J Sci Comput (2007) 33: 239–278
DOI 10.1007/s10915-007-9151-y

Multirate Timestepping Methods for Hyperbolic Conservation Laws

Emil M. Constantinescu · Adrian Sandu

Received: 17 March 2007 / Revised: 12 July 2007 / Accepted: 25 July 2007 /
Published online: 9 September 2007

Abstract This paper constructs multirate time discretizations for hyperbolic conservation laws that allow different timesteps to be used in different parts of the spatial domain. The proposed family of discretizations is second order accurate in time and has conservation and linear and nonlinear stability properties under local CFL conditions. Multirate timestepping avoids the necessity to take small global timesteps (restricted by the largest value of the Courant number on the grid) and therefore results in more efficient algorithms. Numerical results obtained for the advection and Burgers' equations confirm the theoretical findings.

Keywords Multirate time integration · Hyperbolic conservation laws · Nonlinear stability · Strong stability preservation

1 Introduction

Hyperbolic conservation laws are of great practical importance as they model diverse physical phenomena that appear in mechanical and chemical engineering, aeronautics, astrophysics, meteorology and oceanography, financial modeling, environmental sciences, etc. Representative examples are gas dynamics, shallow water flow, groundwater flow, non-Newtonian flows, traffic flows, advection and dispersion of contaminants, etc.

Conservative high resolution methods with explicit time discretization have gained widespread popularity to numerically solve these problems [34]. Stability requirements limit the temporal step size, with the upper bound being determined by the ratio of the temporal and spatial meshes and the magnitude of the wave speed. Local spatial mesh refinement reduces the allowable timestep for the explicit time discretizations. The timestep for the entire domain is restricted by the finest mesh patch or by the highest wave velocity, and is typically (much) smaller than necessary for other variables in the computational domain.

This work was supported by the National Science Foundation through award NSF CCF-0515170.

E.M. Constantinescu · A. Sandu (✉)
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg 24061, USA
e-mail: asandu@cs.vt.edu

E.M. Constantinescu
e-mail: emconsta@cs.vt.edu

One possibility to circumvent this restriction is to use implicit, unconditionally stable timestepping algorithms which allow large global timesteps. However, this approach requires the solution of large (nonlinear) systems of equations. Moreover, the quality of the solution, as given by a maximum principle, may not be conserved with high order implicit schemes unless the timestep is also restricted by a CFL-like condition.

In this work we develop multirate time integration schemes for the simulation of PDEs. In this approach the timestep can vary across the spatial domain and has to satisfy the CFL condition only locally, resulting in substantially more efficient overall computations. We follow the method of lines (MOL) framework, where the temporal and spatial discretizations are independent.

The development of multirate integration is challenging due to the conservation and stability constraints that need to be satisfied by the timestepping schemes. The algorithms used in the solution of conservation laws need to preserve the system invariants. Moreover, the solutions to hyperbolic PDEs may be non-smooth: shock waves or other discontinuous behavior can develop even from smooth initial data. In such cases, strong-stability-preserving (SSP) numerical methods which satisfy nonlinear stability requirements are necessary to avoid certain types of nonphysical behavior (spurious oscillations, etc.)

A zooming technique for wind transport of air pollution discussed in [4] is a positive, conservative finite volume discretization that allows the use of smaller timesteps in the region of fine grid resolution. The flux at the coarse-to-fine interface is applied in the very first fine sub-step in order to preserve positivity.

Dawson and Kirby [8, 30] developed second order local timestepping. The maximum principle, TVD property, and entropy condition are all fulfilled by the second order finite volume method with two level timestepping; however, the timestepping accuracy of the overall method is first order. Tang and Warnecke [52] reformulated Dawson and Kirby's algorithm in terms of solution increments to obtain second order consistency in time for two-rate integration. Savcenco et al. [43, 44] develop a multirate approach for parabolic equations using a locally self-adjusting multirate timestepping.

In this paper we develop a general systematic approach to extend strongly stable Runge–Kutta (RK) methods [13, 17, 26] to multirate Runge–Kutta methods that inherit the strong stability properties of the corresponding single rate integrators. The second order of accuracy of the overall scheme is preserved, unlike previous multirate approaches that lead to first order accuracy due to the interface treatment [4, 30]. Moreover, for conservation laws, this multirate approach is conservative (preserves the system invariants).

This paper is structured as follows: in Sect. 2 we review the main properties and issues with the simulation of hyperbolic conservation laws. Section 3 presents the construction of the multirate time integrators from single rate integrators. Our numerical results with two types of conservation laws are shown and discussed in Sect. 4. Conclusions and future research directions are given in Sect. 5.

2 Hyperbolic Conservation Laws

We consider the one-dimensional scalar hyperbolic equation

$$\frac{\partial y(t,x)}{\partial t} + \frac{\partial f\left(y(t,x)\right)}{\partial x} = 0,$$
$$\text{with } y(0,x) = y^0(x), \text{ on } x \in \Psi \subset (-\infty, \infty),\ t > 0. \tag{1}$$

Conservative space discretizations of (1) are considered in this work. In the one-dimensional finite volume approach, the change in the mean quantity in the ith cell depends on the fluxes through the cell boundaries at $i \pm \frac{1}{2}$. The semi-discrete (MOL) finite volume approximation can be written as

$$y_i' = -\frac{1}{\Delta x}\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right), \qquad y_i(t) = \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} y(t,x)\,dx, \qquad y_i' = \frac{\partial y_i}{\partial t}, \tag{2}$$

where $y_i(t)$ is the numerical solution at time t and grid point x_i, $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is the grid spacing, and $F_{i+\frac{1}{2}} = F(y_{i+k-1}, \ldots, y_{i-k})$ is the numerical flux at the control volume face. The following notation will be used to denote the discrete solution $y_i^n = y(t^n, x_i)$, with $n > 0$ and $x_i \in \Omega$ (the discrete domain).

To provide physically meaningful solutions and avoid weak nonlinear instabilities (spurious oscillations), the numerical solution has to satisfy a stability condition. Next we review some stability properties of the numerical solution which are used throughout this paper.

Maximum principle Exact solutions of hyperbolic problems have a range-diminishing property that prevents the increase of existing maxima, the decrease of existing minima, and the formation of new maxima or minima. Formally, it can be written as

$$\max_i\left(y_i^n\right) \le \max_i\left(y_i^{n-1}\right) \quad \text{and} \quad \min_i\left(y_i^n\right) \ge \min_i\left(y_i^{n-1}\right). \tag{3}$$

TVD The total variation (TV) of the numerical solution is defined as

$$\mathrm{TV}\left(\{y^n\}\right) = \sum_{i\in\Omega}\left|y_{i+1}^n - y_i^n\right|, \tag{4}$$

where $\mathrm{TV}(\{\bullet\})$ is the total variation norm. A numerical method is called total variation diminishing (TVD) [25] if

$$\mathrm{TV}\left(\{y^n\}\right) \le \mathrm{TV}\left(\left\{y^{n-1}\right\}\right). \tag{5}$$

No spurious spatial oscillations are introduced during timestepping with TVD methods.

TVB A numerical method is called total variation bounded (TVB) (see [47]) if

$$\mathrm{TV}\left(\{y^n\}\right) \le B\cdot\mathrm{TV}\left(\left\{y^0\right\}\right), \quad \forall t,\ 0 \le n \le T,\ B > 0. \tag{6}$$

In this case some bounded total variation increase is allowed. TVD methods are also TVB.

Monotonicity-preservation Monotonic schemes have the property that if $y_i^0 = y(t=0, x_i)$ is monotonically increasing or decreasing in i, then so is $\{y_i(t)\}_i$ for all t. A TVD scheme is monotonicity-preserving.

Positivity Solution positivity is a typical requirement in various applications (e.g., chemical engineering, meteorology, financial modeling, etc.). The semi-discrete scheme (2) is positive if, whenever the initial condition is non-negative, the solution at all future times $t > 0$ remains non-negative. A sufficient condition for the positivity of the semi-discrete system (2) [27] is

$$y_i(t) = 0 \quad \text{and} \quad y_j(t) > 0 \quad \text{for } \forall j \ne i \quad \Rightarrow \quad y_i' \ge 0. \tag{7}$$

Typically, the above properties are established for different spatial discretizations in conjunction with the forward Euler timestepping method [34]. The forward Euler time discretization method with certain spatial discretizations has strong CFL restrictions and is only first order accurate (in time). Following [18], in this paper we use convex combinations of forward Euler time steps combined with appropriate spatial discretizations in order to preserve the stability properties while increasing the order of accuracy and alleviating the CFL restrictions. Unless specified otherwise, the nonlinear stability properties of various spatial discretizations of type (2) are defined for forward Euler timestepping under the appropriate CFL restriction.

Several methods to approximate the fluxes in (2) have been developed in the past decades. Godunov's method [15] is based on the exact solution of Riemann problems. The flux-corrected transport method proposed by Boris and Book [5] and further developed by Zalesak [54] and Roe [41] established the basic principles for the construction of high resolution methods. Upwind biased interpolation is coupled with limiters [51] which reduce the order of accuracy of the scheme near discontinuities (e.g., reducing a high order interpolant to first order, and further limiting its slope). Limiters allow the construction of TVD schemes [25] for nonlinear scalar one-dimensional problems.

All these spatial discretization methods satisfy some of the above stability properties (maximum principle, TVD, TVB, monotonicity-preservation, or positivity). High order timestepping methods based on convex combinations of explicit Euler steps [18] are typically used to solve the semi-discrete form (2), which under a CFL-like condition maintains the stability properties of the spatial discretization. One can construct implicit timestepping methods which are unconditionally linearly stable; however, the nonlinear stability properties restrict the integration timestep to a CFL-like condition [18]. Moreover, the implicit methods require the solution of (non)linear systems at each step. Considering these aspects, explicit discretization methods are preferred for the solution of (2). In the next section we briefly review explicit Runge–Kutta methods and relevant stability properties.

2.1 Explicit Runge–Kutta Methods

The MOL approach applied to (1) yields the semi-discrete problem (2) which needs to be solved forward in time. An s stage explicit Runge–Kutta method [23] computes the next step solution y^{n+1} (at time $t^{n+1} = t^n + \Delta t$) from the current solution y^n at t^n using the formula:

$$\begin{aligned} y^{n+1} &= y^n + \Delta t \sum_{i=1}^{s} b_i K_i, \\ K_i &= f\left(t + c_i \Delta t, y^n + \Delta t \sum_{j=1}^{i-1} a_{i,j} K_j \right). \end{aligned} \tag{8}$$

The method is defined by its coefficients $A = \{a_{ij}\}$, $b = \{b_i\}$, and $c = \{c_i\}$, which can be conveniently represented in the form of the Butcher tableau [23]

$$\mathcal{RK} = [A, b, c] := \frac{c \,|\, A}{\quad | b^T} \qquad \begin{array}{c|cccc} c_1 = 0 & 0 & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} \quad b_s \end{array} . \tag{9}$$

All RK methods in this paper have the property that $c_i = \sum_{j=1}^{i-1} a_{i,j}$. The order conditions for these methods are

$$\text{Order I}: \quad \sum_{i=1}^{s} b_i = 1, \tag{10}$$

$$\text{Order II}: \quad \sum_{i=1}^{s} \sum_{j=1}^{s-1} b_i a_{i,j} = \sum_{i=1}^{s} b_i c_i = \frac{1}{2}. \tag{11}$$

2.2 Strong Stability Preservation

Strong stability preserving (SSP) integrators are timestepping schemes that ensure that a certain norm or semi-norm of the solution does not increase:

$$\| y^{n+1} \| \leq \| y^n \|, \tag{12}$$

where $\| \bullet \|$ is some norm (e.g., for ∞ norm we have the maximum principle condition, for TV semi-norm we have the TVD property, etc.). Spurious oscillations (nonlinear instabilities) can occur in a numerical solution obtained with a TVD or MUSCL spatial discretization scheme, when time discretization is done with a linearly stable timestepping scheme [18]. When spatial discretizations that are TVD with forward Euler timestepping are combined with high order SSP timestepping, they compute numerical solutions that do not exhibit nonlinear instabilities. Hence, SSP timestepping schemes are a critical part of the overall solution strategy.

The favorable properties of SSP schemes derive from convexity arguments. In particular, if the forward Euler method is strongly stable (under a certain CFL timestep restriction), higher-order methods can be constructed as convex combinations of forward Euler steps with various step sizes [48, 49]. SSP methods preserve the strong stability of the forward Euler scheme under specific timestep restrictions.

Gottlieb et al. [18] discuss in detail Runge–Kutta and linear multistep SSP schemes. They derive optimal SSP methods with minimal number of function evaluations, high order, low storage, and establish that implicit Runge–Kutta or linear multistep SSP methods are of order one at most. Hundsdorfer et al. [28] provide an analysis of monotonicity properties for linear multistep methods, and Spiteri and Ruuth [50] extend the SSP Runge–Kutta class of methods by removing the constraint that the order and the number of RK stages be equal.

Several examples of SSP Runge–Kutta are given by Shu and Osher [48]. Here we consider the second order method RK2a defined by the Butcher tableau below

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \qquad \begin{aligned} &K_1 = f(y^n), \quad y^{(1)} = y^n + \Delta t K_1, \\ &K_2 = f(y^{(1)}), \\ &y^{n+1} = y^n + \frac{\Delta t}{2}(K_1 + K_2). \end{aligned} \tag{13}$$

We use the following compact notation for explicit Euler steps

$$\mathcal{E}(\Delta t, y) := y(t) + \Delta t \cdot f(t, y). \tag{14}$$

The method (13) can be written as convex combinations of Euler steps [48] and therefore is SSP:

$$
\begin{aligned}
y^{n+1} &= y^n + \frac{\Delta t}{2}\left(K_1 + K_2\right), & y^{(1)} &= y^n + \Delta t f\left(y^n\right) = \mathcal{E}\left(\Delta t, y^n\right), \\
&= \frac{1}{2}y^n + \frac{1}{2}\left(y^{(1)} + \Delta t K_2\right), & y^{(1*)} &= y^{(1)} + \Delta t f\left(y^{(1)}\right) = \mathcal{E}\left(\Delta t, y^{(1)}\right), \\
&= \frac{1}{2}y^n + \frac{1}{2}y^{(1*)}, & y^{n+1} &= \frac{1}{2}\left(y^n + y^{(1*)}\right) \\
& & &= \frac{1}{2}\mathcal{E}\left(0, y^n\right) + \frac{1}{2}\mathcal{E}\left(\Delta t, y^{(1)}\right).
\end{aligned}
\tag{15}
$$

3 Multirate Time Integration

The idea of multirate timestepping is to take different timesteps for different components to achieve the target accuracy. Slower components are integrated using larger step sizes. The large step sizes are integer multiple of the small step sizes: $\Delta t_{\text{slow}} = \mathfrak{m}\Delta t_{\text{fast}}$. All steps are synchronized every largest timestep Δt_{slow} in order to obtain the desired overall accuracy. Conditions for high orders of accuracy for the multirate integrators (at the synchronization times) are derived in the literature [22, 24].

Early efforts to develop multirate Runge–Kutta methods are due to Rice [40] and Andrus [1, 2]. Multirate versions of many of the traditional timestepping schemes have been proposed in the literature, including linear multistep [14, 29], extrapolation [11], Runge–Kutta [20, 31, 32], Rosenbrock–Wanner [3, 21], waveform relaxation [42], Galerkin [35–37], and combined multiscale [10] approaches.

Kværnø and Rentrop [31, 32] developed multirate Runge–Kutta (MRK) methods where the coupling between slow and fast components is done by intermediate stage values. Günther et al. [20] developed multirate partitioned Runge–Kutta (MPRK) schemes which generalize both partitioned Runge–Kutta and multirate ROW methods [19], and Sand and Burrage [42] developed the Jacobi waveform relaxation approach.

For the purpose of simplicity (and without the loss of generality), in the following sections we restrict our discussion to scalar one-dimensional equations. Multidimensional/multivariable extensions follow by the same arguments.

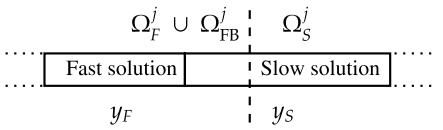
Consider the system of ordinary differential equations in (2) resulting from the application of MOL to (1). Variables are partitioned according to their characteristic times:

$$
y = [y_{\tau(1)}, \ldots, y_{\tau(\mathrm{M})}]^T, \qquad y_i' = f_i(t, y_{\tau(1)}, \ldots, y_{\tau(\mathrm{M})}), \qquad i = \tau\,(1), \ldots, \tau\,(\mathrm{M}). \tag{16}
$$

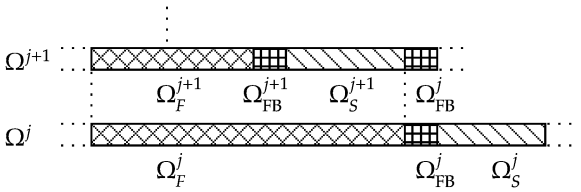
The M subsystems have different characteristic time scales with $y_{\tau(1)}$ being the slowest and $y_{\tau(\mathrm{M})}$ the fastest components. Typically, a small number of fast changing components (or a small number of grid points on the fine grids) restrict the overall timestep of the integration.

