

# Multirate Timestepping Methods for Hyperbolic Conservation Laws

Emil M. Constantinescu and Adrian Sandu

---

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061. E-mail: {emconsta, sandu}@cs.vt.edu

---

## Abstract

This paper constructs multirate time discretizations for hyperbolic conservation laws that allow different time-steps to be used in different parts of the spatial domain. The discretization is second order accurate in time and preserves the conservation and stability properties under local CFL conditions. Multirate timestepping avoids the necessity to take small global time-steps (restricted by the largest value of the Courant number on the grid) and therefore results in more efficient algorithms.

---

## 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. Stability requirements limit the temporal step size, with the upper bound being determined by the ratio of the mesh spacing and the magnitude of the wave speed. Local spatial refinement reduces the allowable time-step for the explicit time discretizations. The time-step 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.

One possibility to circumvent this restriction is to use implicit, unconditionally stable timestepping algorithms which allow large global time-steps. 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 implicit schemes unless the time-step is restricted by a CFL-like condition.

In this work we seek to develop multirate time integration schemes for the simulation of PDEs. In this approach, the time-step can vary spatially and has to satisfy the CFL condition only locally, resulting in substantially more efficient computations. The approach follows 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 which timestepping schemes need to satisfy. The algorithms used in the solution of conservation laws need to preserve the system invariants. Moreover, the solutions to hyperbolic PDEs may not be 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 nonphysical behavior (spurious oscillations, etc.)

A zooming technique for wind transport of air pollution discussed in [Berkvens et al., 1999] is a positive, conservative finite volume discretization that allows the use of smaller time-steps 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 [Dawson and Kirby, 2001; Kirby, 2002] 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 [2003] reformulated Dawson and Kirby’s algorithm in terms of solution increments to obtain second order consistency in time for two-rate integration.

In this paper we develop a general systematic approach to extend Runge-Kutta (RK) to multirate (variable step size) Runge-Kutta methods that inherit the strong stability properties of the corresponding single rate integrators. Additionally, the order of accuracy of the overall scheme is preserved, unlike previous multirate approaches that lead to first order accuracy due to the interface treatment [Kirby, 2002; Berkvens et al., 1999]. Moreover, for conservative laws, we show that this multirate approach preserves the linear invariants (i.e. is also conservative).

This paper is structured as follows: In Section 2 we review the main properties and issues of modeling 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 Section 4. Conclusions and future research directions are given in Section 5.

## 2 Hyperbolic Conservation Laws

We consider the following generic one-dimensional scalar hyperbolic equation

$$\frac{\partial y(t, x)}{\partial t} + \frac{\partial f(y(t, x))}{\partial x} = 0, \quad \text{with } y(0, x) = y^0(x), \quad (1)$$

on  $x \in \Omega^0 \subset (-\infty, \infty), t > 0$ .

The space discretization is usually applied to the equation in conservative form. In the one dimensional finite volume approach, the change in the mean quantity in the  $i^{\text{th}}$  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

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

where  $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ , ( $x_i = x(i)$ ) is the grid spacing, and  $F_{i+\frac{1}{2}} = F(x_{i+k-1}, \dots, x_{i-k})$  is the numerical flux at the control volume face.

Exact solutions of hyperbolic problems have a range-diminishing property that forbids existing maxima from increasing, existing minima from decreasing, and new maxima or minima from forming. To provide physically meaningful solutions and avoid weak non-linear instabilities (spurious oscillations), the numerical solution has to satisfy a stability condition. The following are some properties of the numerical solution of conservation laws that define the stability of the scheme in some sense and are used throughout this paper.

**Maximum principle.** Exact solutions of hyperbolic problems have a range-diminishing property that forbids existing maxima from increasing, existing minima from decreasing, and new maxima or minima from forming. Formally, it can be written as

$$\max(y(t > 0, x)) \leq \max(y(t = 0, x)). \quad (3)$$

**TVD.** A numerical method that is called total variation diminishing (TVD) [Harten, 1997] if

$$\|y(t + \Delta t, x)\| \leq \|y(t, x)\|, \quad \|y\| = \sum_{i \in \Omega^0} |y_{i+1} - y_i|, \quad (4)$$

where  $\|\circ\|$  is the total variation semi-norm. No spurious spatial oscillations are introduced during time-stepping with TVD methods.

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

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

$$y_i(t) = 0 \text{ and } y_j(t) > 0 \text{ for } \forall j \neq i \Rightarrow y'_i \geq 0. \quad (5)$$

We note that the forward Euler method used for time integration maintains all the above stability properties; however, it has strong CFL restrictions and is only first order

accurate. In this paper we use combinations of forward Euler methods in order to preserve the stability properties while increasing the order of accuracy and alleviating the CFL restrictions.

Several methods that approximate the fluxes in (2) have been developed in the past decades. Godunov’s method [Godunov, 1959] is based on the exact solution of Riemann problems. The flux-corrected transport method proposed by Boris and Book [1997] and further developed by Zalesak [1979] and Roe [1981] established the basic principles for the construction of high resolution methods. Upwind biased interpolation is coupled with limiters [Sweby, 1984] 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 [Harten, 1997] for nonlinear scalar one dimensional problems.

All these spatial discretization methods satisfy some of the stability properties mentioned above (maximum principle, TVD, monotonicity-preservation, positivity). Explicit Euler and later convex combinations of explicit Euler methods are used to solve the semi-discrete form (2), which under the CFL-like condition maintains the stability properties of the spatial discretization. Implicit methods are linearly stable; however, the nonlinear stability properties restrict the integration time-step to the CFL-like condition. Moreover, the implicit methods require the solution of (non)linear systems. 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 their 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 [Hairer et al., 1993a] computes the next step solution  $y^1$  (at time  $t^1 = t^0 + \Delta t$ ) from the current solution  $y^0$  at  $t^0$  using the formula:

$$y^1 = y^0 + \Delta t \sum_{i=1}^s b_i K_i, \quad (6)$$

$$K_i = f \left( t + \sum_{j=1}^{i-1} a_{ij} \Delta t, y^0 + \Delta t \sum_{j=1}^{i-1} a_{ij} K_j \right).$$

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 [Hairer et al., 1993a]

$$\begin{array}{c|cccc}
 c_1 = 0 & 0 & & & \\
 c_2 & a_{21} & & & \\
 c_3 & a_{31} & a_{32} & & \\
 \vdots & \vdots & \vdots & \ddots & \\
 c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\
 \hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
 \end{array} \quad (7)$$

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

$$\text{Order I : } \quad \sum_{i=1}^s b_i = 1 \quad (8)$$

$$\text{Order II : } \quad \sum_{i=1}^s \sum_{j=1}^{s-1} b_i a_{i,j} = \sum_{i=1}^s b_i c_i = \frac{1}{2} \quad (9)$$

Next, we discuss some stability properties of Runge-Kutta methods, namely, strong stability preserving.

## 2.2 Strong Stability Preservation

Strong stability preserving (SSP) integrators are high order timestepping schemes that preserve the stability properties of the spatial discretization used with explicit Euler timestepping. Spurious oscillations (nonlinear instabilities) can occur in a numerical solution obtained with a TVD or MUSCL spatial discretization scheme, but with a standard (linearly stable) timestepping scheme [Gottlieb et al., 2001]. When TVD discretizations are combined with SSP timestepping, the numerical solution does 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 with a certain CFL number, higher-order methods can be constructed as convex combinations of forward Euler steps with various step sizes [Shu and Osher, 1988, 1989]. SSP methods preserve the strong stability of the forward Euler scheme for bounded time-steps.

Gottlieb et al. [2001] 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. [2003] provide an analysis of monotonicity properties for linear multistep methods, and Spiteri and Ruuth [2002] extend the SSP Runge-Kutta class of methods.

