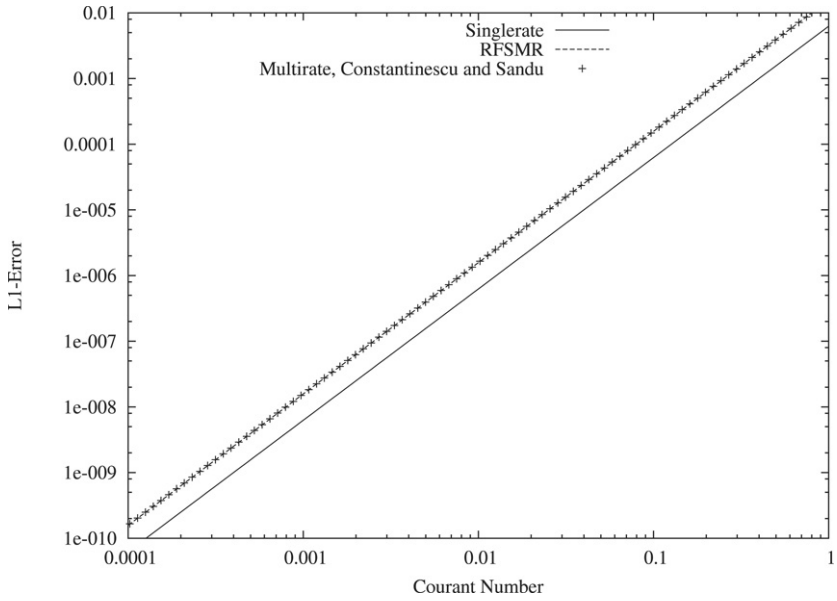
Note that the speedup of any multirate method will grow with the ratio of cells integrated with the largest time step. The theoretical speedup of RFSMR is given by

$$\mathrm{SU}_{\mathrm{RFSMR}} \approx 1 - \frac{\sum_{i=1}^{n} 2^i C_i}{\sum_{i=1}^{n} 2^n C_i},$$

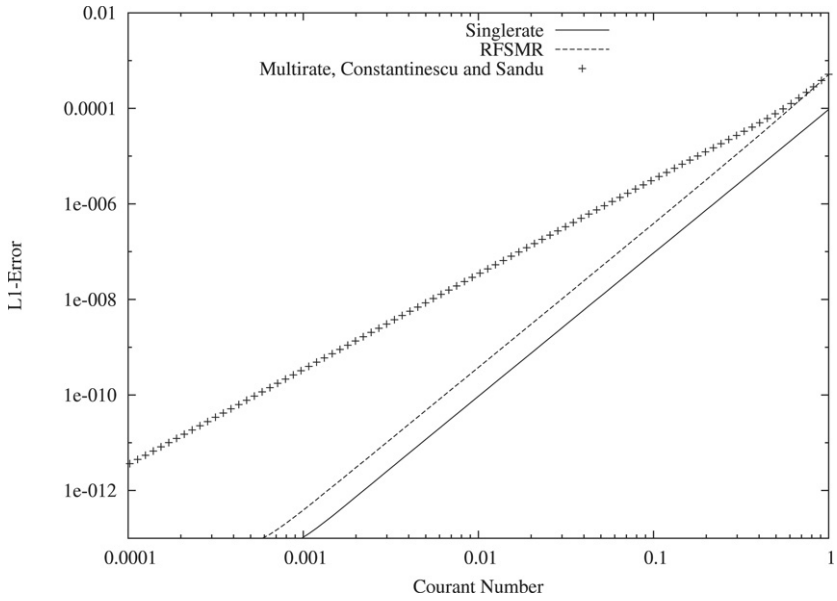
with C_i denoting the number of cells which can be stably integrated with a maximum time step in $[2^{-i}\Delta t_0, 2^{1-i}\Delta t_0)$.