3.1 Domain Partitioning

The domain is partitioned into subdomains, with each subdomain being characterized by a specific time scale. In multirate time integration, different steps are taken on each subsystem such that the numerical method satisfies global accuracy and stability properties. In this context, the transition among subdomains needs special attention to preserve these properties. Consequently, in this study we consider a nested domain decomposition of y defined in (16). First, we split the domain into two partitions: a slow partition and a fast partition that are



(a) Fast and slow solution partitioning



(b) The layout of the nested domain decomposition

Fig. 1 Illustration of the nested domain decomposition into subdomains that are characterized by different timescales

separated by a fast buffer region to accommodate the transition. Formally, the domain Ω is decomposed as

$$\Omega = \Omega^0 = \Omega_F^0 \cup \Omega_{\mathrm{FB}}^0 \cup \Omega_S^0, \tag{17}$$

where we consider an associated "slow" (S) characteristic time for the slow subdomain (Ω_S), a "fast" (F) characteristic time for the fast subdomain (Ω_F), and a fast buffer (Ω_{FB}) that bridges the transition between them. The need for this buffer region will become apparent when the SSP properties are analyzed later in the paper. We refer to Ω's superscript as the "level" of the grid and denote with $\mathfrak{m}$ the ratio between the timesteps associated with the fast subdomain and the slow subdomain on the same level (e.g., j). The timestep on the next level (i.e., $j+1$) associated with the slow subdomain is the same as the one associated with the fast subdomain on the current level.

Each solution component in (16) corresponds to variables in a particular subdomain. Adjacent subdomains have adjacent time scales and the subdomains have buffers between them to accommodate the time scale transition. The characteristic time scale on the fast buffer region corresponds to the time scale of the slow domain; however, the solution is obtained with the small timestep used in the fast region:

$$\begin{aligned} y'_{\tau(j)} &= (y_S)' = f\left(t, y(x)\right), \quad x \in \left\{\Omega_S^j\right\}, \\ y'_{\tau(j+1)} &= (y_F)' = f\left(t, y(x)\right), \quad x \in \left\{\Omega_F^j \cup \Omega_{\mathrm{FB}}^j\right\}, \ j \geq 0. \end{aligned} \tag{18}$$

In Fig. 1a we illustrate this aspect. The variables y_F on $\Omega_F^j \cup \Omega_{\mathrm{FB}}^j$ are fast (evolving) and integrated with a small timestep. The variables y_S on Ω_S^j are slow and are integrated with a large timestep.

The partitioning procedure can be extended recursively with $\Omega_F^j = \Omega^{j+1}$ until the solution characteristic time requirements for each component $y_{\tau(1)\cdots\tau(\mathrm{M})}$ are met:

$$\Omega_F^j = \Omega^{j+1} = \Omega_F^{j+1} \cup \Omega_{\mathrm{FB}}^{j+1} \cup \Omega_S^{j+1}. \tag{19}$$

An illustration of the nested domain decomposition is shown in Fig. 1b.

We note that the nested partitioning (19) decouples the estimation of f exemplified in (16) in

$$\begin{aligned} (y_S)' &= f\,(t, y_{\mathrm{FB}}, y_S)\,, \\ (y_F)' &= f\,(t, y_F, y_{\mathrm{FB}})\,, \qquad j \geq 0. \end{aligned} \tag{20}$$

In this manner, an efficient domain partitioning with an associated characteristic time that satisfies the accuracy and stability requirements of the solution in the corresponding partition is achieved.

Next, we discuss the time integration method applied on different partitions in detail.

3.2 Partitioned Runge–Kutta Methods

Consider a system of ODEs which allows an explicit separation of the fast and the slow components

$$\begin{pmatrix} y_F \\ y_S \end{pmatrix}' = \begin{pmatrix} f_F\,(y_F, y_S) \\ f_S\,(y_F, y_S) \end{pmatrix} \tag{21}$$

Partitioned Runge–Kutta (PRK) schemes [22, 24] are used to solve the problem (21) with two different methods, $\mathcal{RK}^F = [A^F, b^F, c^F]$ for the fast part, and $\mathcal{RK}^S = [A^S, b^S, c^S]$ for the slow part. The PRK solution method reads

$$\begin{aligned} y_F^{n+1} &= y_F^n + \Delta t \sum_{i=1}^{s} b_i^F K_F^j, & y_S^{n+1} &= y_S^n + \Delta t \sum_{i=1}^{s} b_i^S K_S^j, \\ Y_F^i &= y_F^n + \Delta t \sum_{j=1}^{s} a_{ij}^F K_F^j, & Y_S^i &= y_S^n + \Delta t \sum_{j=1}^{s} a_{ij}^S K_S^j, \\ K_F^i &= f_F\left(Y_F^i, Y_S^i\right), & K_S^i &= f_S\left(Y_F^i, Y_S^i\right). \end{aligned} \tag{22}$$

The order of the coupled method is the minimum among the orders of the fast and slow methods and the order of their "coupling". The first order coupling conditions ($\sum_i b_i^F = \sum_i b_i^S = 1$) are implicitly satisfied since each method is at least first order. The second order coupling conditions [24, p. 308] are

$$\sum_{i=1}^{s} b_i^F c_i^S = \frac{1}{2}, \qquad \sum_{i=1}^{s} b_i^S c_i^F = \frac{1}{2}. \tag{23}$$

3.3 Strong Stability Preservation and PRK

We seek to construct PRK methods with the SSP property applied to a partitioning of type (20). In this section we infer intuitively some conditions for PRK methods to be SSP, while a rigorous treatment of this property is addressed later in the paper. A necessary condition for PRK to be SSP is that both $\mathcal{RK}^F$ and $\mathcal{RK}^S$ are SSP methods. A natural question is how to treat the interface region, Ω_{FB}.

Equation (20) shows the inter-dependency of the flux function on the solution in adjacent partitions. By construction, the flux function f evaluated on Ω_S depends on part of the solution on Ω_S and on part on the fast buffer Ω_{FB}. Also by construction, $\mathcal{RK}^S$ applied on $\Omega_S \cup \Omega_{\mathrm{FB}}$ is SSP and $\mathcal{RK}^F$ is SSP with the solution on $\Omega_{\mathrm{FB}} \cup \Omega_F$. In the next section we investigate the properties of a proposed PRK method formed by $\mathcal{RK}^S$ and $\mathcal{RK}^F$, and rigorously analyze the interface between them.

Table 1 Order two multirate partitioned Runge–Kutta method (MPRK-2)

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

(a) Base ($\mathcal{RK}^B$)

$$\begin{array}{c|cccc} \frac{1}{\mathfrak{m}}c & \frac{1}{\mathfrak{m}}A & & & \\ \frac{1}{\mathfrak{m}}\mathbb{1}+\frac{1}{\mathfrak{m}}c & \frac{1}{\mathfrak{m}}\mathbb{1}b^T & \frac{1}{\mathfrak{m}}A & & \\ \vdots & \vdots & \ddots & \ddots & \\ \frac{m-1}{\mathfrak{m}}\mathbb{1}+\frac{1}{\mathfrak{m}}c & \frac{1}{\mathfrak{m}}\mathbb{1}b^T & \cdots & \frac{1}{\mathfrak{m}}\mathbb{1}b^T & \frac{1}{\mathfrak{m}}A \\ \hline & \frac{1}{\mathfrak{m}}b^T & \frac{1}{\mathfrak{m}}b^T & \cdots & \frac{1}{\mathfrak{m}}b^T \end{array}$$

(b) Fast method ($\mathcal{RK}^F$)

$$\begin{array}{c|cccc} c & A & & & \\ c & & A & & \\ \vdots & & & \ddots & \\ c & & & & A \\ \hline & \frac{1}{\mathfrak{m}}b^T & \frac{1}{\mathfrak{m}}b^T & \cdots & \frac{1}{\mathfrak{m}}b^T \end{array}$$

(c) Slow method ($\mathcal{RK}^S$)

To summarize, we consider the partitioned Runge–Kutta method applied in our context (20) with both methods ($\mathcal{RK}^S$ and $\mathcal{RK}^F$) being SSP on their respective partition. $\mathcal{RK}^S$ is applied on Ω_S and $\mathcal{RK}^F$ is applied on $\Omega_{\text{FB}} \cup \Omega_F$.

Additional requirements (linear stability, TVD, TVB, positivity) of the full method need to be satisfied by each pair of spatial and temporal discretization. Our experiments indicate that multirate schemes constructed based on SSP time discretizations preserve the particular stability features of the spatial discretization. We shall further discuss this in the following sections.

3.4 A Second Order Multirate PRK Family

Based on the PRK setting and discussion from Sects. 3.2, 3.3, we propose the following generic family of second order multirate partitioned Runge–Kutta (MPRK) scheme. We denote this scheme with MPRK-2. Consider a second order accurate SSP RK "base" method (RK2a, for instance) $\mathcal{RK}^B = [A, b, c]$ (Table 1a). Using this base method, we extend $\mathcal{RK}^B$ to the fast ($\mathcal{RK}^F$) and slow ($\mathcal{RK}^S$) methods in the manner shown in Table 1. Here we denote a vector of ones with $\mathbb{1}$. Note that the fast and slow methods have the same weight coefficients b, $b^F = b^S$ ($1/\mathfrak{m}b$ repeated $\mathfrak{m}$ times). The reason for this choice will become clear when we discuss the conservation properties of the full method.

Note that the slow method repeats the same stages $\mathfrak{m}$ times. The fast method takes $\mathfrak{m}$ successive steps of length $\Delta t/\mathfrak{m}$ with $\mathcal{RK}^B$. The slow method takes one step of length Δt with $\mathcal{RK}^B$; this step is formally repeated $\mathfrak{m}$ times in the Butcher tableau.

The method presented in this section represents a truly multirate approach since the fast method takes $\mathfrak{m}$ successive steps of the base method with a timestep of $\Delta t/\mathfrak{m}$. This method can be easily extended from $\mathfrak{m} = 2$ to arbitrary $\mathfrak{m}$'s. In Appendix 1 we present the same method for $\mathfrak{m} = 3$.

Proposition 3.1 (MPRK-2) *The partitioned Runge–Kutta methods defined by the Butcher tableau in Table 1*:

(a) *Are second order accurate if the base method* (*Table* 1*a*) *is at least second order accurate, and*
(b) *Have at most second order accurate coupling regardless of the order of the base method*

Proof (a) First, we check the order conditions for each method separately, considering that the s-stage base method is second order. The first order conditions for ($\mathcal{RK}^S$ and $\mathcal{RK}^F$) are verified since by (10) we have

$$\sum_{i=1}^{\mathrm{m}\times s} b_i^S = \mathrm{m}\sum_{j=1}^{s}\frac{b_i}{\mathrm{m}} = 1, \qquad \sum_{i=1}^{\mathrm{m}\times s} b_i^F = \mathrm{m}\sum_{j=1}^{s}\frac{b_i}{\mathrm{m}} = 1.$$

The second order conditions (11) are also satisfied for the slow method ($\mathcal{RK}^S$)

$$\sum_{i=1}^{\mathrm{m}\times s} b_i^S c_i^S = \sum_{i=1}^{s}\frac{{b_i}^T}{\mathrm{m}} c_i = \frac{1}{\mathrm{m}}\frac{\mathrm{m}}{2} = \frac{1}{2},$$

and for the fast method ($\mathcal{RK}^F$)

$$\begin{aligned}\sum_{i=1}^{\mathrm{m}\times s} b_i^F c_i^F &= \frac{1}{\mathrm{m}^2}\left(b^T c + b^T(\mathbb{1}+c) + \cdots + b^T((\mathrm{m}-1)\mathbb{1}+c)\right)\\ &= \frac{1}{\mathrm{m}^2} b^T\left(\mathrm{m}c + \left(\sum_{i=0}^{\mathrm{m}-1} i\right)\mathbb{1}\right) = \frac{1}{\mathrm{m}^2}\left(\mathrm{m}b^T c + \frac{\mathrm{m}(\mathrm{m}-1)}{2}\, b^T\mathbb{1}\right)\\ &= \frac{1}{\mathrm{m}^2}\left(\frac{\mathrm{m}}{2} + \frac{\mathrm{m}(\mathrm{m}-1)}{2}\right) = \frac{1}{2}.\end{aligned}$$

Since $b^F = b^S$, the second order coupling conditions (23) are satisfied directly by the above. Hence, MPRK-2 is at least second order accurate.

(b) There are over 20 third order coupling conditions that can be found in [24]. Here we list two that are contradicted by MPRK-2-like schemes

$$\sum_{i=1}^{s} b_i^F c_i^F c_i^S = \frac{1}{3}, \qquad \sum_{i=1}^{s} b_i^S c_i^S c_i^F = \frac{1}{3}. \tag{24}$$

Consider that $\mathcal{RK}^B$ satisfies the third order accuracy conditions. The third order coupling condition (24) requires the following

$$\begin{aligned}(b^F)^T c^F c^S &= \frac{1}{\mathrm{m}^2}\left(b^T c^2 + b^T\left(c+c^2\right) + \cdots + b^T\left((\mathrm{m}-1)c + c^2\right)\right)\\ &= \frac{1}{\mathrm{m}^2} b^T\left(\mathrm{m}c^2 + \left(\sum_{i=0}^{\mathrm{m}-1} i\right)c\right) = \frac{1}{\mathrm{m}^2}\left(\mathrm{m}b^T c^2 + \frac{\mathrm{m}(\mathrm{m}-1)}{2}\, b^T c\right)\\ &= \frac{1}{\mathrm{m}^2}\left(\frac{\mathrm{m}}{3} + \frac{\mathrm{m}(\mathrm{m}-1)}{4}\right) = \frac{1}{4} + \frac{1}{12\mathrm{m}},\\ (b^F)^T c^F c^S &= \frac{1}{3} \quad \Rightarrow \quad \mathrm{m} = 1\end{aligned}$$

and thus, (at the interface) the coupling reduces to second order accuracy for $\mathrm{m} > 1$. □

To increase the coupling order we need to investigate other schemes that use different base methods for the fast and for the slow subsystems and have different couplings. Such methods will be investigated in future studies.

Proposition 3.2 (Conservation) *Any partitioned Runge–Kutta method with the same fast and slow weights ($b^F = b^S$) is conservative. In particular MPRK-2 (described by the Butcher tableau 1) is conservative.*

Proof (a) First we consider the preservation of linear invariants. This is a direct consequence of having chosen equal weights for the fast and for the slow methods, $b^F = b^S$ and of the fact that the slow and fast functions are evaluated with the same arguments (also see [45]). Consider the system (21) with a linear invariant of the form

$$e_F^T f_F(y_F, y_S) + e_S^T f_S(y_F, y_S) = 0 \quad \forall y_F, y_S \quad \Rightarrow \quad e_F^T y_F(t) + e_S^T y_S(t) = \text{const} \quad \forall t, \tag{25}$$

where e_F, e_S are fixed weight vectors.