Several examples of SSP Runge-Kutta are given in [Shu and Osher, 1988], we consider a second order method denoted RK2a with its Butcher tableau shown below

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \quad \begin{array}{l} K_1 = f(y^0), \quad y^A = y^0 + \Delta t K_1 \\ K_2 = f(y^A) \\ y^1 = y^0 + \frac{\Delta t}{2} (K_1 + K_2) \end{array} \quad (10)$$

Using the following notation to compactly denote Euler steps

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

this method can be written in convex combinations Euler steps, and thus is proven to be SSP by Shu and Osher [1988]:

$$\begin{aligned}
y^1 &= y^0 + \frac{\Delta t}{2} (K_1 + K_2) \\
&= \frac{1}{2}y^0 + \frac{1}{2}y^A + \frac{1}{2}(y^A + \Delta t K_2) & y^A &= y^0 + \Delta t f(y^0) = \mathcal{E}(\Delta t, y^0) \\
&= \frac{1}{2}y^0 + \frac{1}{2}y^A + \frac{1}{2}y^{A*} & y^{A*} &= y^A + \Delta t f(y^A) = \mathcal{E}(\Delta t, y^A) \\
&= \frac{y^0 + y^B}{2} & y^B &= \frac{y^A + y^{A*}}{2} = \frac{\mathcal{E}(\Delta t, y^0) + \mathcal{E}(\Delta t, y^A)}{2} \tag{11}
\end{aligned}$$

### 3 Multirate Time Integration

The idea of multirate timestepping is to take different time-steps for different components to achieve the target accuracy. The time-steps at different levels need to be synchronized in order to obtain the desired overall accuracy.

Early efforts to develop multirate Runge-Kutta methods are due to Rice [1960] and Andrus [1979, 1993]. Multirate versions of many of the traditional timestepping schemes have been proposed in the literature. Such methods include linear multistep [Gear and Wells, 1984; Kato and Kataoka, 1999], extrapolation [Engstler and Lubich, 1997], Runge-Kutta [Kværnø and Rentrop, 1999; Kværnø, 2000; Günther et al., 1998], Rosenbrock-Wanner [Günther and Rentrop, 1993; Bartel and Günther, 2002], waveform relaxation [Sand and Burrage, 1998], Galerkin [Logg, 2003a,b, 2006, 2004], and combined multi-scale approaches [Engquist and Tsai, 2005]. For the majority of methods, slower components are integrated using larger step sizes, integer multiples of the smaller step sizes used for fast components,  $\Delta t_{\text{slow}} = m\Delta t_{\text{fast}}$ . All steps are synchronized every largest time-step  $\Delta t_{\text{slow}}$ . Conditions for high orders of accuracy for the multirate integrators (at the synchronization times) are derived in the literature.

Kværnø and Rentrop [1999]; Kværnø [2000] developed multirate Runge Kutta (MRK) methods where the coupling between slow and fast components is done by intermediate stage values. Günther et al. [1998] developed multirate partitioned Runge-Kutta (MPRK) schemes which generalize both partitioned Runge-Kutta and multirate ROW methods [Günther and Hoschek, 1997], and Sand et. al. developed the Jacobi waveform relaxation approach [Sand and Burrage, 1998].

For the purpose, of simplicity without the loss of generality, in the following sections we restrict our discussion to scalar one dimensional equations. Multidimensional/multivariable extensions are straightforward.

Consider the system of ordinary differential equations in (2) resulting from the application of MOL to (1) where a partitioning of variables according to their characteristic times is possible:

$$y = [y_1, \dots, y_M]^T, \quad y'_i = f_i(t, y_1, \dots, y_M), \quad i = 1, \dots, M. \tag{12}$$

The M subsystems have characteristic times that vary considerably between the slowest ( $y_1$ ) and the fastest components ( $y_M$ ). Typically, a small number of fast changing components (or a small number of grid points on the very fine grids) restrict the overall step size of the integration.

Parallel to the above partitioning, consider the following nested domain decomposition of  $y$  defined in (1) and (12),  $-\infty < \Omega^0 < \infty$ :

$$\Omega^0 = (-\infty <) \Omega_L^{0,S} \cup \Omega_L^{0,I} \cup \Omega^{0,F} \cup \Omega_R^{0,I} \cup \Omega_R^{0,S} (< \infty), \quad (13)$$

with an associated “slow characteristic time” left and right subdomains,  $\Omega_{\{L,R\}}^S$ , a “fast characteristic time” subdomain  $\Omega^F$ , and left and right interfaces between the two associated subdomain types,  $\Omega_{\{L,R\}}^I$ . Further partitioning of  $\Omega^{0,F} \rightarrow \dots \rightarrow \Omega^j$ ,  $j > 0$  is possible:

$$\Omega^j = (\Omega_L^{j-1,I} \cup) \Omega_L^{j,S} \cup \Omega_L^{j,I} \cup \Omega^{j,F} \cup \Omega_R^{j,I} \cup \Omega_R^{j,S} (\cup \Omega_R^{j-1,I}). \quad (14)$$

This process can be applied recursively until the characteristic time requirements for the components  $y_{1..M}$  are satisfied by the associated characteristic time of the domains  $\Omega^{0..k}$ ,  $k \geq 0$ .

The nested domain decomposition described above can be reflected on the solution components in (12) such that we have

$$y'_i = f_i(t, y(x)) , \quad x \in \{ \Omega_L^{j,I} \cup \Omega^{j,F} \cup \Omega_R^{j,I} \} , \quad j \geq 0 , \quad \text{and} \quad (15)$$

$\Omega^{j,F}$ 's resolution “satisfies” the requirements of  $f_i$  (e.g. CFL condition for  $\Omega^{j,F}$  is satisfied for the wave speed of  $f_i(t, y_i)$ ). For  $j = 0$ , the left and right interfaces become the boundary conditions. The spatial numerical discretization of the flux function,  $f_i$ , extends the domain of dependence for  $y_i$  with the left and right parts of the spatial discretization stencil.

We note that the recurrent nesting partitioning (14) decouples the estimation of  $f$  exemplified in (12) in

$$\begin{aligned} (y_k^{S,L})' &= f_k(t, y(x)) , \quad x \in \{ \Omega_R^{j-1,I} \cup \Omega_L^{j,S} \cup \Omega_L^{j,I} \} \\ (y_k^F)' &= f_k(t, y(x)) , \quad x \in \{ \Omega_L^{j,I} \cup \Omega^{j,F} \cup \Omega_R^{j,I} \} , \quad j \geq 0, \\ (y_k^{S,R})' &= f_k(t, y(x)) , \quad x \in \{ \Omega_R^{j,I} \cup \Omega_R^{j,S} \cup \Omega_R^{j-1,I} \} \end{aligned} \quad (16)$$

where  $\Omega^j$ 's resolution “satisfies” the requirements of  $f_k$ .

In this manner, an efficient domain partitioning with an associated characteristic time that satisfies the requirements of the solution in the corresponding partition is achieved. We shall refer to  $\Omega$ 's first superscript index as “level” and denote the subdomain slow (S) – fast (F) time ratio on the same level with  $m$ .

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

### 3.1 Partitioned Runge-Kutta Methods

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

$$\begin{pmatrix} y_F \\ y_S \end{pmatrix}' = \begin{pmatrix} f_F(y_F, y_S) \\ f_S(y_F, y_S) \end{pmatrix} \quad (17)$$

Partitioned Runge-Kutta (PRK) methods [Hairer et al., 1993b; Hairer, 1981] are used to solve the problem (17) 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 coefficients of these methods are  $A^F = \{a_{ij}^F\}$ ,  $b^F, c^F$ , and  $A^S = \{a_{ij}^S\}, b^S, c^S$ . The PRK solution method reads [Hairer et al., 1993b; Hairer, 1981]

$$\begin{aligned}
y_F^1 &= y_F^0 + \Delta t \sum_{i=1}^S b_i^F K_F^i, & y_S^1 &= y_S^0 + \Delta t \sum_{i=1}^S b_i^S K_S^i \\
Y_F^i &= y_F^0 + \Delta t \sum_{j=1}^S a_{ij}^F K_F^j, & Y_S^i &= y_S^0 + \Delta t \sum_{j=1}^S a_{ij}^S K_S^j \\
K_F^i &= f_F(Y_F^i, Y_S^i), & K_S^i &= f_S(Y_F^i, Y_S^i).
\end{aligned} \tag{18}$$

The order of the coupled method is the minimum order among each method taken separately and the order of the ‘‘coupling’’. The first order coupling conditions are implicitly satisfied. The second order coupling conditions are

$$\sum_{i=1}^S b_i^F c_i^S = \frac{1}{2} \qquad \sum_{i=1}^S b_i^S c_i^F = \frac{1}{2}. \tag{19}$$

There are over 20 third order coupling conditions that can be found in [Hairer et al., 1993b]. Here, we list just two that will be used later in this paper

$$\sum_{i=1}^S b_i^F c_i^F c_i^S = \frac{1}{6} \qquad \sum_{i=1}^S b_i^S c_i^S c_i^F = \frac{1}{6}. \tag{20}$$

### 3.2 Strong Stability Preservation and PRK

The SSP schemes are central to this paper and their stability properties are also required in the PRK methods applied to a partitioning of type (16). An immediate condition is that both  $\mathcal{RK}^F$  and  $\mathcal{RK}^S$  need to be SSP methods since  $\mathcal{RK}^F$  is SSP for  $F_F$  (on  $\Omega^F$ ), and  $\mathcal{RK}^S$  is SSP for  $f_S$  (on  $\Omega^S$ ). A natural question is the treatment of the interface partition,  $\Omega^I$ .