5. Conclusions and future work

Realistic simulations often lead to a wide range of spatial grid sizes and consequently very different time step size restrictions across the domain to simulate. Thus a majority of cells is integrated with a time step much smaller than



(a) Base method RK2a.



(b) Base method RK43.

Fig. 4. Multirate methods: Error vs. Courant number.

necessary. Multirate time integration methods can be employed to alter the time step locally, leading to a significant reduction of computational cost.

In this paper we present a right-hand side splitting approach that can be used to extend classical explicit Runge–Kutta methods to multirate schemes. These schemes are of second order consistency if the underlying method is of second order consistency. Third order consistency is not generally inherited from the base method. We are however able to construct a third order consistent multirate scheme based on an explicit third order Runge–Kutta method satisfying additional order conditions. Our results imply that any ERK satisfying these additional conditions can be extended to a third order multirate scheme. We found evidence that the new multirate schemes inherit the stability and TVD properties of the base method, even though this still has to be examined analytically.

The multirate scheme proposed in this paper shall be implemented in the parallel, three-dimensional Multiscale Aerosol Chemistry Transport (MUSCAT) model, employed at the Institute for Tropospheric Research (IfT). Currently this model uses classical explicit time integration schemes. A special issue will be the adaption of the flux exchange strategy to higher

dimensional nonequidistant grids. Furthermore an automatic time step selection must be implemented, which determines time steps per logical partition of the simulation domain so that both local and global time step constraints are satisfied.

References

[1] U. Ascher, S. Ruuth, R. Spiteri, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2–3) (1997) 151–167.
[2] E. Constantinescu, A. Sandu, Multirate timestepping methods for hyperbolic conservation laws, SIAM J. Sci. Comput. 33 (3) (2007) 239–278.
[3] E. Constantinescu, A. Sandu, G. Charmichael, Modeling Atmospheric Chemistry and Transport with Dynamic Adaptive Resolution, in: Lecture Notes in Computer Science, vol. 3515/2005, Springer, Berlin, Heidelberg, 2005, pp. 798–805.
[4] W. Dabberdt, M. Carroll, D. Baumgardner, G. Charmichael, R. Cohen, T. Dye, J. Ellis, G. Grell, S. Grimmond, S. Hanna, J. Irwin, B. Lamb, S. Madronich, J. McQueen, J. Meagher, T. Odman, J. Pleim, H. Schmid, D. Westphal, Meteorological research needs for improved air quality forecasting, Bull. Amer. Meteorological Soc. 85 (2004) 563–586.
[5] C. Gear, D. Wells, Multirate linear multistep methods, BIT 24 (4) (1984) 484–502.
[6] W. Hundsdorfer, B. Koren, M. van Loon, J. Verwer, A positive finite-difference advection scheme, J. Comput. Phys. 117 (1995) 35–46.
[7] W. Hundsdorfer, J. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion Reaction Equations, in: Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, New York, 2003.
[8] Z. Jackiewicz, R. Vermiglio, Order conditions for partitioned Runge–Kutta methods, Appl. Math. 45 (4) (1998) 301–316.
[9] O. Knoth, R. Wolke, An explicit–implicit numerical approach for atmospheric chemistry-transport modeling, Atm. Environ. 32 (10) (1998) 1785–1797.
[10] O. Knoth, R. Wolke, Implicit–explicit Runge–Kutta methods for computing atmospheric reactive flows, Appl. Numer. Math. 28 (2–4) (1998) 327–341.
[11] S. Osher, R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, Math. Comput. 41 (1983) 321–336.
[12] H. Tang, G. Warnecke, A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and space grids, SIAM J. Sci. Comput. 26 (4) (2005) 1415–1431.
[13] L. Wicker, W. Skamarock, A time-splitting scheme for the elastic equations incorporating second-order Runge–Kutta time differencing, Month. Weather Rev. 126 (1998) 1992–1999.
[14] R. Zvan, P. Forsyth, K. Vetzal, Robust numerical methods for PDE models of Asian options, J. Comp. Fin. 1 (1998) 39–78.