From the method (22) with $b^F = b^S = b^*$ we have that

$$y_F^{n+1} = y_F^n + \Delta t \sum_{i=1}^{s} b_i^* f_F\left(Y_F^i, Y_S^i\right), \qquad y_S^{n+1} = y_S^n + \Delta t \sum_{i=1}^{s} b_i^* f_S\left(Y_F^i, Y_S^i\right)$$

and therefore

$$\begin{aligned} e_F^T y_F^{n+1} + e_S^T y_S^{n+1} &= e_F^T y_F^n + e_S^T y_S^n + \Delta t \sum_{i=1}^{s} b_i^* \underbrace{\left(e_F^T f_F\left(Y_F^i, Y_S^i\right) + e_S^T f_S\left(Y_F^i, Y_S^i\right)\right)}_{0} \\ &= e_F^T y_F^n + e_S^T y_S^n. \end{aligned}$$

(b) Next we consider the conservation for hyperbolic PDEs. This property is important because multirate Runge–Kutta methods, used in conjunction with conservative space discretizations, lead to conservative full discretizations of the PDE. Consider a one-dimensional finite volume scheme (in the conservative formulation):

$$y_i' = \frac{1}{\Delta x_i}\left(F_{i-\frac{1}{2}}(y) - F_{i+\frac{1}{2}}(y)\right), \quad 1 \le i \le N,$$

where $F_{i+\frac{1}{2}}$ is the numerical flux through the $i+\frac{1}{2}$ interface. Assuming no fluxes through the leftmost and the rightmost boundaries ($F_{\frac{1}{2}} = F_{N+\frac{1}{2}} = 0$), the finite volume discretization is conservative in the sense that

$$\left(\sum_{i=1}^{N} \Delta x_i y_i\right)' = \sum_{i=1}^{N} \Delta x_i y_i' = \sum_{i}\left(F_{i-\frac{1}{2}}(y) - F_{i+\frac{1}{2}}(y)\right) = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \Delta x_i y_i = \text{const}.$$

The time discretization with a classical (single-rate) Runge Kutta method gives a conservative fully discrete method. We want to show that the multirate method is also conservative. For this, assume that the leftmost $\ell - 1$ grid cells are the fast domain, and the remaining cells are the slow domain:

$$y_F = \{y_1, \ldots, y_{\ell-1}\}, \qquad y_S = \{y_\ell, \ldots, y_N\}. \tag{26}$$

The $\ell - 1$ interface separates the fast and the slow domains. Each subdomain is advanced in time with a classical Runge–Kutta method, therefore the fluxes exchanged between the

Table 2 Order 2 Butcher tableau for the **a** RK2a base method, and the **b** fast, and **c** slow methods for $\mathfrak{m} = 2$

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

(a) Base method

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

(b) Fast method

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

(c) Slow method

boundaries of same domain cells are conserved. The question remains whether the fluxes crossing the fast-slow interface are conserved. We now show that the total flux lost by the fast domain through the fast-slow interface is exactly the total flux received by the slow domain through the same interface. From the multirate Runge–Kutta formula it follows that

$$y^n_{\ell-1} = y^n_{\ell-1} + \frac{\Delta t}{\Delta x_{\ell-1}} \sum_{i=1}^{s} b^F_i \left(F_{\ell-\frac{3}{2}}(Y^i) - F_{\ell-\frac{1}{2}}(Y^i) \right), \tag{27}$$

the total flux lost by the fast domain during one full timestep through the fast-slow interface is

$$\Delta t \sum_{i=1}^{s} b^F_i F_{\ell-\frac{1}{2}}(Y^i). \tag{28}$$

Similarly,

$$y^{n+1}_{\ell} = y^n_{\ell} + \frac{\Delta t}{\Delta x_{\ell}} \sum_{i=1}^{s} b^S_i \left(F_{\ell-\frac{1}{2}}(Y^i) - F_{\ell+\frac{1}{2}}(Y^i) \right), \tag{29}$$

and the total flux received by the slow domain during one timestep through the fast-slow interface is

$$\Delta t \sum_{i=1}^{s} b^S_i F_{\ell-\frac{1}{2}}(Y^i). \tag{30}$$

At each stage i of the multirate formula the flux functions are evaluated at the same argument values, Y^i. Therefore, a sufficient condition to have conservation of the flux through the fast-slow interface is that the fast and slow method weights are equal to each other, $b^S_i = b^F_i$.

Finally, MPRK-2 has the same coefficients ($b^S_i = b^F_i$) by construction and hence is conservative. □

We note that conservation is achieved without explicitly storing fluxes at the fast-slow interface, as it was proposed in earlier works [30].

3.5 A Second Order SSP PRK Method with $\mathfrak{m} = 2$

In this section we consider the SSP Runge–Kutta RK2a [48] as the base method in Table 2a, and extend it to a multirate method with $\mathfrak{m} = 2$ using the approach described in Sect. 3.4.

The Butcher tableau for the fast and slow methods for $\mathfrak{m} = 2$ are shown in Table 2; together they form a partitioned RK method. The RK stages are computed as follows:

$$
\begin{aligned}
K_F^1 &= f_F(y_F^n, y_S^n), & K_S^1 &= f_S(y_F^n, y_S^n),\\
y_F^{(1)} &= y_F^n + \frac{\Delta t}{2} K_F^1, & y_S^{(1)} &= y_S^n + \Delta t K_S^1,\\
K_F^2 &= f_F(y_F^{(1)}, y_S^{(1)}), & K_S^2 &= f_S(y_F^{(1)}, y_S^{(1)}),\\
y_F^{(2)} &= y_F^n + \frac{\Delta t}{4} K_F^1 + \frac{\Delta t}{4} K_F^2, & y_S^{(2)} &= y_S^n,\\
K_F^3 &= f_F(y_F^{(2)}, y_S^n), & K_S^3 &= f_S(y_F^{(2)}, y_S^n),\\
y_F^{(3)} &= y_F^{(2)} + \frac{\Delta t}{2} K_F^3, & y_S^{(3)} &= y_S^n + \Delta t K_S^3,\\
K_F^4 &= f_F(y_F^{(3)}, y_S^{(3)}), & K_S^4 &= f_S(y_F^{(3)}, y_S^{(3)}),\\
y_F^{n+1} &= y_F^n + \frac{\Delta t}{4}(K_F^1 + K_F^2 + K_F^3 + K_F^4), & y_S^{n+1} &= y_S^n + \frac{\Delta t}{4}(K_S^1 + K_S^2 + K_S^3 + K_S^4).
\end{aligned}
\tag{31}
$$

Using the following notation to compactly denote Euler steps

$$
\mathcal{E}_{\{F,S\}}(\Delta t, y_F, y_S) := y_{\{F,S\}}(t) + \Delta t \cdot f_{\{F,S\}}(t, y_F, y_S), \tag{32}
$$

the MPRK-2 scheme can be written as convex combinations of Euler steps in the following way:

$$
\begin{aligned}
y_F^{n+1} &= \frac{1}{2}\left(y_F^n + y_F^n + \frac{\Delta t}{2} K_F^1 + \frac{\Delta t}{2} K_F^2 \right. & y_F^{(1)} &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^n, y_S^n\right),\\
&\quad \left. + \frac{\Delta t}{2} K_F^3 + \frac{\Delta t}{2} K_F^4\right), & &\\
&= \frac{1}{2}\left(y_F^n + y_F^n + \frac{\Delta t}{2} K_F^2 + \frac{\Delta t}{2} K_F^3 + \frac{\Delta t}{2} K_F^4\right), & y_F^{(1*)} &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^{(1)}, y_S^{(1)}\right),\\
&= \frac{1}{2}\left(y_F^n + y_F^{(1*)} + \frac{\Delta t}{2} K_F^3 + \frac{\Delta t}{2} K_F^4\right), & y_F^{(2)} &= \frac{1}{2}(y_F^n + y_F^{(1*)}),\\
&= \frac{1}{2}\left(y_F^{(2)} + y_F^{(3)} + \frac{\Delta t}{2} K_F^4\right), & y_F^{(3)} &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^{(2)}, y_S^n\right),\\
&= \frac{1}{2}(y_F^{(2)} + y_F^{(3*)}), & y_F^{(3*)} &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^{(3)}, y_S^{(3)}\right)
\end{aligned}
\tag{33}
$$

and

$$
\begin{aligned}
y_S^{n+1} &= \frac{1}{4}\Big(2y_S^n + y_S^n + \Delta t K_S^1 + \Delta t K_S^2 + y_S^n & y_S^{\{(1),(3)\}} &= \mathcal{E}_S\left(\Delta t, y_F^{\{n,(2)\}}, y_S^n\right),\\
&\quad + \Delta t K_S^3 + \Delta t K_S^4\Big), & &\\
&= \frac{1}{4}\Big(2y_S^n + y_S^{(1)} + \Delta t K_S^2 & y_S^{\{(1*),(3*)\}} &= \mathcal{E}_S\left(\Delta t, y_F^{\{(1),(3)\}}, y_S^{\{(1),(3)\}}\right),\\
&\quad + y_S^{(3)} + \Delta t K_S^4\Big), & &\\
&= \frac{1}{4}\Big(2y_S^n + y_S^{(1*)} + y_S^{(3*)}\Big). & &
\end{aligned}
$$

Table 3 MPRK-2 Euler steps for the fast and slow methods for $\mathfrak{m} = 2$. The slow buffer represents a small region where the solution depends on both fast and slow partitions

Fast method ($\mathcal{RK}^F$)	Slow method ($\mathcal{RK}^S$) in slow buffer	Slow method ($\mathcal{RK}^B$) in slow region
y_F^n	y_S^n	y_S^n
$y_F^{(1)} = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^n, y_S^n\right)$	$y_S^{(1)} = \mathcal{E}_S\left(\Delta t, y_F^n, y_S^n\right)$	$y_S^{(1)} = \mathcal{E}\left(\Delta t, y_S^n\right)$
$y_F^{(1*)} = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^{(1)}, y_S^{(1)}\right)$	$y_S^{(1*)} = \mathcal{E}_S\left(\Delta t, y_F^{(1)}, y_S^{(1)}\right)$	$y_S^{(1*)} = \mathcal{E}\left(\Delta t, y_S^{(1)}\right)$
$y_F^{(2)} = \frac{1}{2}\left(y_F^n + y_F^{(1*)}\right)$	$y_S^{(2)} = y_S^n$	
$y_F^{(3)} = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^{(2)}, y_S^n\right)$	$y_S^{(3)} = \mathcal{E}_S\left(\Delta t, y_F^{(2)}, y_S^n\right)$	$y_S^{(3)} = y_S^{(1)}$
$y_F^{(3*)} = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^{(3)}, y_S^{(3)}\right)$	$y_S^{(3*)} = \mathcal{E}_S\left(\Delta t, y_F^{(3)}, y_S^{(3)}\right)$	$y_S^{(3*)} = y_S^{(1*)}$
$y_F^{n+1} = \frac{1}{2}\left(y_F^{(2)} + y_F^{(3*)}\right)$	$y_S^{n+1} = \frac{1}{2}y_S^n + \frac{1}{4}y_S^{(1*)} + \frac{1}{4}y_S^{(3*)}$	$y_S^{n+1} = \frac{1}{2}\left(y_S^n + y_S^{(1*)}\right)$

(34)

The sequence of fast and slow Euler steps and their convex combination that constitute MPRK-2 are summarized in Table 3. We now show that this timestepping method preserves the maximum principle and nonlinear stability properties of the discretization. Kirby [30] has carried out a similar analysis for the multirate explicit Euler method.

Proposition 3.3 (Positivity) *If each fast multirate Euler step*

$$y_F^{(n+\frac{\Delta t}{\mathfrak{m}})} = \mathcal{E}_F\left(\frac{\Delta t}{\mathfrak{m}}, y_F^{(n)}, y_S^{(n)}\right)$$

and each slow multirate Euler step

$$y_S^{(n+\Delta t)} = \mathcal{E}_S\left(\Delta t, y_F^{(n)}, y_S^{(n)}\right)$$

preserves positivity properties under a local CFL condition, then the multirate solution also preserves the positivity.

Proof The method is constructed using convex combinations of forward Euler solutions. Each Euler step is positivity preserving. More precisely, if we have $y_F^n \geq 0$ and $y_S^n \geq 0$, by applying Euler steps, it follows that the first macrostep of MPRK-2 (see Table 3) is positivity preserving:

$$\left.\begin{aligned} \left\{y_F^{(1)}, y_S^{(1)}\right\} \geq 0 \Rightarrow \left\{y_F^{(1*)}, y_S^{(1*)}\right\} \geq 0 \\ y_F^{(2)} = \frac{1}{2}\left(y_F^n + y_F^{(1*)}\right) \geq 0 \end{aligned}\right\} \Rightarrow \left\{y_F^{(2)}, y_S^n\right\} \geq 0. \tag{35}$$

Using (35) and following the same rationale, we have that all intermediate solution components are positive:

$$\left.\begin{aligned} \left\{y_F^{(3)}, y_S^{(3)}\right\} \geq 0 \Rightarrow \left\{y_F^{(3*)}, y_S^{(3*)}\right\} \geq 0 \\ y_F^{n+1} = \tfrac{1}{2}\left(y_F^{(2)} + y_F^{(3*)}\right) \geq 0 \\ y_S^{n+1} = \tfrac{1}{4}\left(y_S^n + y_S^{(1*)}\right) + \tfrac{1}{4}\left(y_S^n + y_S^{(3*)}\right) \end{aligned}\right\} \Rightarrow \left\{y_F^{n+1}, y_S^{n+1}\right\} \geq 0, \tag{36}$$

and hence the full method is positivity preserving. □

Note that intermediate stage solutions of mixed forward Euler steps are not consistent solutions of the PDE, since the fast subsystem and the slow subsystem are advanced with different timesteps, and the intermediate solutions are at different intermediate times. For example $\{y_F^{(1)}, y_S^{(1)}\}$ are solutions at $\Delta t/2$ for the fast component and Δt for the slow one. We call such a step a "mixed Euler step".

Proposition 3.4 (Maximum principle) *If each fast and each slow multirate Euler step satisfies the maximum principle then MPRK-2 also satisfies the maximum principle.*

Proof Based again on the properties of the forward Euler method, a quick inspection of Table 3 shows the following

$$\begin{aligned} &\max\{y_F^{(1)}, y_S^{(1)}\} \leq \max\{y_F^n, y_S^n\}, && \max\{y_F^{(1*)}, y_S^{(1*)}\} \leq \max\{y_F^{(1)}, y_S^{(1)}\}, \\ &\max\{y_F^{(2)}, y_S^n\} \leq \max\{y_F^n, y_S^n\}, \\ &\max\{y_F^{(3)}, y_S^{(3)}\} \leq \max\{y_F^{(2)}, y_S^n\}, && \max\{y_F^{(3*)}, y_S^{(3*)}\} \leq \max\{y_F^{(3)}, y_S^{(3)}\}, \end{aligned}$$

and thus, clearly $\max\{y_F^{n+1}, y_S^{n+1}\} \leq \max\{y_F^n, y_S^n\}$. Similarly, using the forward Euler properties we have $\min\{y_F^{n+1}, y_S^{n+1}\} \geq \min\{y_F^n, y_S^n\}$ □

For the next properties we consider finite volume (conservative) spatial discretizations (2). Specifically, we consider discretizations which assume the following form (when forward Euler timesteps are used)

$$\begin{aligned} &y_j^{n+\Delta t} = y_j^n + \frac{\Delta t}{\Delta x}\left(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\right) = y_j^n + \frac{\Delta t}{\Delta x}\left(C_{j+\frac{1}{2}}\Delta_+ y_j^n - D_{j-\frac{1}{2}}\Delta_- y_j^n\right), \\ &\Delta_+ y_j^n = y_{j+1}^n - y_j^n, \Delta_- y_j^n = y_j^n - y_{j-1}^n, \end{aligned} \tag{37}$$

where C, D are the spatial discretization coefficients that depend on y. This is the framework in which most of the total variation related properties are studied [7, 39]. The following lemma is due to Harten [25].

Lemma 3.5 (Harten [25] (2.1)) *If coefficients C and D satisfy the following inequalities*

$$C_{j+\frac{1}{2}} \geq 0, \; D_{j+\frac{1}{2}} \geq 0 \quad \textit{and} \tag{38}$$

$$1 - \frac{\Delta t}{\Delta x}\left(C_{j+\frac{1}{2}} + D_{j+\frac{1}{2}}\right) \geq 0 \tag{39}$$

then scheme (37) is TV non increasing or TVD.

Now consider a discrete solution y^n on a bounded domain, Ω, partitioned in two subdomains (26), and assume that the scheme (37) with a grid ratio of $\Delta t/(\mathfrak{m}\Delta x)$, $\mathfrak{m} \geq 1$ ("fast"

method) satisfies the TVD conditions (38, 39) for every point in Ω. Moreover, consider that the same scheme verifies the TVD conditions for a grid ratio of $\Delta t/\Delta x$ ("slow" method) for every point of the domain Ω that has the index larger or equal to $\ell - L$, where $L > 1$ is the spatial domain of dependency (stencil). More precisely, the TVD conditions (38, 39) are satisfied by the scheme (37) for the following timesteps:

$$\frac{\Delta t}{\mathfrak{m}} \quad \text{for } j \le \ell - L - 1, \tag{40}$$

$$\Delta t \quad \text{for } j \ge \ell - L. \tag{41}$$

These are essentially the local CFL conditions.

The solution is advanced in time using scheme (37) with a timestep Δt on the slow domain ($j \ge \ell$), and with $\Delta t/\mathfrak{m}$ on the fast domain ($j \le \ell - 1$):

$$y_j^{n+\Delta t} = y_j^n + \frac{\Delta t}{\Delta x}\left(C_{j+\frac{1}{2}}\Delta_+ y_j^n - D_{j-\frac{1}{2}}\Delta_- y_j^n\right), \qquad j \ge \ell, \quad \text{(slow)}, \tag{42}$$

$$y_j^{n+\frac{\Delta t}{\mathfrak{m}}} = y_j^n + \frac{\Delta t}{\mathfrak{m}\Delta x}\left(C_{j+\frac{1}{2}}\Delta_+ y_j^n - D_{j-\frac{1}{2}}\Delta_- y_j^n\right), \quad j \le \ell - 1, \quad \text{(fast)}. \tag{43}$$

Obviously, the TVD conditions (38, 39) are satisfied under the local CFL restriction. In this case, we consider the region between ℓ and $\ell - L$ as the "fast buffer" where even if the TVD conditions hold for a large timestep Δt, the solution is advanced with the small step $\Delta t/\mathfrak{m}$.