Equation (16) shows the inter-dependency of the (flux) function on the solution in adjacent partitions. From (16), function  $f$  on  $\Omega^S$  depends on part of the solution that is SSP (on  $\Omega^S$ ), and on part that is SSP in  $\Omega^F$ . Since  $\mathcal{RK}^F$  applied in the slow partition,  $\Omega^S$ , is SSP, it implies that  $\mathcal{RK}^F$  applied in  $\Omega^I$  guarantees that the method is SSP in  $\Omega^I$ .

To summarize, the partitioned Runge-Kutta method applied in our context (16) is SSP if both methods ( $\mathcal{RK}^S$  and  $\mathcal{RK}^F$ ) are SSP on their respective partition and, in addition,  $\mathcal{RK}^F$  is applied on the interface ( $\Omega^I$ ). The latter is SSP by construction and requirements in (15). If each of the partitions is SSP and the interface partitions use the fast method at their corresponding level, then the resulting method is SSP on  $\Omega^0$ .

Additional stability-like requirements TVD, positivity, etc., of the full method need to be addressed by each pair of spatial and temporal discretization. Our experiments



$\mathcal{RK}^B$	$\mathcal{RK}^F$	$\mathcal{RK}^S$
$\begin{array}{c c} c & A \\ \hline & b \end{array}$	$\begin{array}{c cc} \frac{1}{m}c & \frac{1}{m}A & \\ \frac{1}{m}\mathbb{1} + \frac{1}{m}c & \frac{1}{m}\mathbb{1}b^T & \frac{1}{m}A \\ \vdots & \vdots & \ddots \\ \frac{m-1}{m}\mathbb{1} + \frac{1}{m}c & \frac{1}{m}\mathbb{1}b^T & \dots \quad \frac{1}{m}\mathbb{1}b^T \quad \frac{1}{m}A \end{array}$ <hr style="border: 0.5px solid black;"/> $\begin{array}{cccc} \frac{1}{m}b & \frac{1}{m}b & \dots & \frac{1}{m}b \end{array}$	$\begin{array}{c ccc} c & A & & \\ c & & A & \\ \vdots & & & \ddots \\ c & & & A \end{array}$ <hr style="border: 0.5px solid black;"/> $\begin{array}{cccc} \frac{1}{m}b & \frac{1}{m}b & \dots & \frac{1}{m}b \end{array}$
(a) Base	(b) Fast method	(c) Slow method

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

indicate that multirate schemes constructed based on SSP time discretizations preserve the particular stability features of the spatial discretization. We will further discuss this in the results section.

### 3.3 A Second Order Multirate PRK Family

Based on the PRK setting and discussion from 3.1, we propose the following generic family of second order multirate partitioned Runge-Kutta (MPRK2) schemes.

Consider a second order accurate SSP RK “base” method (RK2a, for instance)  $\mathcal{RK}^B = [A, b, c]$  (Table 1.a). 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  $b, b^F = b^S$  ( $\frac{1}{m}b$  repeated  $m$  times).

Note that the slow method repeats the same stages  $m$  times. The dependence of  $f_S$  on  $y_F$  is exhibited only in the interface region ( $\Omega^I$ , with length equal with the spatial stencil). The same is true for the fast method, i.e.  $f_F$  does not depend on  $y_S$  away from the interface. Moreover, inside the slow domain ( $\Omega^S$ ), the slow method reduces to the base method ( $\mathcal{RK}^S \rightarrow \mathcal{RK}^B$ ), resulting in large steps ( $\Delta t$ ) since the stages are just repeated, with  $y_F$  being zero (or  $f_S(y_S, 0)$ ). The fast method takes small steps of  $\Delta t/m$ .

**Proposition 3.1 (MPRK2)** *The partitioned Runge-Kutta methods described by the Butcher tableau in Table 1 are*

- a) 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 (8) we have

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

The second order conditions (9) are also satisfied

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

and for  $\mathcal{RK}^F$  we have

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

Since  $b^F = b^S$ , the coupling conditions (19) are satisfied directly by the above. Hence, MPRK2 is at least second order accurate.

b) Consider that  $\mathcal{RK}^B$  satisfies the third order accuracy conditions, then the third order coupling condition (20) requires the following

$$\begin{aligned} b_F^T c_F c_S &= b_F^T \left[ \frac{1}{2} c^2 \right] \\ &= \frac{1}{4} b^T c^2 + \frac{1}{4} (b^T c + b^T c^2) \\ &= \frac{1}{4} \cdot \frac{1}{3} + \frac{1}{4} \left( \frac{1}{2} + \frac{1}{3} \right) \\ &= \frac{7}{24} \neq \frac{1}{6}, \end{aligned}$$

and thus, (at the interface) the method reduces to second order coupling accuracy.

■

The method presented in this section represents a truly multirate approach since the fast method takes  $m$  successive steps of the base method with a time-step of  $\Delta t/m$ . This method can be easily extended from  $m = 2$  to  $m = 3$ . In Appendix B we present the same method for  $m = 3$ .

In order to increase the coupling order, we need to investigate other schemes that have a different layout (perhaps using different base methods for the fast and for the slow subsystems). Such methods will be investigated in future studies.

**Proposition 3.2 (Conservation)** *The partitioned Runge-Kutta methods described by the Butcher tableau in Table 1 preserve the linear invariants of the system.*

**Proof** This is a direct consequence of having chosen equal weights for the fast and for the slow methods,  $b^F = b^S$ . Consider the system (17) with a linear invariant of the form

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

where  $e_F, e_S$  are fixed weight vectors.

From the method (18) with  $b^F = b^S = b^*$  we have that

$$y_F^1 = y_F^0 + \Delta t \sum_{i=1}^S b_i^* f_F(Y_F^i, Y_S^i), \quad y_S^1 = y_S^0 + \Delta t \sum_{i=1}^S b_i^* f_S(Y_F^i, Y_S^i)$$

and therefore

$$e_F y_F^1 + e_S y_S^1 = e_F y_F^0 + e_S y_S^0 + \Delta t \sum_{i=1}^S b_i^* \underbrace{\left( e_F f_F(Y_F^i, Y_S^i) + e_S f_S(Y_F^i, Y_S^i) \right)}_0 = e_F y_F^0 + e_S y_S^0.$$

■

This property is important because multirate Runge-Kutta methods, used in conjunction with conservative space discretizations, lead to conservative full discretizations of the PDE. For example, consider a one-dimensional finite volume scheme in the conservative formulation:

$$y'_i = \frac{1}{\Delta x_i} (F_{i-\frac{1}{2}}(y) - F_{i+\frac{1}{2}}(y)), \quad 1 \leq i \leq N.$$

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

$$\sum_{i=1}^N \Delta x_i y'_i = 0, \quad \sum_{i=1}^N \Delta x_i y_i = \text{const}.$$

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  $\ell$  grid cells are the fast domain ( $y_F = \{y_1, \dots, y_\ell\}$ ), and the remaining cells are the slow domain  $y_S = \{y_{\ell+1}, \dots, y_N\}$ . Each subdomain is advanced in time with a classical Runge-Kutta method, therefore the fluxes exchanged between the boundaries of same-class 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.

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

(a) Base method
(b) Fast method
(c) Slow method

Table 2: Order 2 Butcher tableau for the two, slow and fast methods for  $m = 2$ .

From the multirate Runge-Kutta formula

$$y_\ell^1 = y_\ell^0 + \frac{\Delta t}{\Delta x_\ell} \sum_{i=1}^S b_i^F (F_{\ell-\frac{1}{2}}(Y^i) - F_{\ell+\frac{1}{2}}(Y^i))$$

the total flux lost by the fast domain during one time-step through the fast-slow interface is

$$\Delta t \sum_{i=1}^S b_i^F F_{\ell+\frac{1}{2}}(Y^i).$$

Similarly

$$y_{\ell+1}^1 = y_{\ell+1}^0 + \frac{\Delta t}{\Delta x_{\ell+1}} \sum_{i=1}^S b_i^S (F_{\ell+\frac{1}{2}}(Y^i) - F_{\ell+3/2}(Y^i))$$

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

$$\Delta t \sum_{i=1}^S b_i^S F_{\ell+\frac{1}{2}}(Y^i).$$

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,  $b_i^S = b_i^F$ .