The following lemma states that mixed forward Euler steps (42, 43) do not increase the TV of the solution.

Lemma 3.6 (TV for mixed Euler step) *Consider the scheme* (37) *that satisfies the TVD conditions* (38, 39) *for timesteps restricted by* (40, 41). *Then the mixed forward Euler timesteps defined by* (42, 43) *do not increase the total variation of the solution.*

Proof The forward Euler steps (42) and (43) can be written as

$$y_j^{n+\Delta t/\xi} = y_j^n + \frac{\Delta t}{\mathfrak{m}\Delta x}\left(\widetilde{C}_{j+\frac{1}{2}}\Delta_+ y_j^n - \widetilde{D}_{j-\frac{1}{2}}\Delta_- y_j^n\right), \quad j \in \Omega, \tag{44}$$

where

$$\widetilde{C}_{j+\frac{1}{2}} = \begin{cases} C_{j+\frac{1}{2}} & \text{if } j \le \ell - 1, \\ \mathfrak{m}C_{j+\frac{1}{2}} & \text{if } j \ge \ell, \end{cases} \qquad \widetilde{D}_{j+\frac{1}{2}} = \begin{cases} D_{j+\frac{1}{2}} & \text{if } j \le \ell - 2, \\ \mathfrak{m}D_{j+\frac{1}{2}} & \text{if } j \ge \ell - 1, \end{cases}$$

$$\xi = \begin{cases} \mathfrak{m} & \text{if } j \le \ell - 1, \\ 1 & \text{if } j \ge \ell. \end{cases}$$

Since the TVD conditions are satisfied under (40, 41), the TVD condition (38) is also satisfied:

$$C_{j+\frac{1}{2}}, D_{j+\frac{1}{2}} \ge 0, \ j \in \Omega \quad \Rightarrow \quad \widetilde{C}_{j+\frac{1}{2}}, \widetilde{D}_{j+\frac{1}{2}} \ge 0, \ \forall j \in \Omega. \tag{45}$$

The TVD condition (39) for (44) is

$$1 - \frac{\Delta t}{\mathfrak{m}\Delta x}\left(\widetilde{C}_{j+\frac{1}{2}} + \widetilde{D}_{j+\frac{1}{2}}\right) \ge 0, \quad j \in \Omega, \tag{46}$$

and in detail we have

$$1-\frac{\Delta t}{\mathfrak{m}\Delta x}\left(C_{j+\frac{1}{2}}+D_{j+\frac{1}{2}}\right)\geq 0,\qquad j\leq \ell-2, \tag{47}$$

$$1-\frac{\Delta t}{\mathfrak{m}\Delta x}\left(C_{j+\frac{1}{2}}+\mathfrak{m}D_{j+\frac{1}{2}}\right)\geq 0,\qquad j=\ell-1, \tag{48}$$

$$1-\frac{\Delta t}{\mathfrak{m}\Delta x}\left(\mathfrak{m}C_{j+\frac{1}{2}}+\mathfrak{m}D_{j+\frac{1}{2}}\right)\geq 0,\quad j\geq \ell. \tag{49}$$

The TVD condition (47) applies to the fast method in the fast domain and is satisfied for a grid ratio of $\Delta t/(\mathfrak{m}\Delta x)$ by (40). Condition (49) applies strictly to the slow method and is locally satisfied for $\Delta t/\Delta x$ by (41).

The condition (48) involves the mixed slow/fast flux approximations on the interface region. By the fact that the slow method satisfies the TVD conditions on the fast buffer region (41), the following condition

$$1-\frac{\Delta t}{\mathfrak{m}\Delta x}\left(\mathfrak{m}C_{j+\frac{1}{2}}+\mathfrak{m}D_{j+\frac{1}{2}}\right)\geq 0,\quad y_{j=\ell-L}, \tag{50}$$

holds. It follows that condition (48) is also satisfied due to the fact that $C_{j+\frac{1}{2}}$ and $D_{j+\frac{1}{2}}$ are positive and $\mathfrak{m}\geq 1$:

$$\begin{aligned}1-\frac{\Delta t}{\mathfrak{m}\Delta x}\left(C_{j+\frac{1}{2}}+\mathfrak{m}D_{j+\frac{1}{2}}\right)&\geq 1-\frac{\Delta t}{\mathfrak{m}\Delta x}\left(\mathfrak{m}C_{j+\frac{1}{2}}+\mathfrak{m}D_{j+\frac{1}{2}}\right)\geq 0,\\ &\geq 1-\frac{\Delta t}{\Delta x}\left(C_{j+\frac{1}{2}}+D_{j+\frac{1}{2}}\right)\geq 0,\quad j=\ell-1.\end{aligned} \tag{51}$$

Therefore all TVD conditions (45, 46) are satisfied for the mixed scheme (42, 43). □

The MPRK-2 scheme uses convex combinations of mixed forward Euler timesteps. We showed that the total variation of solutions of mixed Euler steps is not increased. Now we investigate the total variation of convex combinations of solutions of forward Euler steps.

We continue by following the traditional framework for the stability analysis of hyperbolic conservation laws [46–48]. Let $y^n\in\Omega$ be a smooth initial solution and y^A, $y^B\in\Omega$ two solutions obtained through a finite sequence (of $\mathcal{O}(\Delta t)$) time integrations starting with y^0 using the forward Euler method. Moreover, consider that all the steps satisfy the CFL condition and the solution remains smooth throughout its integration steps. Using the forward Euler method properties, we have the following:

$$\begin{cases}\mathrm{TV}\left(\{y^A\}\right)\leq \mathrm{TV}\left(\{y^0\}\right),\\ \mathrm{TV}\left(\{y^B\}\right)\leq \mathrm{TV}\left(\{y^0\}\right),\end{cases}\quad\text{and}\quad\begin{cases}\|y^A-y^0\|_\infty\approx\mathcal{O}(\Delta t),\\ \|y^B-y^0\|_\infty\approx\mathcal{O}(\Delta t).\end{cases} \tag{52}$$

Consider the mixed convex combinations of solutions y^A and y^B:

$$\left\{\alpha y^A_{j\leq\ell-1}+\beta y^B_{j\leq\ell-1},\gamma y^A_{j\geq\ell}+\delta y^B_{j\geq\ell}\right\}, \tag{53}$$

where $0\leq\alpha,\ \beta,\ \gamma,\ \delta\leq 1$, and $\alpha+\beta=1,\ \gamma+\delta=1$. Note that different combinations of coefficients are allowed on different subdomains. This situation corresponds to some of the MPRK-2 steps. The next lemma quantifies the TV increase of (53) from TV $(\{y_0\})$.

Lemma 3.7 (TV for convex combinations of solutions) *Convex combinations of TVD forward Euler steps of smooth solutions defined by* (53) *with the TVD and smoothness conditions* (52) *satisfied have at most a bounded TV increase of* $\mathcal{O}(\Delta t)$:

$$\mathrm{TV}\left(\left\{\alpha y^A_{j\le \ell-1} + \beta y^B_{j\le \ell-1}, \gamma y^A_{j\ge \ell} + \delta y^B_{j\ge \ell}\right\}\right) \le \mathrm{TV}\left(\left\{y^0\right\}\right) + \mathcal{O}(\Delta t). \tag{54}$$

Proof The TV semi-norm satisfies the triangle inequality, and in particular

$$\mathrm{TV}\left(\left\{\nu y^A + \omega y^B\right\}\right) \le \nu \mathrm{TV}\left(\left\{y^A\right\}\right) + \omega \mathrm{TV}\left(\left\{y^B\right\}\right),$$

where y^A and y^B are solutions defined on the same domain, and $\nu, \omega \in \mathbb{R}^+$.

We can expand (53) as follows

$$\begin{aligned}
\mathcal{TV} &= \mathrm{TV}\left(\left\{\alpha y^A_{j<\ell} + \beta y^B_{j<\ell}, \gamma y^A_{j\ge \ell} + \delta y^B_{j\ge \ell}\right\}\right) \\
&= \sum_{j\le \ell-1} \left|\alpha y^A_{j+1} + \beta y^B_{j+1} - \left(\alpha y^A_j + \beta y^B_j\right)\right| \\
&\quad + \left|\gamma y^A_\ell + \delta y^B_\ell - \left(\alpha y^A_{\ell-1} + \beta y^B_{\ell-1}\right)\right| + \sum_{j\ge \ell} \left|\gamma y^A_{j+1} + \delta y^B_{j+1} - \left(\gamma y^A_j + \delta y^B_j\right)\right|, \qquad (55) \\
&\le \alpha \sum_{j\le \ell-1} \left|y^A_{j+1} - y^A_j\right| + \left|\gamma y^A_\ell - \alpha y^A_{\ell-1}\right| + \gamma \sum_{j\ge \ell} \left|y^A_{j+1} - y^A_j\right| \\
&\quad + \beta \sum_{j\le \ell-1} \left|y^B_{j+1} - y^B_j\right| + \left|\delta y^B_\ell - \beta y^B_{\ell-1}\right| + \delta \sum_{j\ge \ell} \left|\delta y^B_{j+1} - y^B_j\right|. \qquad (56)
\end{aligned}$$

In (56) we regrouped the terms and factored out the linear combination coefficients. We also have that the coefficients are positive and $\beta = 1-\alpha$ and $\delta = 1-\gamma$:

$$\begin{aligned}
\left|\gamma y^A_\ell - \alpha y^A_{\ell-1}\right| &= \left|\gamma y^A_\ell - \gamma y^A_{\ell-1} + \gamma y^A_{\ell-1} - \alpha y^A_{\ell-1}\right| \\
&\le \gamma \left|y^A_\ell - y^A_{\ell-1}\right| + |\gamma - \alpha| \left|y^A_{\ell-1}\right|, \qquad (57) \\
\left|\delta y^B_\ell - \beta y^B_{\ell-1}\right| &= \left|\delta y^B_\ell - \delta y^B_{\ell-1} + \delta y^B_{\ell-1} - \beta y^B_{\ell-1}\right| \\
&\le \delta \left|y^B_\ell - y^B_{\ell-1}\right| + |\delta - \beta| \left|y^B_{\ell-1}\right|. \qquad (58)
\end{aligned}$$

We introduce (57) and (58) in (56), and substitute $\delta - \beta$ with $\alpha - \gamma$:

$$\begin{aligned}
\mathcal{TV} &\le \alpha \sum_{j\le \ell-1} \left|y^A_{j+1} - y^A_j\right| + \beta \sum_{j\le \ell-1} \left|y^B_{j+1} - y^B_j\right| \\
&\quad + \gamma \sum_{j\ge \ell-1} \left|y^A_{j+1} - y^A_j\right| + \delta \sum_{j\ge \ell-1} \left|\delta y^B_{j+1} - y^B_j\right| + |\gamma - \alpha| \left|y^A_{\ell-1} - y^B_{\ell-1}\right|. \qquad (59)
\end{aligned}$$

Since each solution is obtained through a TVD method (52), and the linear combination is convex, we have that

$$\mathcal{TV} \le \mathrm{TV}\left(\{y_0\}\right) + |\gamma - \alpha| \left|y^A_{\ell-1} - y^B_{\ell-1}\right|, \tag{60}$$

and thus, by using the assumptions in (52), we have that the maximum total variation increase is of order Δt. □

Next we use Lemmas 3.6 and 3.7 to show that MPRK-2 is TVB if each Euler step is TVD.

Proposition 3.8 (TVB) *If each fast and each slow multirate Euler steps are TVD under the local CFL condition and if the solution is smooth then MPRK-2 is TVB.*

Proof We consider MPRK-2 (see Table 3) applied to a bounded domain that has one fast region, a fast buffer, and one slow region ($\Omega = \Omega_F \cup \Omega_{\text{FB}} \cup \Omega_S$). Since each Euler step is considered to be TVD, the TVD conditions assumed in Lemma 3.6 are satisfied by the MPRK-2 construction (i.e., we allow for a buffer between the fast and the slow regions, in which the fast method is used, albeit the slow method is also TVD). In what follows we find an upper bound to the additional total variation that may be introduced by MPRK-2 in one (full) timestep. We consider each step in Table 3 and quantify the TV increase.

By Lemma 3.6, the first two steps ($\{y_F^{\{(1),(1*)\}}, y_S^{\{(1),(1*)\}}\}$) do not increase the TV of the initial solution (y^n):

$$\text{TV}\left(\left\{y_F^{(1*)}, y_S^{(1*)}\right\}\right) \le \text{TV}\left(\left\{y_F^{(1)}, y_S^{(1)}\right\}\right) \le \text{TV}\left(\left\{y_F^n, y_S^n\right\}\right). \tag{61}$$

The third step, $\{y_F^{(2)}, y_S^{(2)=n}\}$, represents a convex combination of Euler steps, and by using Lemma 3.7 together with the previous relations (61), we have that

$$\text{TV}\left(\left\{y_F^{(2)} = \frac{1}{2}y_F^n + \frac{1}{2}y_F^{(1*)}, y_S^{(2)} = 1y_F^n + 0y_F^{(1*)}\right\}\right) \le \text{TV}\left(\left\{y_F^n, y_S^n\right\}\right) + \mathcal{O}(\Delta t). \tag{62}$$

The first two steps of the second macro step $\{y_F^{\{(3),(3*)\}}, y_S^{\{(3),(3)*\}}\}$ do not increase the TV. Using Lemma 3.6 and the results from the previous Euler steps, we have the following TV bounds:

$$\text{TV}\left(\left\{y_F^{(3)}, y_S^{(3)}\right\}\right) \le \text{TV}\left(\left\{y_F^{(2)}, y_S^n\right\}\right) \le \text{TV}\left(\left\{y_F^n, y_S^n\right\}\right) + \mathcal{O}(\Delta t), \quad \text{and}$$

$$\text{TV}\left(\left\{y_F^{(3*)}, y_S^{(3*)}\right\}\right) \le \text{TV}\left(\left\{y_F^{(3)}, y_S^{(3)}\right\}\right) \le \text{TV}\left(\left\{y_F^n, y_S^n\right\}\right) + \mathcal{O}(\Delta t).$$

The last step, $\{y_F^{n+1}, y_S^{n+1}\}$, is a convex combination of previous Euler steps, and hence, using Lemma 3.7 we have an increase in the TV of at most $\mathcal{O}(\Delta t)$.

$$\begin{aligned}\text{TV}\left(\left\{y_F^{n+1} = \frac{1}{2}\left(y_F^{(2)} + y_F^{(3*)}\right), y_S^{n+1} = \frac{1}{2}\left(y_S^n + y_S^{(1*)}\right) + \frac{1}{2}\left(y_S^n + y_S^{(3*)}\right)\right\}\right)\\ \le \text{TV}\left(\left\{y_F^n, y_S^n\right\}\right) + \mathcal{O}(\Delta t).\end{aligned} \tag{63}$$

The last relation proves the TVB result for MPRK-2

$$\text{TV}\left(\left\{y_F^{n+1}, y_S^{n+1}\right\}\right) \le \text{TV}\left(\left\{y_F^n, y_S^n\right\}\right) + \mathcal{O}(\Delta t).$$

This proof can be extended to multiple subdomains and interfaces, and to arbitrary step size ratios. □

Forms (33, 34) represent a convex combination of Euler steps using the MPRK-2 construction "algorithm" presented in Sect. 3.4. The method (33, 34) with partitioning (20) is second order accurate in time and SSP (i.e., preserves the stability properties of the spatial method).

Table 4 The Butcher tableau for the two, slow and fast methods for Kirby's [30] first order multirate forward Euler step method

(a) Base method

0	0
	1

(b) Fast method

0	0				
η_1	σ_1	0			
η_2	σ_1	σ_2	0		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	
$\eta_{\mathfrak{m}-1}$	σ_1	σ_2	$\cdots$	$\sigma_{\mathfrak{m}-1}$	0
	σ_1	σ_2	$\cdots$	$\sigma_{\mathfrak{m}-1}$	$\sigma_{\mathfrak{m}}$

(c) Slow method

0	0				
0	0	0			
0	0	0	0		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	
0	0	0	$\cdots$	0	0
	σ_1	σ_2	$\cdots$	$\sigma_{\mathfrak{m}-1}$	$\sigma_{\mathfrak{m}}$

3.6 Other Multirate Explicit Methods

In this section we present other multirate forward Euler methods. First, we show the first order multirate Euler steps method proposed by Kirby [30] written in our framework and second, we construct other second order PRK configurations.

3.6.1 First Order Multirate Explicit Methods

Kirby [30] proposed a first order multirate method based on Euler steps. Using our notation this method can be written in the following way

$$\begin{aligned} y_F^{\eta_k} &= \mathcal{E}_F\left(\sigma_k \Delta t, y_F^{\eta_{k-1}}, y_S^n\right), \quad k = 1, 2, \ldots, \mathfrak{m}-1, \\ y_F^{n+1} &= \mathcal{E}_F\left(\sigma_{\mathfrak{m}} \Delta t, y_F^{\eta_{\mathfrak{m}-1}}, y_S^n\right), \\ y_S^{n+1} &= \mathcal{E}_S\left(\Delta t, y_F^n, y_S^n\right), \end{aligned}$$

where

$$\sum_{k=1}^{\mathfrak{m}} \sigma_k = 1, \quad 0 < \sigma_k \leq 1, \ 1 \leq k \leq \mathfrak{m}; \qquad \eta_k = \sum_{k=1}^{\mathfrak{m}-1} \sigma_k, \quad \eta_0 = 0.$$

Kirby [30] proves that the above scheme is TVD (i.e., $\mathrm{TV}(\{y_F^{n+1}, y_S^{n+1}\}) \leq \mathrm{TV}(\{y_F^n, y_S^n\})$), satisfies the maximum principle and the entropy condition; however, it is only first order accurate. A Butcher tableau representation of this method is shown in Table 4.

3.6.2 Other Second Order PRK Configurations

Other second order PRK configurations are possible. Using the fast method from our MPRK-2 example for $\mathfrak{m} = 2$, we define another family of second order PRK methods. The Butcher tableau is shown in Table 5. These methods can be written in convex combinations of Euler steps for $0 \leq \mu \leq 1$. The slow method is further discussed in Appendix 2 where we show that the Euler steps that form the method are in convex combinations. The properties proved for MPRK-2 also extend to this family.

Table 5 The Butcher tableau for the two, slow and fast methods for another family of second order PRK for $\mathfrak{m} = 2$. Here we consider $0 \leq \mu \leq 1$

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

(a) Fast method

0	0			
0	0	0		
1	μ	$1-\mu$	0	
1	$1-\mu$	μ	0	0
	1/4	1/4	1/4	1/4

(b) Slow method

3.7 Order Two MPRK Methods for Multiple Partitions

The MPRK-2 method described in a previous section (3.5) along with its properties can be extended to multiple levels of refinement. The procedure is illustrated in Table 6 for three levels of partitioning (S—slow, M—medium, and F—fast). The construction is as follows. Start with the base method for level zero, $\mathcal{RK}^B$ (in Table 1a). Then construct the slow method with A's on the diagonal. The top left quadrant in Table 6c becomes the base method for the medium partition, and so on.

For multiple levels, each method depends on its corresponding partition and only on the neighboring partitions. The neighboring partitions are characterized by adjacent time scales (there is no direct transition between the fast and the slow subdomains). In this case, the dependency of $f_{\{F,M,S\}}$ on $y_{\{F,M,S\}}$ is the following

$$\begin{aligned} y'_F &= f_F(y_F, y_M), \\ y'_M &= f_M(y_F, y_M, y_S), \\ y'_S &= f_S(y_M, y_S). \end{aligned}$$

We note that there is no direct dependency between flux functions f_F and f_S. The transition between the fast and the slow methods is smoothly resolved in this context.

The order conditions for each method and for the coupling are satisfied pairwise: (F, M) and (M, S). This is sufficient to guarantee the global accuracy of the method. Moreover, on level zero, $\mathcal{RK}_0^S$ reduces to the base method on Ω_S^0 (away from the interface), and in turn, $\mathcal{RK}_1^S$ reduces to the top left quadrant of $\mathcal{RK}_0^S$ on $\Omega_S^{(1)}$, which becomes the base method for the medium partition. Thus, we have a systematic way to extend methods to increasingly faster partitions.

3.8 Implementation Aspects

We describe the implementation of MPRK-2 in a general framework. For simplicity, we consider a solution that has two characteristic time scales that require two integration timestep lengths: large (for slow) and small (for fast). The domain is partitioned as follows: the fast solution together with the fast buffer form the fast domain (recall that variables in the fast buffer are characterized by a slow timescale, but are resolved with a small timestep). An additional slow buffer at the fast-slow interface is considered as illustrated in Fig. 2 to computationally decouple the fast and slow solutions. The rest of the slow solution forms the slow domain. The size of the slow buffer is the "largest" half of the maximum stencil size (denoted with Δ) times the ratio between the fast-slow subdomains $\mathfrak{m}$. For instance, in our experiments the spatial stencil has five points, and in this case we choose the slow buffer to

Table 6 Multirate partitioned Runge–Kutta method (MPRK-2) with 3 levels of refinement and an example for the nested domain decomposition

$\frac{1}{4}c$	$\frac{1}{4}A$			
$\frac{\mathbb{1}+c}{4}$	$\frac{1}{4}b$	$\frac{1}{4}A$		
$\frac{2\cdot\mathbb{1}+c}{4}$	$\frac{1}{4}b$	$\frac{1}{4}b$	$\frac{1}{4}A$	
$\frac{3\cdot\mathbb{1}+c}{4}$	$\frac{1}{4}b$	$\frac{1}{4}b$	$\frac{1}{4}b$	$\frac{1}{4}A$
	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$

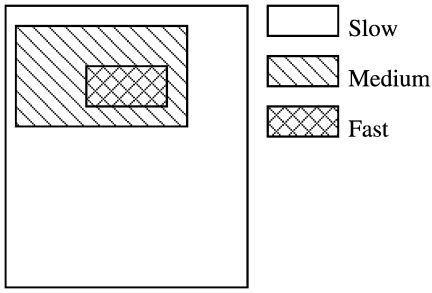
(a) Fast method ($\mathfrak{m}=4$)

$\frac{c}{2}$	$\frac{1}{2}A$			
$\frac{c}{2}$		$\frac{1}{2}A$		
$\frac{\mathbb{1}+c}{2}$	$\frac{1}{4}b$	$\frac{1}{4}b$	$\frac{1}{2}A$	
$\frac{\mathbb{1}+c}{2}$	$\frac{1}{4}b$	$\frac{1}{4}b$		$\frac{1}{2}A$
	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$

(b) Medium method ($\mathfrak{m}=2$)

c	A			
c		A		
c			A	
c				A
	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$	$\frac{1}{4}b^T$

(c) Slow method ($\mathfrak{m}=1$)



(d) Nested partitioning example

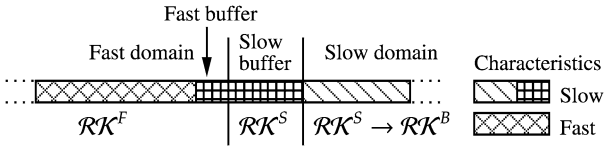


Fig. 2 The efficient implementation of MPRK treats separately the variables in the slow buffer

be two grid points ($\Delta = 2$). It is easy to see that this condition guarantees sufficient solution decoupling between the fast and the slow such that the base method can be applied for the slow partition. The size of the fast buffer is Δ according to Proposition 3.8. To illustrate the implementation we consider the MPRK-2 (RKa) method described in Table 3. The fast method ($\mathcal{RK}^F$) is applied in the fast domain (with step size $\Delta t/\mathfrak{m}$), the slow (buffer) method ($\mathcal{RK}^S$) is applied in the slow buffer, and the base method ($\mathcal{RK}^B$) in the slow domain. Note that inside the slow buffer $\mathfrak{m}$ large steps are taken with a stepsize Δt; but since the slow buffer is narrow the associated computational cost is only $\mathcal{O}(\mathfrak{m}\Delta)$. Outside the slow buffer a single large step of length Δt is taken with the base method to advance the slow solution in the large slow domain. A representation of the procedure described above is shown in Fig. 2. This analysis can be extended to any MPRK-2 setting.

Optimal (in terms of CFL) single rate SSP Runge–Kutta methods expressed directly in forward Euler steps can be found in [18] and are very good candidates for the base method in the multirate approach. The multirate sequence of Euler steps can be inferred directly from

the base method as the multirate ($\mathfrak{m} \times$ base method stages) stages have a repetitive pattern. An example is shown in Appendix 1 for $\mathfrak{m} = 3$.

In practice the timestep refinement criterion is imposed by stability restrictions (CFL) due to the fact that the spatial discretization errors typically dominate the temporal discretization errors (i.e., the time step is restricted by stability and not by accuracy requirements). It is nevertheless possible to choose an adaptive timestep for the time integration based on temporal error control [12, 44].

3.9 Computational Efficiency

There is no additional computational overhead away from the interface regions which are typically very small (narrow) compared to the fast and slow partitions. Even if the slow method has formally as many stages as the fast method, these stages are identical away from the interface, and thus no additional calculations are necessary. This means that the slow method really uses large steps.

We now estimate the efficiency of the multirate method under the assumption that the Euler steps (and in particular the flux function evaluations) carry the bulk of the computational cost. We define the multirate integration speedup as the ratio of the workload for single rate scheme with fast steps used throughout the domain, to the workload of the multirate scheme. Consider the multirate scheme applied on a fast domain with L_F grid points, a slow domain with L_S grid points, and with b interfaces among them. The speedup of the MPRK-2 is

$$S = \frac{\mathfrak{m}(L_F + L_S)}{\mathfrak{m}(L_F + b\mathfrak{m}\Delta) + L_S - b\mathfrak{m}\Delta}. \tag{64}$$

The speedup depreciates as $\mathfrak{m}$ grows; however, this is not of concern in practical applications since large $\mathfrak{m}$ can be avoided through the nested partitioning. In practice we have that $b\mathfrak{m}\Delta \ll \min(L_S, L_F)$ and therefore

$$S \approx \frac{\mathfrak{m}(L_F + L_S)}{\mathfrak{m}L_F + L_S}.$$

The speedup is close to the ideal value of $\mathfrak{m}$ if $L_F \ll L_S$. For k nested grids the ideal speedup is $\mathfrak{m}^k$. We expect the speedup to be considerable for multi-dimensional problems. A numerical experiment that validates this theoretical speedup considerations for Burgers' equation is shown in Sect. 4.2.3.

4 Numerical Results

We illustrate the theoretical findings using two standard test problems: the advection equation and the inviscid Burgers' equation. Since TVD methods in multiple dimensions are at most first order accurate [16], we look at one-dimensional problems. Accurate multiple dimension problems can be solved using dimension splitting. The solutions are computed using the method of lines approach. The linear advection spatial discretization is a second order limited finite volume scheme on nonuniform grids that is both conservative and positive (described in Sect. 4.1). Burgers' equation, presented in Sect. 4.2, is implemented on a fixed grid, using the third order scheme of Osher and Chakravarthy [39].

4.1 The Advection Equation

The one-dimensional advection equation (4.1) models the transport of a tracer y with the constant velocity u along the x

$$\frac{\partial y(t,x)}{\partial t} + u \cdot \frac{\partial y(t,x)}{\partial x} = 0.$$

4.1.1 Positive Spatial Discretization

In what follows, we describe the positive (7) flux limited spatial discretization scheme [27, 53]. We start by introducing the flux limited formulation of Hundsdorfer et al. [27] on uniform grids, and extend the scheme to nonuniform grids.

The numerical flux is defined as

$$F_{i+\frac{1}{2}} = f_i + \frac{1}{2}\phi_{i+\frac{1}{2}}(f_i - f_{i-1}), \tag{65}$$

where ϕ is a nonlinear limiter function. The scheme (2) becomes

$$y_i' = -\frac{\left(1 + \frac{1}{2}\phi_{i+\frac{1}{2}}\right)(f_i - f_{i-1}) - \frac{1}{2}\phi_{i-\frac{1}{2}}(f_{i-1} - f_{i-2})}{\Delta x}. \tag{66}$$

Define the flux slope ratio as

$$r_{i-\frac{1}{2}} = \frac{f_i - f_{i-1}}{f_{i-1} - f_{i-2}}. \tag{67}$$

If $r_{i-\frac{1}{2}} \neq 0$, from (66) and (67) we have

$$y_i' = -\frac{1}{\Delta x}\left[\left(1 + \frac{1}{2}\phi_{i+\frac{1}{2}}\right) - \frac{\frac{1}{2}\phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}}\right](f_i - f_{i-1}). \tag{68}$$

The positivity requirement (7) applied to the scheme (68) yields the following condition

$$\frac{\phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}} - \phi_{i+\frac{1}{2}} \leq 2. \tag{69}$$

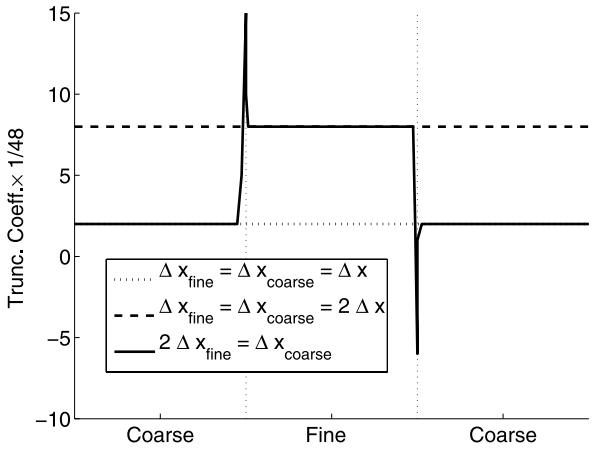
This condition is used to impose bounds on the limiter ϕ for the numerical flux defined by (65) in order to preserve positivity for the scheme (2).

Next we extend the scheme to a nonuniform grid by redefining the numerical flux (and the limiter). Consider a quadratic (spatial) flux interpolant for the numerical flux function F at $i + \frac{1}{2}$, using the flux function f, evaluated at gridpoints $i-1$, i, $i+1$, and a nonuniform spatial grid spacing, $\Delta x_{[\bullet]}$, in the following form:

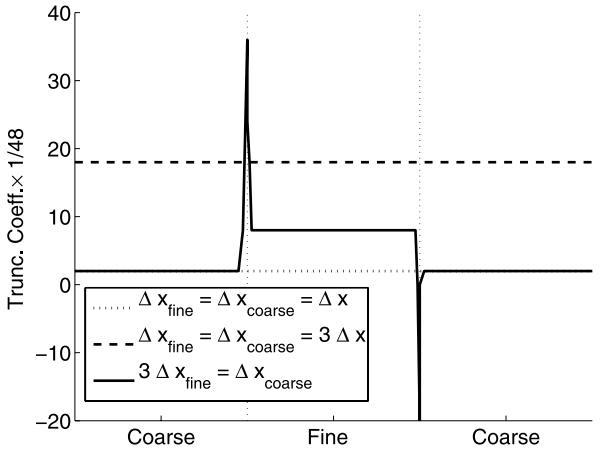
$$F_{i+\frac{1}{2}} = f_i + \alpha_i f_{i-1} + \beta_i f_i + \gamma_i f_{i+1},$$

where

$$\alpha_i = -\frac{\Delta x_i \Delta x_{i+1}}{(\Delta x_{i-1} + \Delta x_i)(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1})},$$



(a) $m = 2$



(b) $m = 3$

Fig. 3 Representation of the discretization leading order error term for two instances of $\mathfrak{m}$ as the wave passes through the interfaces

$$\beta_i = -\frac{\Delta x_i(\Delta x_{i-1} + \Delta x_i - \Delta x_{i+1})}{(\Delta x_{i-1} + \Delta x_i)(\Delta x_i + \Delta x_{i+1})},$$
$$\gamma_i = \frac{\Delta x_i(\Delta x_{i-1} + 2\Delta x_i)}{(\Delta x_i + \Delta x_{i+1})(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1})}.$$

The flux can be written in terms of $(f_i - f_{i-1})$ and $r_{i+\frac{1}{2}}$ as

$$F_{i+\frac{1}{2}} = f_i + (-\alpha_i + \gamma_i r_{i+\frac{1}{2}})(f_i - f_{i-1}). \tag{70}$$