Finally, we note that conservation is achieved without explicitly storing fluxes at the fast-slow interface, as it was suggested in earlier works Kirby [2002].

### 3.4 A Second Order SSP PRK Method with $m = 2$

In this section we consider the SSP Runge-Kutta RK2a [Shu and Osher, 1988] as the base method in Table 2.a, and extend it to a multirate method with  $m = 2$  using the approach described in Section 3.3. The Butcher tableau for the fast and slow methods for  $m = 2$  is described in Table 2. The RK stages are computed as follows:

$$\begin{aligned} K_F^1 &= f_F(y_F^0, y_S^0) & K_S^1 &= f_S(y_F^0, y_S^0) \\ y_F^A &= y_F^0 + \frac{\Delta t}{2} K_F^1 & y_S^A &= y_S^0 + \Delta t K_S^1 \end{aligned}$$

$$\begin{aligned}
K_F^2 &= f_F(y_F^A, y_S^A) & K_S^2 &= f_S(y_F^A, y_S^A) \\
y_F^B &= y_F^0 + \frac{\Delta t}{4}K_F^1 + \frac{\Delta t}{4}K_F^2 & y_S^B &= y_S^0 \\
K_F^3 &= f_F(y_F^B, y_S^0) & K_S^3 &= f_S(y_F^B, y_S^0) \\
y_F^C &= y_F^B + \frac{\Delta t}{2}K_F^3 & y_S^C &= y_S^0 + \Delta t K_S^3 \\
K_F^4 &= f_F(y_F^C, y_S^C) & K_S^4 &= f_S(y_F^C, y_S^C) \\
y_F^1 &= y_F^0 + \frac{\Delta t}{4}(K_F^1 + K_F^2 + K_F^3 + K_F^4) & y_S^1 &= y_S^0 + \frac{\Delta t}{4}(K_S^1 + K_S^2 + K_S^3 + K_S^4)
\end{aligned} \tag{21}$$

Using the following notation to compactly denote Euler steps

$$\mathcal{E}_{\{F,S\}}(\Delta t, y_F, y_S) := y_{\{F,S\}}(t) + \Delta t \cdot f_{\{F,S\}}(t, y_F, y_S), \tag{22}$$

the above MPRK2 can be written in Euler steps in the following way:

$$\begin{aligned}
y_F^1 &= \frac{1}{2}(y_F^0 + y_F^0 + \frac{\Delta t}{2}K_F^1 + \frac{\Delta t}{2}K_F^2 + \frac{\Delta t}{2}K_F^3 + \frac{\Delta t}{2}K_F^4), & y_F^A &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^0, y_S^0\right) \\
&= \frac{1}{2}(y_F^0 + y_F^A + \frac{\Delta t}{2}K_F^2 + \frac{\Delta t}{2}K_F^3 + \frac{\Delta t}{2}K_F^4), & y_F^{A*} &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^A, y_S^A\right) \\
&= \frac{1}{2}(y_F^0 + y_F^{A*} + \frac{\Delta t}{2}K_F^3 + \frac{\Delta t}{2}K_F^4), & 2y_F^B &= y_F^0 + y_F^{A*} & y_F^C &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^B, y_S^0\right) \\
&= \frac{1}{2}(y_F^B + y_F^C + \frac{\Delta t}{2}K_F^4), & y_F^{C*} &= \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^C, y_S^C\right) \\
&= \frac{1}{2}(y_F^B + y_F^{C*}), & & & & 
\end{aligned} \tag{23}$$

and

$$\begin{aligned}
y_S^1 &= \frac{1}{4}(2y_S^0 + y_S^0 + \Delta t K_S^1 + \Delta t K_S^2 + y_S^0 + \Delta t K_S^3 + \Delta t K_S^4), & y_S^{\{A,C\}} &= \mathcal{E}_S(\Delta t, y_F^{\{0,B\}}, y_S^0) \\
&= \frac{1}{4}(2y_S^0 + y_S^A + \Delta t K_S^2 + y_S^C + hK_S^4), & y_S^{\{A*,C*\}} &= \mathcal{E}_S(\Delta t, y_F^{\{A,C\}}, y_S^{\{A,C\}}) \\
&= \frac{1}{4}(2y_S^0 + y_S^{A*} + y_S^{C*}). & & 
\end{aligned} \tag{24}$$

The Euler steps for the fast and slow methods are summarized in the appendix A in Table 5.

We now assess how this timestepping method preserves the maximum principle and nonlinear stability properties of the discretization. Kirby [2002] has carried out this kind of analysis for the multirate explicit Euler method.

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

$$y_F^1 = \mathcal{E}_F\left(\frac{\Delta t}{m}, y_F^0, y_S^0\right)$$

and each slow multirate Euler step

$$y_S^1 = \mathcal{E}_S(\Delta t, y_F^0, y_S^0)$$

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 multirate forward Euler solutions. Note that the solutions of multirate Euler steps are not necessarily consistent solutions of the PDE, since the fast subsystem and the slow subsystem are advanced with different time-steps, and the solutions are at different times. ■

**Proposition 3.4 (Maximum principle)** *If each fast and each slow multirate Euler steps satisfy the maximum principle then MPRK2 also satisfies the maximum principle.*

**Proof** Based again on the forward Euler properties, a quick inspection of the method (Table 5) shows the following

$$\begin{aligned} \max\{y_F^A, y_S^A\} &\leq \max\{y_F^0, y_S^0\}, & \max\{y_F^{A*}, y_S^{A*}\} &\leq \max\{y_F^A, y_S^A\}, \\ \max\{y_F^B, y_S^0\} &\leq \max\{y_F^0, y_S^0\}, & & \\ \max\{y_F^C, y_S^C\} &\leq \max\{y_F^B, y_S^0\}, & \max\{y_F^{C*}, y_S^{C*}\} &\leq \max\{y_F^C, y_S^C\}, \end{aligned}$$

and thus, clearly  $\max\{y_F^1, y_S^1\} \leq \max\{y_F^0, y_S^0\}$ . ■

In addition to positivity and the maximum principle, we have strong experimental evidence that MPRK2 preserves the TVD property. These properties can also be verified for other  $m$  values. An example is shown in Appendix B for  $m = 3$ .

**Conjecture 3.5 (TVD)** *If each fast and each slow multirate Euler steps are TVD then MPRK2 is also TVD.*

Forms (23) and (24) represent a convex combination of Euler steps using the MPRK2 construction “algorithm” presented in Section 3.3. The method (23, 24) with partitioning (16) is second order accurate in time and SSP (i.e. preserves stability properties of the spatial method).

We again remark that the slow method reduces to the base method inside the slow domain since there is no dependence of  $f_S$  on  $y_F$ .

### 3.5 Order Two MPRK Methods for Multiple Partitions

The method described in the previous section (3.4) can be extended to multiple levels of refinement. In this section we show one example that is presented in Table 3 for three levels of partitioning ( $S$  - slow,  $M$  - medium, and  $F$  - fast). The construction is very simple: Start with the base method for level 0,  $\mathcal{RK}^B$  (in Table 1.a). Then construct the slow method with  $A$ 's on the diagonal. The top left quadrant in Table 3.c becomes the base method for the medium partition, and so on.

$\chi$	$[\mathcal{RK}_1^F]$	$[\mathcal{RK}_0^F]$	$[\mathcal{RK}_1^S]$	$[\mathcal{RK}_0^S]$	$\chi$	
$\frac{1}{4}c$	$\frac{1}{4}A$		$\frac{c}{2}$	$\frac{1}{2}A$	$c$	$A$
$\frac{1+c}{4}$	$\frac{1}{4}b \quad \frac{1}{4}A$		$\frac{c}{2}$	$\frac{1}{2}A$	$c$	$A$
$\frac{2 \cdot 1+c}{4}$	$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}A$		$\frac{1+c}{2}$	$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{2}A$	$c$	$A$
$\frac{3 \cdot 1+c}{4}$	$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}A$		$\frac{1+c}{2}$	$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{2}A$	$c$	$A$
	$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b$			$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b$		$\frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b \quad \frac{1}{4}b$

(a) Fast method ( $m = 4$ )      (b) Medium method ( $m = 2$ )      (c) Slow method ( $m = 1$ )

Table 3: Multirate partitioned Runge-Kutta method (MPRK2) with 3 levels of refinement

For multiple levels, each method depends on its corresponding partition and only on the neighboring (left/right) partitions. 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.