Define $\mathcal{K}$ as

$$\mathcal{K}(r) = 2(-\alpha_i + \gamma_i r). \tag{71}$$

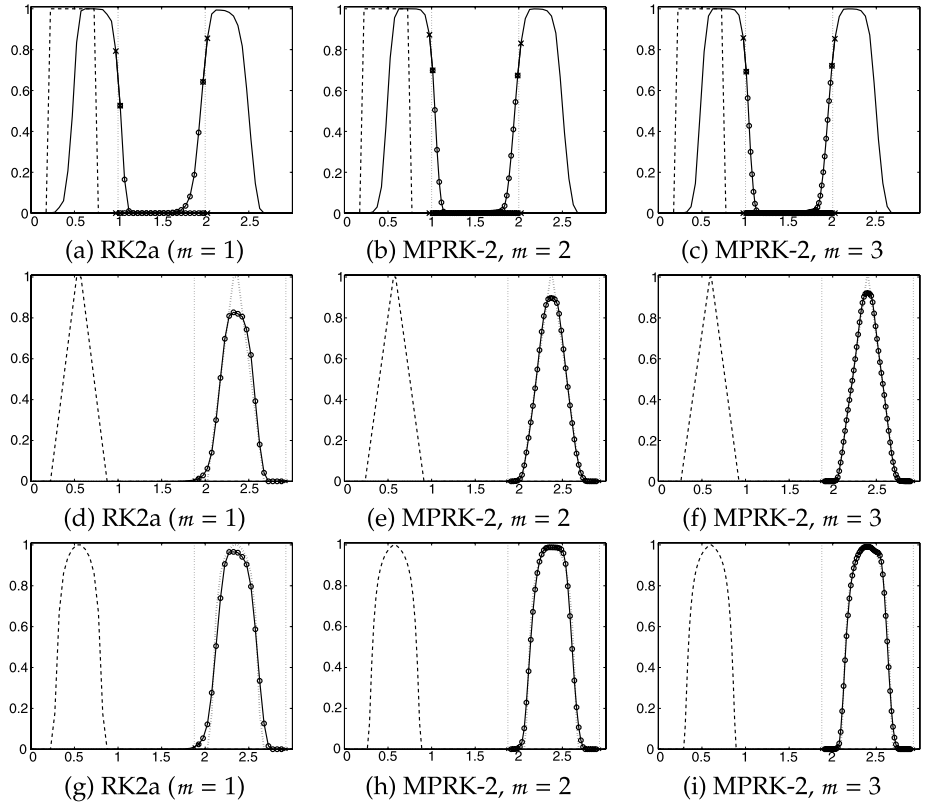


Fig. 4 Fixed grid advection solution with three function profiles that pass through a fixed fine $(\Delta x/\mathrm{m}, \Delta t/\mathrm{m})$ region (between 1 and 2). The initial profile (*dashed line*) is advected from the left to right part of the domain

Then, the numerical flux can be expressed as

$$F_{i+\frac{1}{2}} = f_i + \frac{1}{2}\mathcal{K}(r_{i+\frac{1}{2}})(f_i - f_{i-1}). \tag{72}$$

Define the following flux limiter [27, 51]

$$\phi(r) = \max(0, \min(2r, \min(2, \mathcal{K}(r)))). \tag{73}$$

Hundsdorfer et al. [27, Sect. 3.2] show that forward Euler timestepping used to integrate the semi-discrete form (2) using the numerical flux defined by (67) and the limiter (69) leads to a positive, limited second order (in space) scheme under a CFL restriction. We extend this scheme to nonuniform grids by using the numerical flux defined by (72) with the limiter (73). This extension preserves the positivity and accuracy properties (the proof follows immediately from [27]). In addition, if the timestepping scheme is positivity preserving, then the entire method (each multirate step, in our case) is positivity preserving as discussed in Sect. 3.5.

Figure 3 shows that the leading order truncation error of the spatial discretization using the unlimited numerical flux (70), i.e., the coefficient that multiplies $\partial^3 f/\partial x^3$, increases with increased mesh ratio.

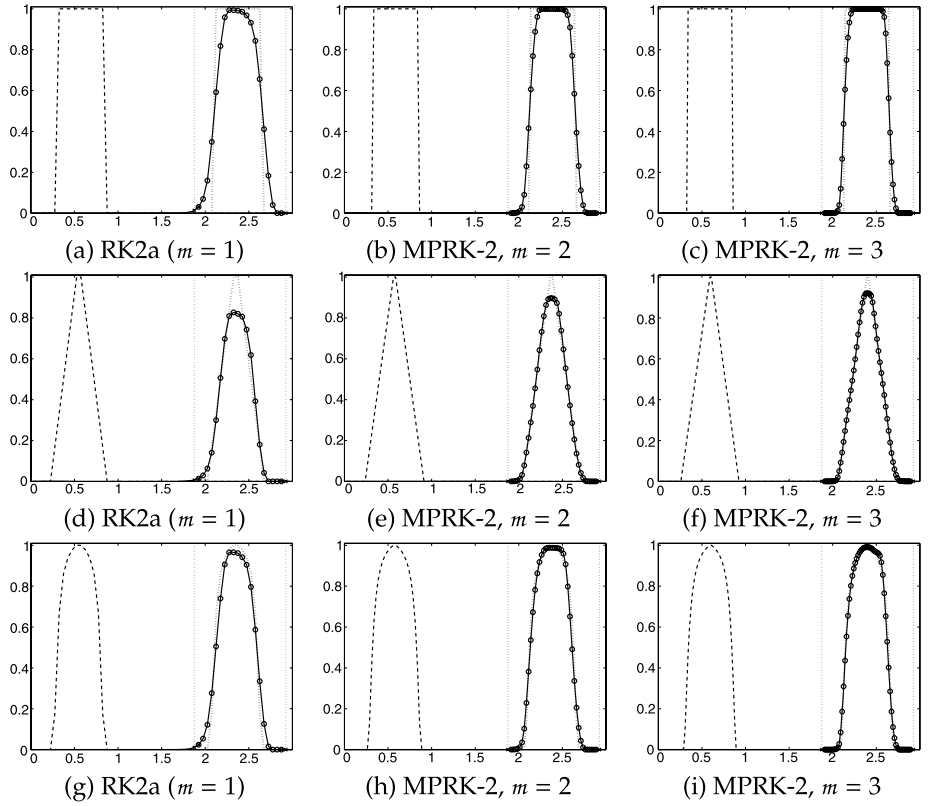


Fig. 5 Moving grid advection solution with three initial profiles (marked with *dashed lines*). The solution is advected to the right part of the domain. The fine grid ($\Delta x/\mathfrak{m}$, $\Delta t/\mathfrak{m}$) delimited by two *vertical dotted thin lines* (at the final time) follows the profile. The *dotted line* represents the exact solution at the final time

4.1.2 Numerical Experiments

In this section we apply the MPRK-2 time integration for the linear advection equation. The spatial discretization is positive and the time integration scheme is SSP, which results in an overall positive scheme.

Our test cases include three different initial conditions (of different regularity): a step function, a triangular shape, and an exponential shape.

The computational domain has three distinct regions. The middle region is discretized using a fine grid with spacing $\Delta x/\mathfrak{m}$, while the left and right regions are covered by a coarse mesh with spacing Δx. For simplicity we consider periodic boundary conditions. The timestepping interval is proportional with the grid size in order to satisfy the CFL restriction, i.e., we take Δt wherever we have Δx grid spacing and $\Delta t/\mathfrak{m}$ wherever we have $\Delta x/\mathfrak{m}$.

Figure 4 shows the advection numerical solution with the three function profiles that pass through a fixed fine ($\Delta x/\mathfrak{m}$) region (located between $x = 1$ and $x = 2$). The dashed line represents the exact solution and solid line corresponds to the solution evolved with unit wave speed ($u = 1$) in time (at two different times). The solution is not qualitatively affected by the interface. Moreover, with the higher spatial resolution, the solution improves qualitatively (as $\mathfrak{m}$ is increased), and the wave is not distorted by passing through the fast-

Table 7 The moving grid advection experiment values for: ℓ_1 error norm, the ratio between the initial and final mass, and the solution minimum and maximum over all timesteps (all). The initial solution minimum is zero and maximum is one

Type	$\mathrm{m}=1$	$\mathrm{m}=2$	$\mathrm{m}=3$
	mass final	mass final	mass final
	min all	min all	min all
	max all	max all	max all
Step	$\ell_1 = 0.1085$	$\ell_1 = 0.1069$	$\ell_1 = 0.1021$
	9.9999999e-01	1.0e+00	9.99999999e-01
	1.4966e-17	4.4738e-19	9.3703e-20
	0.9938	0.9999	1
Triangular	$\ell_1 = 0.0401$	$\ell_1 = 0.0224$	$\ell_1 = 0.0154$
	1	1	9.99999999e-01
	1.2242e-17	9.2255e-19	3.3607e-19
	0.8272	0.8969	0.9209
Exponential	$\ell_1 = 0.0466$	$\ell_1 = 0.0344$	$\ell_1 = 0.0270$
	9.99999999e-01	9.99999999e-01	9.99999999e-01
	1.0845e-17	5.7082e-19	1.489e-19
	0.9659	0.9880	0.9904

Table 8 Effective order for RK2a (i.e., MPRK-2 with $\mathrm{m}=1$) in time for the advection equation with respect to (left columns) the exact solution and (right columns) a high order time discretization method

	Space and time refinement				Time refinement only (ODE coarse)			
Level	ℓ_1	ℓ_2	EO_1	EO_2	ℓ_1^{ODE}	ℓ_2^{ODE}	$\mathrm{EO}_1^{\mathrm{ODE}}$	$\mathrm{EO}_2^{\mathrm{ODE}}$
1	8.10e-02	5.62e-02			2.04e-03	1.33e-03		
2	2.00e-02	1.64e-02	2.018	1.775	4.88e-04	3.21e-04	2.063	2.056
4	4.00e-03	4.31e-03	2.322	1.929	1.22e-04	8.21e-05	1.992	1.968
8	8.21e-04	1.13e-03	2.284	1.933	3.09e-05	2.06e-05	1.991	1.991
16	1.66e-04	2.99e-04	2.303	1.915	7.68e-06	5.18e-06	2.008	1.995
32	3.00e-05	7.25e-05	2.471	2.048	1.92e-06	1.30e-06	2.001	1.994
64	5.77e-06	1.81e-05	2.379	2.002	4.78e-07	3.20e-07	2.005	2.023

slow interface. To further quantify the benefits of having a finer region in this setting, we investigate a moving fine mesh that is centered around the "interesting region," where the large gradients occur in the solution. Figure 5 shows the advection solution for the three corresponding initial profiles (marked with dashed lines). The fine $(\Delta x/\mathrm{m})$ mesh travels with the solution from the left to the right part of the domain with constant speed. The figures show the final state of the solution with the exact solution superimposed (marked with dotted lines), and vertical dotted lines that delimit the fine domain. Table 7 shows the ℓ_1 error norm at the final time, minimum, maximum, and the total mass ratio for the profiles shown in Fig. 5 at the initial and final time. Clearly, the solution is improved both qualitatively and quantitatively with higher spatial resolution. Moreover, every timestep is

conservative, and the solutions are positive, obey the maximum principle, and are wiggle free.

All the results for the advection equation presented in this section show that this specific finite volume approach and MPRK-2 yield a conservative multirate solution on a nonuniform grid that is positive and non-oscillatory as discussed in Sect. 3.5.

4.1.3 Numerical Error Analysis

In this section we analyze the effective (numerical) order of accuracy of the multirate time integration method. For this investigation we use the experiment setting (i.e., space discretization, grid, etc.) described above and consider the following elements.

Given the reference (or exact) solution y^{ref}, ℓ_q error norms of the numerical solution y, on a variable (or fixed) grid are computed as

$$\ell_q(y) = \left(\sum_i \Delta x_i \left| y(x_i) - y^{\text{ref}}(x_i) \right|^q \right)^{\frac{1}{q}}.$$

The effective order of the discretization EO_q in q norm is estimated from two numerical solutions with different time resolutions: h and h/γ, $\gamma > 1$, as follows:

$$\text{EO}_q = \log\left(\frac{\ell_q(y^{[h/\gamma]})}{\ell_q(y^{[h]})} \right) \Big/ \log\left(\frac{1}{\gamma} \right),$$

In our numerical experiments, we use the ℓ_1 and ℓ_2 norms and consider the sine initial solution shown in Fig. 6a that is advected half of its period (i.e., 1.5 units in the spatial dimension).

For the numerical order validation we use two space-time grids. In the first case, we consider space-time refinement for the fine region and compute the error norms ℓ_1 and ℓ_2 with respect to the exact solution. They are used to determine the corresponding effective order of the (space/time) method EO_1 and EO_2, respectively. In the second case, we consider only time refinement while keeping the space grid fixed and compute the error norms ℓ_1^{ODE} and ℓ_2^{ODE}. The reference solution in the fixed grid is obtained with respect to a high order numerical approximation, RK45. RK45 is obtained using explicit Runge–Kutta (4, 5) with RelErr = AbsErr = 1E-08 [9]. The spatial discretization is the same in both MPRK-2 and RK45. The error norms are used to approximate directly the time accuracy order EO_1^{ODE} and EO_2^{ODE} for MPRK-2, masking the spatial errors.

First, we consider the single rate RK2a method (Table 8). The results subscribe to the well-known second order of accuracy. Next, we show the order estimation for MPRK-2 with $\mathfrak{m} = 2$ in Table 9 and with $\mathfrak{m} = 3$ in Table 10. The second order of accuracy is not affected by the time refinement in the MPRK-2 method and confirms the theoretical findings.

We remark that the errors (in norm) decrease as we increase $\mathfrak{m}$. Moreover, the errors with respect to the exact solution (left columns) that also include the spatial discretization errors are orders of magnitude larger than the "ODE" errors (right columns). This shows that the spatial discretization errors dominate, as expected.

4.2 Burgers' Equation

The simplified (inviscid) Burgers' equation is

$$\frac{\partial y(t,x)}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2} y(t,x)^2 \right) = 0. \tag{74}$$

Table 9 Effective order for MPRK-2 (RK2a) with $\mathfrak{m} = 2$ in time for the advection equation with respect to (left columns) the exact solution and space-time refinement and (right columns) a single rate high order time discretization method with time refinement for MPRK-2

	Space and time refinement				Time refinement only (ODE coarse)			
Level	ℓ_1	ℓ_2	EO_1	EO_2	ℓ_1^{ODE}	ℓ_2^{ODE}	EO_1^{ODE}	EO_2^{ODE}
1	6.49e-02	4.75e-02			1.62e-03	1.12e-03		
2	1.57e-02	1.41e-02	2.047	1.753	3.91e-04	2.70e-04	2.052	2.061
4	3.27e-03	3.61e-03	2.264	1.963	9.76e-05	6.84e-05	2.003	1.982
8	7.12e-04	9.37e-04	2.201	1.948	2.43e-05	1.71e-05	2.003	1.995
16	1.52e-04	2.47e-04	2.221	1.924	6.11e-06	4.31e-06	1.994	1.990
32	3.04e-05	6.14e-05	2.326	2.008	1.53e-06	1.08e-06	1.994	1.993
64	6.24e-06	1.55e-05	2.287	1.980	3.74e-07	2.64e-07	2.037	2.034

Table 10 Effective order for MPRK-2 (RK2a) with $\mathfrak{m} = 3$ in time for the advection equation with respect to (left columns) the exact solution and space-time refinement and (right columns) a single rate high order time discretization method with time refinement for MPRK-2

	Space and time refinement				Time refinement only (ODE coarse)			
Level	ℓ_1	ℓ_2	EO_1	EO_2	ℓ_1^{ODE}	ℓ_2^{ODE}	EO_1^{ODE}	EO_2^{ODE}
1	4.12e-02	3.40e-02			1.61e-03	1.13e-03		
2	1.00e-02	9.79e-03	2.041	1.795	3.87e-04	2.70e-04	2.059	2.068
4	2.23e-03	2.49e-03	2.165	1.973	9.59e-05	6.81e-05	2.013	1.987
8	5.05e-04	6.58e-04	2.144	1.922	2.39e-05	1.70e-05	2.006	1.996
16	1.08e-04	1.73e-04	2.220	1.927	6.00e-06	4.30e-06	1.993	1.990
32	2.24e-05	4.41e-05	2.269	1.973	1.50e-06	1.08e-06	1.995	1.993
64	4.75e-06	1.16e-05	2.241	1.919	3.67e-07	2.63e-07	2.037	2.035

Burgers' equation numerical experiments are based on the third order upwind-biased TVD flux limited scheme described below for the spatial discretization, and MPRK-2 for the time integration.

4.2.1 TVD Spatial Discretization

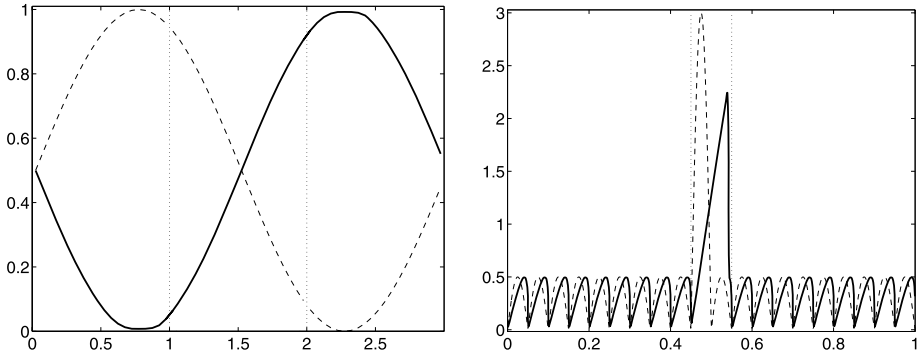
This section is based on the work of Osher and Chakravarthy [6, 38, 39]. A generic recipe for high order TVD finite volume schemes can be found in [7]. In what follows, we briefly present their method.

Consider the flux $F(y_{j+1}, y_j)$ to be a scalar numerical flux defined for an E-scheme [7]. The following

$$df^-_{j+\frac{1}{2}} = F(y_{j+1}, y_j) - f(y_j), \quad \text{and} \tag{75}$$

$$df^+_{j+\frac{1}{2}} = f(y_{j+1}) - F(y_{j+1}, y_j), \tag{76}$$

represent the positive and negative flux difference on the cell face.



(a) Solution for accuracy results (b) Solution for efficiency results

Fig. 6 Representation of the initial (*dashed*) and final (*solid*) solutions for: **a** the advection equation used in the numerical accuracy analysis, and **b** Burgers' equation used in the numerical efficiency experiments. The *dashed vertical lines* denote the fast-slow interface

With (75, 76), consider the following numerical flux

$$F_{j+\frac{1}{2}} = F(y_{j+1}, y_j) - \left[\frac{1-\kappa}{4}\widetilde{d}f^-_{j+\frac{3}{2}} + \frac{1+\kappa}{4}\overline{d}f^-_{j+\frac{1}{2}}\right] + \left[\frac{1+\kappa}{4}\widetilde{d}f^+_{j+\frac{1}{2}} + \frac{1-\kappa}{4}\overline{d}f^+_{j-\frac{1}{2}}\right], \tag{77}$$

$$F(y_{j+1}, y_j) = \frac{1}{2}(f(y_{j+1}) + f(y_j)) - \frac{1}{2}\left(df^+_{j+\frac{1}{2}} + df^-_{j+\frac{1}{2}}\right), \tag{78}$$

where $f^\pm$ are the negative and positive flux contributions, $\widetilde{d}f^\pm$ and $\overline{d}f^\pm$ show that they are in flux limited form and are defined below. The scheme defined by (77) is called a κ-scheme. If $\kappa = 1/3$, (77) becomes a limited third order scheme. If we consider $\overline{d}f^\pm = df^\pm$ and $\widetilde{d}f^\pm = df^\pm$, we have the unlimited scheme. The limited fluxes are defined as follows

$$\widetilde{d}f^-_{j+\frac{3}{2}} = \text{minmod}\left[df^-_{j+\frac{3}{2}}, bdf^-_{j+\frac{1}{2}}\right], \qquad \overline{d}f^-_{j+\frac{1}{2}} = \text{minmod}\left[df^-_{j+\frac{1}{2}}, bdf^-_{j+\frac{3}{2}}\right], \tag{79}$$

$$\widetilde{d}f^+_{j+\frac{1}{2}} = \text{minmod}\left[df^+_{j+\frac{1}{2}}, bdf^+_{j-\frac{1}{2}}\right], \qquad \overline{d}f^+_{j-\frac{1}{2}} = \text{minmod}\left[df^+_{j-\frac{1}{2}}, bdf^+_{j+\frac{1}{2}}\right], \tag{80}$$

where

$$\text{minmod}[x, y] = \text{sign}(x) \cdot \max[0, \min[|x|, y\,\text{sign}(x)]], \qquad 1 \le b \le \frac{3-\kappa}{1-\kappa}. \tag{81}$$

Using forward Euler steps in time to discretize form (2) with the numerical flux defined by (79) and $\kappa = 1/3$ leads to a linearly stable, TVD, third order spatially accurate method (when the limiter is not "active", otherwise the order is degraded) under a CFL restriction, as shown in [39]. Additional information can be found in [7, 33]. Our numerical experiments confirm that that these properties extend to MPRK-2 (with RK2a) as predicted by our theoretical results in Sect. 3.5.

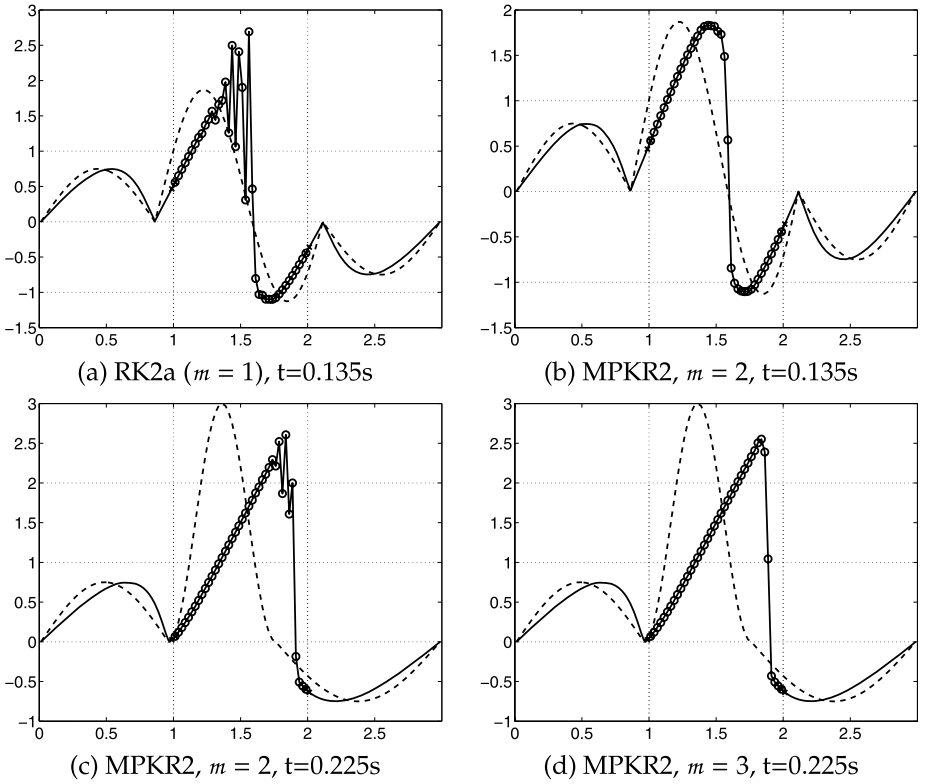
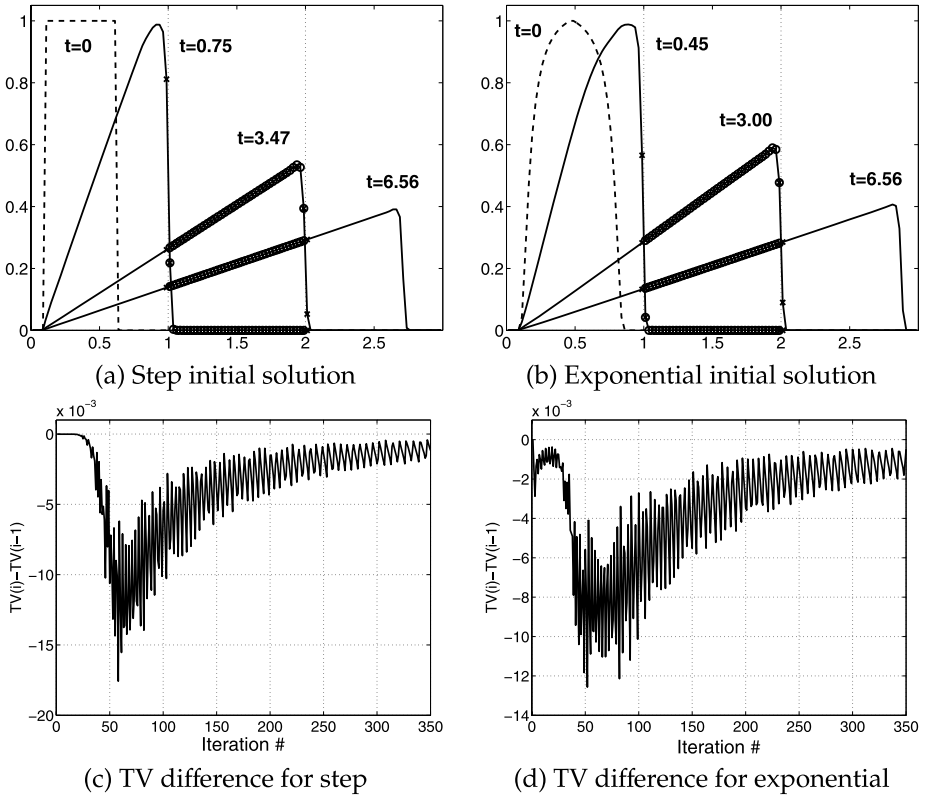


Fig. 7 Burgers' solution with the initial profile (*dashed*), $dx = 0.025$, $dt_{\text{coarse}} = 0.022$ (CFL = 0.9), and **a** $\mathfrak{m} = 1$, using RK2a; **b** $\mathfrak{m} = 2$, solved with MPRK-2. Second row shows the solution at $t = 0.225$ s solved with MPKR2 and same grid with **c** $\mathfrak{m} = 2$; **d** $\mathfrak{m} = 3$. CFL condition is violated in (**a**) and (**c**) and are unstable. Figures (**b**) and (**d**) satisfy the CFL condition and are stable

4.2.2 Numerical Experiments

The computational domain has three distinct regions. The middle region ($x \in [1, 2]$) is discretized using a fast method with the timestep of $\Delta t/\mathfrak{m}$, while the left ($x \in [0, 1]$) and right ($x \in [2, 3]$) form the slow regions and are solved with a time step of Δt. Again, for simplicity, we consider periodic boundary conditions. The time integration is done with MPRK-2.

Results for MPRK-2 that use smaller local timesteps for Burgers' equation are presented in Fig. 7. The local CFL condition is violated for Fig. 7a, c, and in these cases the solution becomes unstable. In Fig. 7b, d. the CFL condition is locally satisfied, and the solution is stable. More exactly: The solution in Fig. 7a uses the same timestep everywhere ($\mathfrak{m} = 1$); however, in the fast region ($x = [1, 2]$), the CFL condition is violated (for $y > 1.11$) and the method becomes unstable. In Fig. 7b we use a smaller timestep in the fast region ($\mathfrak{m} = 2$) and the local CFL condition is now satisfied, and the oscillations present in Fig. 7a are avoided. Similarly, the solution in Fig. 7c is oscillatory for $\mathfrak{m} = 2$ due to a violation of the CFL condition. Increasing the time ratio to $\mathfrak{m} = 3$ stabilizes the solution, and hence, the solution (shown in Fig. 7d) is non-oscillatory. This approach with different $\mathfrak{m}$ can be employed in order to dynamically stabilize the schemes by choosing step sizes that locally satisfy the CFL condition. Figure 8a, b shows the Burgers' solution obtained using two



(a) Step initial solution (b) Exponential initial solution

(c) TV difference for step (d) TV difference for exponential

Fig. 8 Burgers' solution with the initial profile (*dashed*), $dx = 0.025$, $dt_{\text{fine}} = 0.019$ (CFL $= 0.75$) and solved with MPRK-2 for **a** the step profile, and **b** exponential profile, for $\mathfrak{m} = 2$ at different time locations. For each profile we show the TV variation of the solution: in **c** for the **a** setting, and in **d** for the **b** setting

initial profiles that pass through the fine ($\Delta t/2$) region at different times. In both cases the solution is not qualitatively affected by the wave passing through the interfaces. The spatial discretization scheme is TVD and stable under a CFL-like condition. The time integration scheme, MPRK-2, with $\mathfrak{m} = 2, 3$ keeps a bounded total variation. Figure 8c, d shows the TV difference (between successive steps), i.e., $\text{TV}(y(t = t_i)) - \text{TV}(y(t = t_{i-1}))$, for the solutions presented in Fig. 8a, b. This difference is always negative, and thus the scheme is TVD on this particular example.

4.2.3 Computational Efficiency

In this section we validate the theoretical speedup results computed in Sect. 3.9. We consider Burgers' equation (74) with the solution shown in Fig. 6b using the Osher–Chakravarthy spatial discretization (77) described above.

We consider a fast region with ratios $\mathfrak{m} = 2$ or 3 that covers 10% of the entire grid in order to preserve the stability of the method for the given initial solution. We choose: $L_F = 100$, $L_S = 900$, $\Delta = 2$ with CFL (fine) = 0.75 and two interfaces ($b = 2$): fast-slow and slow-fast. In Table 11 we show the CPU time for 45 integration timesteps and compare the multirate solutions with $\mathfrak{m} = 2$ and 3 with the single rate solutions. The experimental results

Table 11 Effective and theoretical computational speedup for MPRK-2 with ($\mathtt{m} = 2$ and 3). The fast region covers 10% of the entire domain. A considerable speedup increase can be obtained for multidimensional applications

Time ratio	Single rate time [sec]	Multirate time [sec]	Experimental speedup	Theoretical speedup
$\mathtt{m} = 2$	25.28	13.71	1.84	1.80
$\mathtt{m} = 3$	36.73	15.07	2.43	2.45

confirm a speedup of about 1.8 for $\mathtt{m} = 2$ and 2.43 for $\mathtt{m} = 3$, as predicted by the theoretical calculation.

We note that in 2-D and 3-D applications the speedup is expected to be considerably larger, as discussed in Sect. 3.9. Similarly, more impressive speedups are expected for nested grids.

5 Conclusions and Future Work

Adaptive simulations of hyperbolic conservation laws refine the spatial grid to obtain the target accuracy. Due to the CFL restrictions, finer local grids lead to smaller global timesteps for the entire simulation. Therefore mesh refinement is accompanied by a considerable increase in the computational time. Moreover, even for fixed grid simulations, the wave speeds may vary considerably across the entire domain and the global timestep is restricted by the fastest wave speed. In both cases, the majority of the variables are solved with a timestep much smaller than necessary.

Multirate integration schemes use different timesteps for distinct components of the solution; in particular, they allow to use different timesteps in different parts of the domain when simulating hyperbolic systems. In this paper we present a multirate approach that allows to solve each subdomain with a timestep that matches the local characteristic time scale of the solution. The proposed multirate schemes have high order of accuracy and have nonlinear stability properties.

We have developed a systematic way to extend SSP Runge–Kutta schemes to multirate integration methods. Our approach is rooted in the theory of partitioned Runge–Kutta methods. The proposed time discretizations are (1) second order accurate, (2) conservative, and (3) nonlinearly stable under local CFL timestep restrictions. Nonlinear stability properties include positivity, maximum principle preserving, and TVB. Note that current multirate methods with these properties, available in the literature, are at most first order accurate. The proposed multirate family of schemes can be extended to accommodate an arbitrary number of partitions (time scales), with arbitrary step size ratios between partitions.

Two test problems are used to illustrate the theory. In both problems we used MPRK-2 with different timestep ratios. The first test is the linear advection equation with mesh refinement. Distinct timesteps are used in areas of different mesh sizes. Under local CFL conditions, the integration is linearly stable and conservative, and the solution remains positive and free of spurious oscillations. The second test is the inviscid Burgers' equation. The grid size is fixed but the wave speed varies significantly

Table 12 MPRK order 2 Butcher tableau for $\mathfrak{m} = 3$

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

(a) Base method

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

(b) Fast method

0	0					
1	1	0				
0	0	0	0			
1	0	0	1	0		
0	0	0	0	0	0	
1	0	0	0	0	1	0
	1/6	1/6	1/6	1/6	1/6	1/6

(c) Slow method

in different parts of the domain. Different timesteps that obey the local CFL conditions are used. The numerical solution is conserved and its total variation decreases with time.

A check for the entropy inequality in the multirate context will be addressed in future studies. We plan to extend the proposed framework to construct methods of third or higher order. Adaptive and automated partitioning methods need to be investigated in order to thoroughly take advantage of the these multirate methods. Further, we plan to develop multirate SSP time integrators based on linear multistep methods. We shall apply these multirate timestepping algorithms to the solution of large scale 3-D PDEs arising in air quality modeling.