Clearly, the order conditions for each method and for the coupling are satisfied. Moreover, on level 0,  $\mathcal{RK}_0^S$  reduces to the base method on  $\Omega_{0,S}$  (away from the interface), and in turn,  $\mathcal{RK}_1^S$  reduces to the top left quadrant of  $\mathcal{RK}_0^S$  on  $\Omega_{1,S}$ , which becomes the base method for the medium partition. Thus, we have a clear and systematic way to expand methods on increasingly faster partitions.

As stated before in this paper, from the efficiency stand point, there is no additional computational load away from the interface regions which are typically very small compared to the fast and slow partitions.

## 4 Numerical Results

We consider two standard test problems: the advection equation and the simplified (inviscid) Burgers' equation. Since TVD methods in multiple dimensions are at most first order accurate [Goodman et al., 1985], we look at one dimensional problems. Very accurate multiple dimension problems can be implemented using dimension splitting. The solutions are computed using a 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 Section 4.1). Burgers' equation, described in Section 4.2, is implemented on a fixed grid, using a third order scheme developed by Osher and Chakravarthy [1986].

## 4.1 The Advection Equation

The one-dimensional advection equation (25) models the transport of a tracer  $y$  with the constant velocity  $u$  along the  $x$  axis.

$$\frac{\partial y(t, x)}{\partial t} + u \cdot \frac{\partial y(t, x)}{\partial x} = 0. \quad (25)$$

### 4.1.1 Positive Spatial Discretization

In what follows, we describe the positive (5) flux limited spatial discretization scheme [Hundsdorfer et al., 1995; Vreugdenhil and Koren, 1993]. We start by introducing the flux limited formulation of Hundsdorfer et al. [1995] on uniform grids, and extend the scheme to nonuniform grids.

The numerical flux can be defined as

$$F_{i+\frac{1}{2}} = f_i + \frac{1}{2}\phi_{i+\frac{1}{2}}(f_i - f_{i-1}), \quad (26)$$

where  $\phi$  is a nonlinear function called limiter. Then scheme (2) can be written as

$$\frac{\partial}{\partial t} y_i = -\frac{\left(1 + \frac{1}{2}\phi_{i+\frac{1}{2}}\right)(f_i - f_{i-1}) - \frac{1}{2}\phi_{i-\frac{1}{2}}(f_{i-1} - f_{i-2})}{\Delta x}. \quad (27)$$

Define the flux slope ratio

$$r_{i-\frac{1}{2}} = \frac{f_i - f_{i-1}}{f_{i-1} - f_{i-2}}, \quad (28)$$

If we consider  $r_{i-\frac{1}{2}} \neq 0$ , from (27) and (28) we have

$$\frac{\partial}{\partial t} u_i = -\frac{1}{\Delta x} \left[ \left(1 + \frac{1}{2}\phi_{i+\frac{1}{2}}\right) - \frac{\frac{1}{2}\phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}} \right] (f_i - f_{i-1}). \quad (29)$$

The positivity requirement (5) applied to the scheme (29) yields the following condition

$$\frac{\phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}} - \phi_{i+\frac{1}{2}} \leq 2. \quad (30)$$

This condition is used to impose bounds on the limiter  $\phi$  for the numerical flux defined by (26) in order to preserve positivity for the scheme (2).

Now, we extend the scheme and the limiter to a nonuniform grid by redefining the numerical flux. 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,  $\Delta x_{[0]}$ , 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

$$\begin{aligned} \alpha_i &= -\frac{\Delta x_i \Delta x_{i+1}}{(\Delta x_{i-1} + \Delta x_i) (\Delta x_{i+1} + 2 \Delta x_i + \Delta x_{i-1})} \\ \beta_i &= -\frac{\Delta x_i (\Delta x_{i-1} + \Delta x_i - \Delta x_{i+1})}{(\Delta x_{i-1} + \Delta x_i) (\Delta x_i + \Delta x_{i+1})} \\ \gamma_i &= \frac{\Delta x_i (\Delta x_{i-1} + 2 \Delta x_i)}{(\Delta x_i + \Delta x_{i+1}) (\Delta x_{i+1} + 2 \Delta x_i + \Delta x_{i-1})}, \text{ with} \\ \alpha_i + \beta_i + \gamma_i &= 0. \end{aligned}$$

The flux can be written in terms of  $(f_i - f_{i-1})$  and  $r$  as

$$F_{i+\frac{1}{2}} = f_i + \left(-\alpha_i + \gamma_i r_{i+\frac{1}{2}}\right) (f_i - f_{i-1}). \quad (31)$$

Define  $\mathcal{K}$  as

$$\mathcal{K}(r) = 2(-\alpha_i + \gamma_i r). \quad (32)$$

Then, the numerical flux can be expressed as

$$F_{i+\frac{1}{2}} = f_i + \frac{1}{2} \mathcal{K}(r_{i+\frac{1}{2}}) (f_i - f_{i-1}). \quad (33)$$

Just as in [Hundsdorfer et al., 1995; Sweby, 1984], we define the following flux limiter

$$\phi(r) = \max\left(0, \min\left(2r, \min\left(\delta, \mathcal{K}^2(r)\right)\right)\right), \quad (34)$$

and take  $\delta = 2$ .

The semi-discrete form (2) with the limiter (34) using the numerical flux defined by (33) is a positive, second order (wherever the limiter is set to one) semi-discrete scheme on a nonuniform grid. The proofs follow immediately from [Hundsdorfer et al., 1995; Sweby, 1984] with the extension of the nonuniform mesh. In addition, if the timestepping scheme is positivity preserving, then the entire method (each multirate step, in our case) is positivity preserving.

Figure 1 shows the leading order truncation error of the spatial discretization using the unlimited numerical flux (31), i.e. the coefficient that multiplies  $\partial f''' / \partial^3 x$ .

#### 4.1.2 Numerical Experiments

In this section we show a few examples and instances of MPRK2 schemes for the linear advection equation that clearly show one of their applications and potential. The spatial

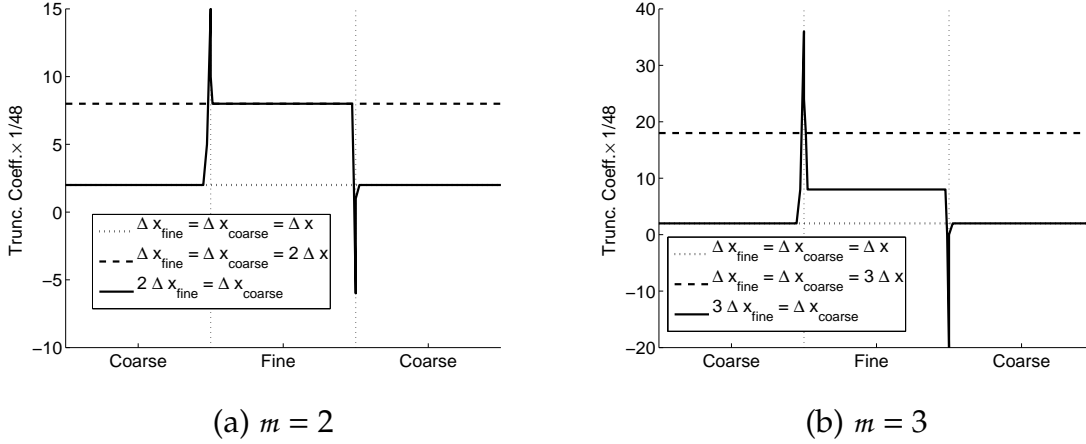


Figure 1: Representation of the discretization leading order error term for two instances of  $m$  as the wave passes through the interfaces.

discretization is positive and the time integration scheme is SSP, which results in an overall positive scheme.

Our test cases include three different function shapes (ordered by their 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  $\Delta x/m$ , while the left and right regions are covered by a coarse mesh with spacing  $\Delta x$ . 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  $\Delta t$  wherever we have  $\Delta x$  grid spacing and  $\Delta t/m$  wherever we have  $\Delta x/m$ .