Appendix 1: An Order Two MPRK with $\mathfrak{m} = 3$

Here we present the MPRK-2 using a factor of three ($\mathfrak{m} = 3$) between the fast and slow partitions using the same construction algorithm described for $\mathfrak{m} = 2$ (see Sect. 3.5). The method showed here takes advantage of the repetition pattern which is inherent for MPRK-2. Note that the base method is repeated in various combinations. The Butcher tableau for this method is shown in Table 12. The method can be written as follows ($k = 0, 1, 2$):

$$
\begin{aligned}
K_F^{1+2k} &= f_F(y_F^{n+2k}, y_S^n), & K_S^{1+2k} &= f_S(y_F^{n+2k}, y_S^n),\\
y_F^{(1+2k)} &= y_F^{n+2k} + \frac{\Delta t}{3} K_F^{1+2k}, & y_S^{(1+2k)} &= y_S^n + \Delta t K_S^{1+2k},\\
K_F^{2+2k} &= f_F(y_F^{(1+2k)}, y_S^{(1+2k)}), & K_S^{2+2k} &= f_S(y_F^{(1+2k)}, y_S^{(1+2k)}),\\
y_F^{(2+2k)} &= y_F^{n+2k} + \frac{\Delta t}{6} K_F^{1+2k} + \frac{\Delta t}{6} K_F^{2+2k}, & y_S^{(2+2k)} &= y_S^n,\\
y_F^{n+1} &= y_F^n + \frac{\Delta t}{6} \sum_{j=1}^{6} K_F^j, & y_S^{n+1} &= y_S^n + \frac{\Delta t}{6} \sum_{j=1}^{6} K_S^j.
\end{aligned}
\tag{82}
$$

The above MPRK-2 can be written in Euler steps in the following way ($k = 0, 1, 2$):

$$y_F^{(1+2k)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{n(+2k)}, y_S^{n(+2k)}\right), \qquad y_F^{((1+2k)*)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{(1+2k)}, y_S^{(1+2k)}\right),$$
$$y_F^{(2+2k)} = \frac{1}{2} y_F^{n(+2k)} + \frac{1}{2} y_F^{((1+2k)*)},$$

$$\begin{aligned}
y_F^{n+1} &= \frac{1}{2}\left(y_F^n + y_F^n + \frac{\Delta t}{3}K_F^1 + \frac{\Delta t}{3}K_F^2 + \frac{\Delta t}{3}K_F^3 + \frac{\Delta t}{3}K_F^4 + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^n + y_F^{(1)} + \frac{\Delta t}{2}K_F^2 + \frac{\Delta t}{3}K_F^3 + \frac{\Delta t}{3}K_F^4 + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^n + y_F^{(1*)} + \frac{\Delta t}{3}K_F^3 + \frac{\Delta t}{3}K_F^4 + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^{(2)} + y_F^{(2)} + \frac{\Delta t}{3}K_F^3 + \frac{\Delta t}{3}K_F^4 + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^{(2)} + y_F^{(3)} + \frac{\Delta t}{3}K_F^4 + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^{(2)} + y_F^{(3*)} + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^{(4)} + y_F^{(4)} + \frac{\Delta t}{3}K_F^5 + \frac{\Delta t}{3}K_F^6\right),\\
&= \frac{1}{2}\left(y_F^{(4)} + y_F^{(5*)}\right),
\end{aligned} \tag{83}$$

and

$$\begin{aligned}
y_S^{n+1} &= \frac{1}{6}\left(6y_S^n + \Delta t K_S^1 + \Delta t K_S^2 + \Delta t K_S^3 + \Delta t K_S^4 + \Delta t K_S^5 + \Delta t K_S^6\right),\\
y_S^{(1+2k)} &= \mathcal{E}_S\left(\Delta t, y_F^{n(+2k)}, y_S^n\right)\\
&= \frac{1}{6}\left(3y_S^n + y_S^{(1)} + \Delta t K_S^2 + y_S^{(3)} + \Delta t K_S^4 + y_S^{(5)} + \Delta t K_S^6\right),\\
y_S^{(1+2k)*} &= \mathcal{E}_S\left(\Delta t, y_F^{(1+2k)}, y_S^{(1+2k)}\right) = \frac{1}{6}\left(3y_S^n + y_S^{(1*)} + y_S^{(3*)} + y_S^{(5*)}\right).
\end{aligned} \tag{84}$$

The Euler steps for the fast and the slow methods are summarized in Table 13.

Appendix 2: The Slow Method as Convex Combination of Euler Steps

In Sect. 3.6.2 we showed a multirate PRK method family and argued that it can be expressed as convex combinations of Euler steps. The sequence of forward Euler steps of the fast method has already been presented in Sect. 3.5. Here we show that the slow method

Table 13 MPRK-2 Euler steps for the fast and slow methods for $\mathfrak{m} = 3$

Fast method ($\mathcal{RK}^F$) y_F^n	Slow method ($\mathcal{RK}^S$) in slow buffer y_S^n	Slow method ($\mathcal{RK}^B$) in slow region y_S^n
$y_F^{(1)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^n, y_S^n\right)$	$y_S^{(1)} = \mathcal{E}_S\left(\Delta t, y_F^n, y_S^n\right)$	$y_S^{(1)} = \mathcal{E}_S\left(\Delta t, y_F^n, y_S^n\right)$
$y_F^{(1*)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{(1)}, y_S^{(1)}\right)$	$y_S^{(1*)} = \mathcal{E}_S\left(\Delta t, y_F^{(1)}, y_S^{(1)}\right)$	$y_S^{(1*)} = \mathcal{E}_S\left(\Delta t, y_F^{(1)}, y_S^{(1)}\right)$
$y_F^{(2)} = \frac{1}{2}\left(y_F^n + y_F^{(1*)}\right)$		$y_S^{n+1} = \frac{1}{2}\left(y_S^n + y_S^{(1*)}\right)$
$y_F^{(3)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{(2)}, y_S^n\right)$	$y_S^{(3)} = \mathcal{E}_S\left(\Delta t, y_F^{(2)}, y_S^n\right)$	
$y_F^{(3*)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{(3)}, y_S^{(3)}\right)$	$y_S^{(3*)} = \mathcal{E}_S\left(\Delta t, y_F^{(3)}, y_S^{(3)}\right)$	
$y_F^{(4)} = \frac{1}{2}t\left(y_F^{(2)} + y_F^{(3*)}\right)$		
$y_F^{(5)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{(4)}, y_S^n\right)$	$y_S^{(5)} = \mathcal{E}_S\left(\Delta t, y_F^{(4)}, y_S^n\right)$	
$y_F^{(5*)} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^{(5)}, y_S^{(5)}\right)$	$y_S^{(5*)} = \mathcal{E}_S\left(\Delta t, y_F^{(5)}, y_S^{(5)}\right)$	
$y_F^{n+1} = \frac{1}{2}\left(y_F^{(4)} + y_F^{(5*)}\right)$	$y_S^{n+1} = \frac{1}{6}\left(3y_S^n + y_S^{(1*)} + y_S^{(3*)} + y_S^{(5*)}\right)$	

Table 14 The Butcher tableau for the slow methods for another family of second order PRK for $\mathfrak{m} = 2$. Here we consider $0 \le \mu \le 1$.

0	0			
0	0	0		
1	μ	$1-\mu$	0	
1	$1-\mu$	μ	0	0
	1/4	1/4	1/4	1/4

(a) Slow method in Table 5

0	0			
0	0	0		
1	$a_S^{3,1}$	$a_S^{3,2}$	0	
1	$a_S^{4,1}$	$a_S^{4,2}$	0	0
	1/4	1/4	1/4	1/4

(b) Slow method (in generic terms)

described in Table 5 can also be represented as convex combinations of Euler steps. The RK stages for the slow method in Table 14b are as follows:

$$\begin{aligned} K_S^1 &= f_S(y_F^n, y_S^n), & y_S^{(1)} &= y_S^n, \\ K_S^2 &= f_S(y_F^{(1)}, y_S^n), & y_S^{(2)} &= y_S^n + a_S^{3,1}\Delta t K_S^1 + a_S^{3,2}\Delta t K_S^2, \\ K_S^3 &= f_S(y_F^{(2)}, y_S^{(2)}), & y_S^{(3)} &= y_S^n + a_S^{4,1}\Delta t K_S^1 + a_S^{4,2}\Delta t K_S^2, \\ K_S^4 &= f_S(y_F^{(3)}, y_S^{(3)}), & y_S^{n+1} &= y_S^n + \frac{\Delta t}{4}\left(K_S^1 + K_S^2 + K_S^3 + K_S^4\right). \end{aligned} \tag{85}$$

The above RK stages can be expressed in the following way:

$$\begin{aligned} y_S^{n+1} &= \frac{1}{4}\left(4y_S^n + \Delta t K_S^1 + \Delta t K_S^2 + \Delta t K_S^3 + \Delta t K_S^4\right) \\ &= \frac{1}{4}\left(\left(y_S^n + a_S^{3,1}\Delta t K_S^1 + y_S^n + a_S^{3,2}\Delta t K_S^2\right)\right. \\ &\quad \left. + \left(y_S^n + a_S^{4,1}\Delta t K_S^1 + y_S^n + a_S^{4,2}\Delta t K_S^2\right) + \Delta t K_S^3 + \Delta t K_S^4\right). \end{aligned} \tag{86}$$

From (86), by convexity we have that

$$a_S^{3,1} + a_S^{4,1} \leq 1 \quad \text{and} \quad a_S^{3,2} + a_S^{4,2} \leq 1. \tag{87}$$

By construction we have

$$0 \leq a_S^{3,1}, a_S^{3,2}, a_S^{4,1}, a_S^{4,2} \leq 1 \quad \text{and} \quad a_S^{3,1} + a_S^{3,2} = 1, a_S^{4,1} + a_S^{4,2} = 1. \tag{88}$$

Note that the second order (coupling) conditions:

$$a_S^{3,1} + a_S^{3,2} + a_S^{4,1} + a_S^{4,2} = 2$$

are satisfied automatically.

Using (87, 88) the scheme in Table 14b can be written as in Table 14a, and using $0 \leq \mu \leq 1$, it can be expressed as convex combinations of Euler steps.

References

1. Andrus, J.: Numerical solution for ordinary differential equations separated into subsystems. SIAM J. Numer. Anal. **16**, 605–611 (1979)
2. Andrus, J.: Stability of a multi-rate method for numerical integration of ODEs. Comput. Math. Appl. **25**, 3–14 (1993)
3. Bartel, A., Günther, M.: A multirate W-method for electrical networks in state-space formulation. J. Comput. Appl. Math. **147**, 411–425 (2002)
4. Berkvens, P., Botchev, M., Lioen, W., Verwer, J.: A zooming technique for wind transport of air pollution. Tech. report, Centrum voor Wiskundeen Informatica, 1999
5. Boris, J., Book, D.: Flux-corrected transport I. SHASTA, a fluid transport algorithm that works. J. Comput. Phys. **135**, 172–186 (1997)
6. Chakravarthy, S., Osher, S.: Numerical experiments with the Osher upwind scheme for the Euler equations. AIAA J. **21**, 241–1248 (1983)
7. Chakravarthy, S., Osher, S.: Computing with high-resolution upwind schemes for hyperbolic equations. Lect. Appl. Math. **22**, 57–86 (1985)
8. Dawson, C., Kirby, R.: High resolution schemes for conservation laws with locally varying time steps. SIAM J. Sci. Comput. **22**, 2256–2281 (2001)
9. Dormand, J., Prince, P.J.: A family of embedded Runge–Kutta formulae. J. Comput. Appl. Math. **6**, 19–26 (1980)
10. Engquist, B., Tsai, R.: Heterogeneous multiscale methods for stiff ordinary differential equations. Math. Comput. **74**, 1707–1742 (2005)
11. Engstler, C., Lubich, C.: Multirate extrapolation methods for differential equations with different time scales. Computing **58**, 173–185 (1997)
12. Ferm, L., Lötstedt, P.: Space–time adaptive solution of first order PDES. J. Sci. Comput. **26**, 83–110 (2006)
13. Ferracina, L., Spijker, M.: Stepsize restrictions for total-variation-boundedness in general Runge–Kutta procedures. Appl. Numer. Math. **53**, 265–279 (2005)
14. Gear, C., Wells, D.: Multirate linear multistep methods. BIT **24**, 484–502 (1984)
15. Godunov, S.: A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sb. **47**, 271–290 (1959)
16. Goodman, J., LeVeque, R., Randall, J.: On the accuracy of stable schemes for 2D scalar conservation laws. Math. Comput. **45**, 15–21 (1985)
17. Gottlieb, S., Shu, C.-W.: Total variation diminishing Runge–Kutta schemes. Math. Comput. **67**, 73–85 (1998)
18. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. SIAM Rev. **43**, 89–112 (2001)
19. Günther, M., Hoschek, M.: ROW methods adapted to electric circuit simulation packages. In: ICCAM '96: Proceedings of the 7th International Congress on Computational and Applied Mathematics, pp. 159–170. Elsevier, Amsterdam (1997)

20. Günther, M., Kværnø, A., Rentrop, P.: Multirate partitioned Runge–Kutta methods. BIT **41**, 504–514 (2001)
21. Günther, M., Rentrop, P.: Multirate ROW-methods and latency of electric circuits, Appl. Numer. Math. (1993)
22. Hairer, E.: Order conditions for numerical methods for partitioned ordinary differential equations. Numer. Math. **36**, 431–445 (1981)
23. Hairer, E., Norsett, S., Wanner, G.: Solving Ordinary Differential Equations I. Nonstiff Problems. Springer, Berlin (1993), Chapter II. 1
24. Hairer, E., Norsett, S., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems. Springer, Berlin (1993) Chapter. II. 15
25. Harten, A.: High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. **49**, 357–393 (1983)
26. Higueras, I.: On strong stability preserving time discretization methods. J. Sci. Comput. **21**, 193–223 (2004)
27. Hundsdorfer, W., Koren, B., van Loon, M.: A positive finite-difference advection scheme. J. Comput. Phys. **117**, 35–46 (1995)
28. Hundsdorfer, W., Ruuth, S., Spiteri, R.: Monotonicity-preserving linear multistep methods. SIAM J. Numer. Anal. **41**, 605–623 (2003)
29. Kato, T., Kataoka, T.: Circuit analysis by a new multirate method. Electr. Eng. Jpn. **126**, 55–62 (1999)
30. Kirby, R.: On the convergence of high resolution methods with multiple time scales for hyperbolic conservation laws. Math. Comput. **72**, 1239–1250 (2002)
31. Kværnø, A.: Stability of multirate Runge–Kutta schemes. Int. J. Differ. Equ. Appl. **1**, 97–105 (2000)
32. Kvæ rnø, A., Rentrop, P.: Low order multirate Runge–Kutta methods in electric circuit simulation (1999)
33. Laney, C.: Computational Gasdynamics. Cambridge University Press, Cambridge (1998)
34. LeVeque, R.: Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge (2002)
35. Logg, A.: Multi-adaptive Galerkin methods for ODEs I. SIAM J. Sci. Comput. **24**, 1879–1902 (2003)
36. Logg, A.: Multi-adaptive Galerkin methods for ODEs II: Implementation and applications. SIAM J. Sci. Comput. **25**, 1119–1141 (2003)
37. Logg, A.: Multi-adaptive Galerkin methods for ODEs III: A priory estimates. SIAM J. Numer. Anal. **43**, 2624–2646 (2006)
38. Osher, A., Chakravarthy, S.: High resolution schemes and the entropy condition. SIAM J. Numer. Anal. **21**, 955–984 (1984)
39. Osher, A., Chakravarthy, S.: Very high order accurate TVD schemes. In: Oscillation Theory, Computation, and Methods of Compensated Compactness. IMA Vol. Math. Appl., vol. 2, pp. 229–274. Springer, Berlin (1986)
40. Rice, J.: Split Runge–Kutta methods for simultaneous equations, J. Res. National Inst. Standards Technol. **60** (1960)
41. Roe, P.: Approximate Riemann solvers, parameter vectors and difference schemes. J. Comput. Phys. **43**, 357–372 (1981)
42. Sand, J., Burrage, K.: A Jacobi waveform relaxation method for ODEs. SIAM J. Sci. Comput. **20**, 534–552 (1998)
43. Savcenco, V., Hundsdorfer, W., Verwer, J.: A multirate time stepping strategy for parabolic PDE. Tech. Report MAS-E0516, Centrum voor Wiskundeen Informatica, 2005
44. Savcenco, V., Hundsdorfer, W., Verwer, J.: A multirate time stepping strategy for stiff ODEs. BIT (2006, to appear)
45. Shampine, L.: Conservation laws and the numerical solution of ODEs. Comput. Math. Appl. B **12**, 1287–1296 (1986)
46. Shu, C.-W.: Total-variation-diminishing time discretizations. SIAM J. Sci. Stat. Comput. **9**, 1073–1084 (1988)
47. Shu, C.-W., Osher, S.: TVB uniformly high-order schemes for conservation laws. Math. Comput. **49**, 105–121 (1987)
48. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
49. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes II. J. Comput. Phys. **83**, 32–78 (1989)
50. Spiteri, R., Ruuth, S.: A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. **40**, 469–491 (2002)
51. Sweby, P.: High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J. Numer. Anal. **21**, 995–1011 (1984)

52. Tang, H.-Z., Warnecke, G.: A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and space grids. J. Comput. Math. **24**, 121–140 (2006)
53. Vreugdenhil, C., Koren, B. (eds.): Numerical Methods for Advection–Diffusion Problems, vol. 45. Vieweg, Wiesbaden (1993)
54. Zalesak, S.: Fully multidimensional flux corrected transport algorithms for fluids. J. Comput. Phys. **31**, 335–362 (1979)