Figures 2 show the advection equation with the three function profiles that pass through a fixed fine ( $\Delta x/m$ ) 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 time indices). Here, we see that the solution is not qualitatively affected by the interface. Moreover, with the higher spatial resolution, the solution improves qualitatively (as  $m$  is larger), and the wave is not distorted by passing through the interface.

To 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. Figures 3 show the advection equation with the three corresponding initial function profiles (marked with dashed lines) located on the right part of the domain on a fine ( $\Delta x/m$ ) mesh. The initial profile is advected with unit wave speed (to the left part of the domain). The Figures show the final state of the solution with the exact solution superimposed (marked with dotted lines), and the vertical dotted lines delimit the fine domain. Table 4 shows the  $L_1$  error norm of the moving for the profiles shown in Figures 3. Clearly, the solution is improved both qualitatively and quantitatively with higher spatial resolution.

All the results for the advection equation presented in this section show that this

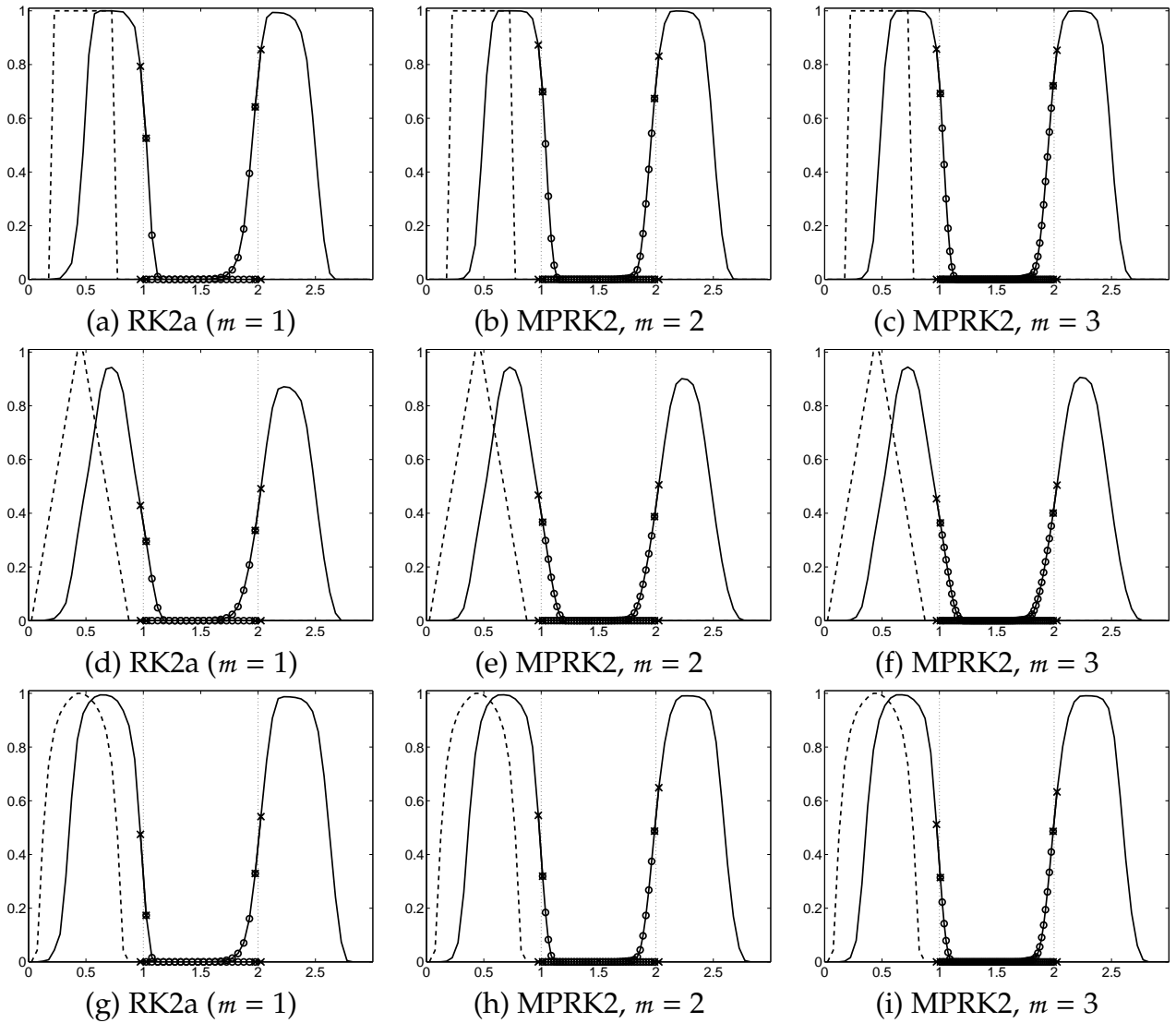


Figure 2: Fixed grid advection equation with three function profiles that pass through a fixed fine ( $\Delta x/m$ ,  $\Delta t/m$ ) region (between 1 and 2). The dashed line represents the exact solution and solid line corresponds to the solution evolved in time (at two different time indices)

Type	$m = 1$	$m = 2$	$m = 3$
Step	0.1085	0.1069	0.1021
Triangular	0.0401	0.0224	0.0154
Exponential	0.0466	0.0344	0.0270

Table 4:  $L_1$  error norm of the moving grid advection equation.

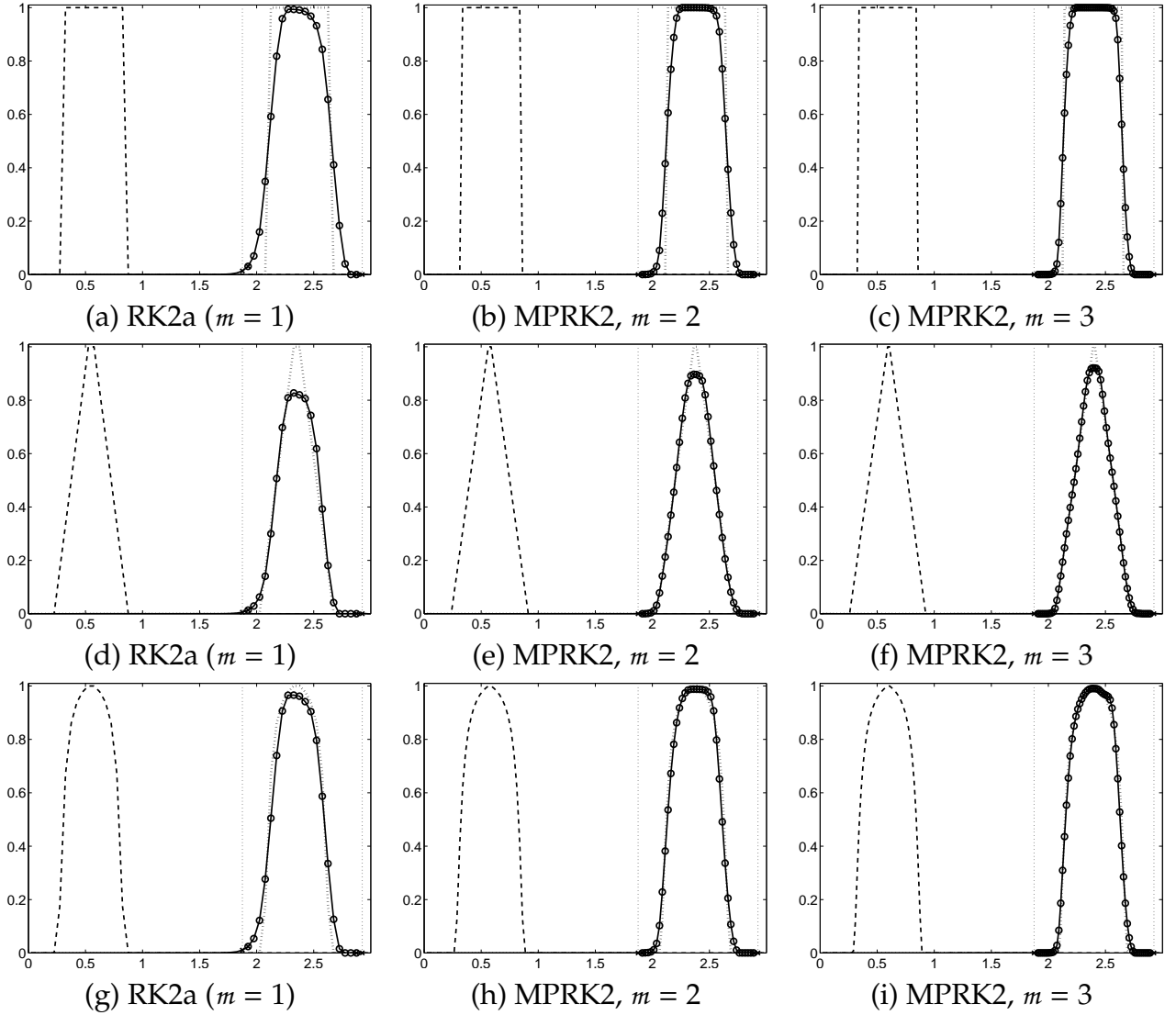


Figure 3: Moving grid advection equation with three function profiles (initially marked with dashed lines) that pass through a fixed fine  $(\Delta x/m, \Delta t/m)$  region (between  $x = 1$  and  $x = 2$ ). The dotted line represents the exact solution and solid line corresponds to the solution evolved in time.

specific finite volume approach and MPRK2 yield a multirate solution on a nonuniform grid that is conservative and positive, as discussed in Section 3.4.

## 4.2 Burgers' equation

The simplified inviscid Burgers equation is described by

$$\frac{\partial y(t, x)}{\partial t} + \frac{\partial}{\partial x} \left( \frac{1}{2} y(t, x)^2 \right) = 0 \quad (35)$$

Burgers' equation numerical experiments are based on the *third order upwind-biased* TVD flux limited scheme described below.

#### 4.2.1 TVD Spatial Discretization

This section is based on the work of Osher and Chakravarthy [1984, 1986]; Chakravarthy and Osher [1983]. A generic recipe for high order TVD finite volume schemes can be found in [Chakravarthy and Osher, 1985]. 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 [Chakravarthy and Osher, 1985]. The following

$$df_{j+\frac{1}{2}}^- = F(y_{j+1}, y_j) - f(y_j), \text{ and} \quad (36)$$

$$df_{j+\frac{1}{2}}^+ = f(y_{j+1}) - F(y_{j+1}, y_j), \quad (37)$$

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

With (36, 37), consider the following numerical flux

$$F_{j+\frac{1}{2}} = F(y_{j+1}, y_j) - \left[ \frac{1-\kappa}{4} \widetilde{df}_{j+\frac{3}{2}}^- + \frac{1+\kappa}{4} \overline{df}_{j+\frac{1}{2}}^- \right] + \left[ \frac{1+\kappa}{4} \widetilde{df}_{j+\frac{1}{2}}^+ + \frac{1-\kappa}{4} \overline{df}_{j-\frac{1}{2}}^+ \right], \quad (38)$$

$$F(y_{j+1}, y_j) = \frac{1}{2} \left( f(y_{j+1}) + f(y_j) \right) - \frac{1}{2} \left( df_{j+\frac{1}{2}}^+ + df_{j+\frac{1}{2}}^- \right), \quad (39)$$

where  $f^\pm$  are the negative and positive flux contributions,  $\widetilde{df}^\pm$  and  $\overline{df}^\pm$  show that they are in flux limited form and are defined below. The scheme defined by (38) is called a  $\kappa$  scheme. If  $\kappa = \frac{1}{3}$ , (38) becomes the limited third order upwind-biased scheme. If we consider  $\overline{df}^\pm = df^\pm$  and  $\widetilde{df}^\pm = df^\pm$ , we have the unlimited scheme. The limited fluxes are defined as follows

$$\widetilde{df}_{j+\frac{3}{2}}^- = \text{minmod} \left[ df_{j+\frac{3}{2}}^-, b df_{j+\frac{1}{2}}^- \right], \quad \overline{df}_{j+\frac{1}{2}}^- = \text{minmod} \left[ df_{j+\frac{1}{2}}^-, b df_{j+\frac{3}{2}}^- \right], \quad (40)$$

$$\widetilde{df}_{j+\frac{1}{2}}^+ = \text{minmod} \left[ df_{j+\frac{1}{2}}^+, b df_{j-\frac{1}{2}}^+ \right], \quad \overline{df}_{j-\frac{1}{2}}^+ = \text{minmod} \left[ df_{j-\frac{1}{2}}^+, b df_{j+\frac{1}{2}}^+ \right], \quad (41)$$

where

$$\text{minmod} [x, y] = \text{sign}(x) \cdot \max [0, \min [|x|, y \text{sign}(x)]]], \quad 1 \leq b \leq \frac{3-\kappa}{1-\kappa}. \quad (42)$$

The semi-discrete form (2) using the numerical flux defined by (40) is a TVD, third order accurate scheme for  $\kappa = \frac{1}{3}$  (when the limiter is not "active", otherwise the order is degraded). Additional information can be found in [Chakravarthy and Osher, 1985].

#### 4.2.2 Numerical Experiments

The numerical examples showed in this section explore the application of varying time-steps on different regions of the domain for a TVD scheme that approximates nonlinear hyperbolic conservation laws, in order to avoid the CFL limitation (time-step restriction)

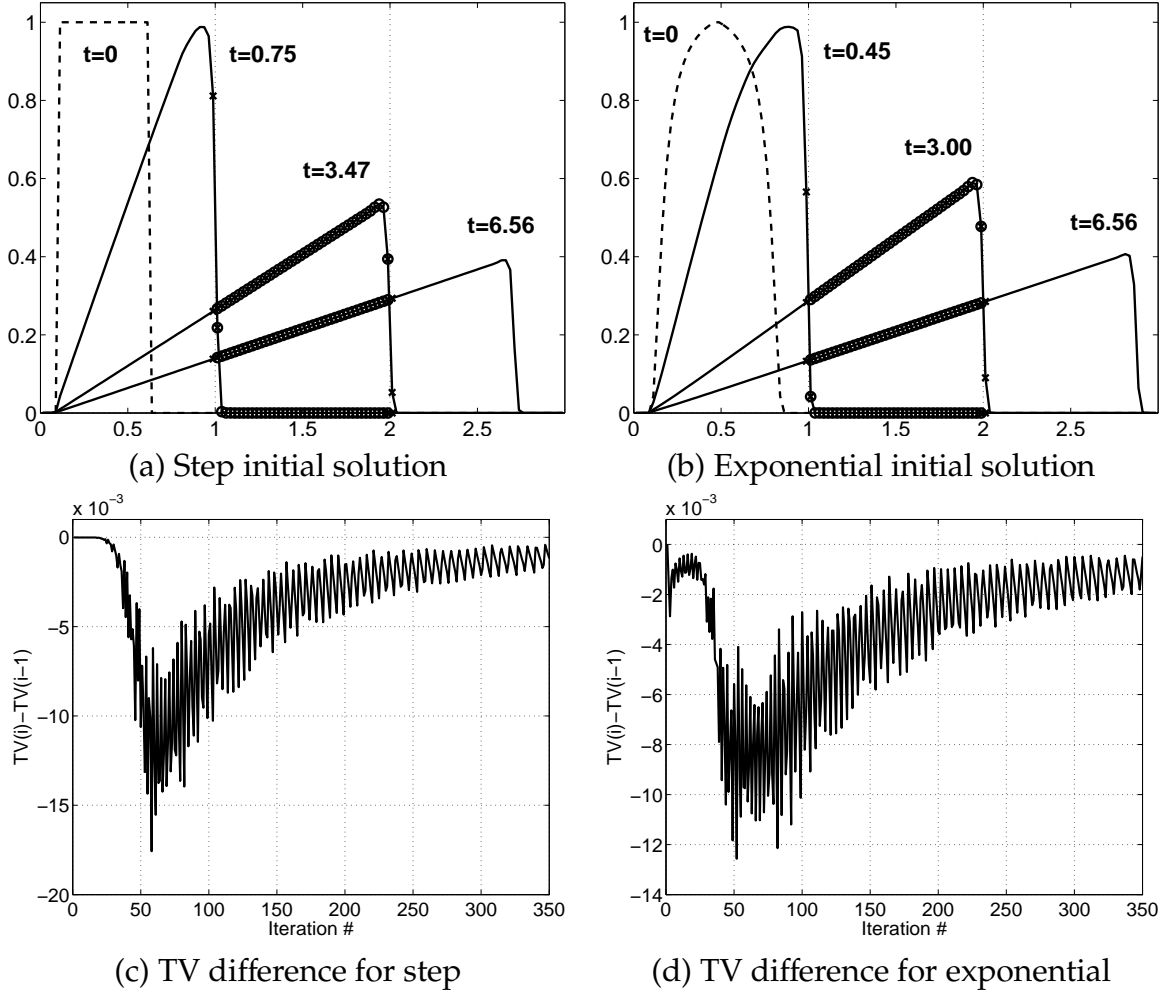


Figure 4: Burgers' equation with the initial profile dashed,  $dx = 0.025$ ,  $dt_{\text{fine}} = 0.019$  (CFL=0.75) and solved with MPRK2 for (a) the step profile and (b) exponential profile, for  $m = 2$  at different time locations. For each profile we show the TV variation of the solution: in (c) for the (a) setting, and in (d) for the (b) setting.

of the fastest wave for the entire spatial domain. The time integration scheme MPRK2 is second order and SSP, i.e. it preserves the TVD properties of the spatial discretization.

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

Figures 4.(a,b) show the Burgers' equation with two function profiles that pass through the fine ( $\Delta t/2$ ) region for different time positions. In both cases, as for the linear advection test case, we remark that the solution is not qualitatively affected as the wave passes through the interfaces. Figures 4.(c,d) show the TV difference (from the previous step), i.e.  $TV(y(t = t_i)) - TV(y(t = t_{i-1}))$ , for the solutions presented in Figure 4.(a,b). The difference is always negative, and thus the scheme is TVD.

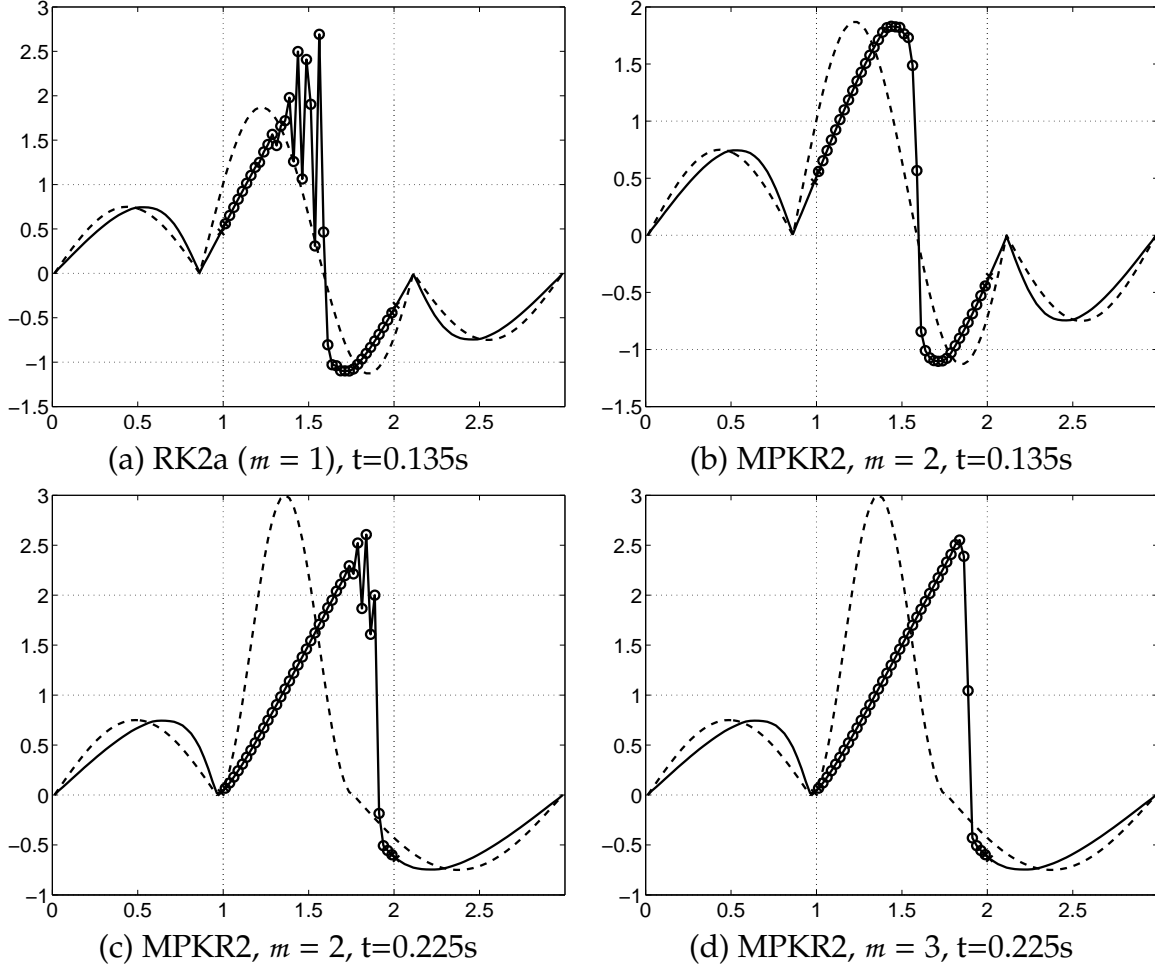


Figure 5: Burgers' equation with the initial profile dashed,  $dx = 0.025$ ,  $dt_{\text{coarse}} = 0.022$  (CFL=0.9) and (a)  $m = 1$ , using RK2a; (b)  $m = 2$ , solved with MPRK2. Second row shows the solution at  $t=0.225\text{s}$  solved with MPRK2 and same grid with (c)  $m = 2$ ; (d)  $m = 3$ . CFL condition is violated in (a) and (c) and are unstable. Figures (b) and (d) satisfy the CFL condition and are stable.

Results that use smaller local time-steps for Burgers' equation are presented in Figures 5. Here, we show a setting in which the CFL condition is violated for 5.(a,c) which become unstable. However, Figures 5.(b,d) locally satisfy the CFL condition and are stable. This approach uses MPRK2 with different  $m$  that can be adjusted dynamically according to the solution's characteristics in order to efficiently stabilize the scheme: The solution in Figure 5.(b) uses  $m = 2$  in the fast region and avoids the instabilities that occur in Figure 5.(a), while the solution in Figure 5.(d) uses  $m = 3$  and circumvents the oscillations present in fig. 5.(c).

The spatial discretization scheme is TVD and stable under a CFL-like condition. The time integration scheme, MPRK2, with  $m = 2, 3$  preserves the TVD of the spatial discretization scheme for our particular examples, as seen in Figures 4.(c,d). Although we do not give a formal proof for the MPRK2 TVD preserving, we consider that this empirical

evidence is strong enough to support further investigation and keep the TVD claim as a pro forma conjecture.

## 5 Conclusions and Future Work

We have studied a systematic way to expand SSP Runge-Kutta methods to SSP multirate partitioned RK schemes. In the context of high performance computing, this approach allows an efficient large scale solution computation on partitions that have different characteristics. Moreover, using the method of lines approach, the properties (positivity, TVD) of the spatial discretization are preserved by the MPRK routines.

This paper is aimed toward hyperbolic conservation laws. In this setting we show an efficient way to use MPRK2 in an adaptive mesh refinement framework, and to stabilize the CFL restricted schemes while preserving the stability properties of the original method. The interface treatment between subsequent domains is very important in this research. Dawson and Kirby [2001]; Kirby [2002] and Berkvens et al. [1999] present a multirate locally high order time discretization scheme. However, the slow (or coarse) flux is kept constant at the interface between the slow - fast partitions, and thus the overall method is reduced to first order of accuracy. We extend their approach to second order accurate methods by a very simple and general method that also satisfies a large set of stability criteria.

A very simple and intuitive construction algorithm from an SSP Runge-Kutta base scheme to an  $M$  partitioned multirate second order accurate is presented in this paper. We show that this method is positivity and maximum principle preserving, and provide strong evidence for TVD preserving.

The test problems showed in this paper demonstrate the applicability for this method. First, we present a linear example applied in an adaptive mesh refinement context. The adaptive fine mesh traces the large gradient part of the solution alleviating the diffusion errors. Here, we extend the original fixed grid method of Hundsdorfer et al. [1995] to a nonuniform mesh and maintain the time-step proportional to the grid size. Second, we consider a fixed grid nonlinear test case that adapts its time-step according to the CFL condition, and thus maintains the stability of the scheme.

We restrict our numerical results to one dimensional scalar test cases. It is straightforward to extend this method to vectorial cases and multidimensional domains via dimension splitting.

A rigorous proof for TVD and check for the entropy inequality will be addressed in future studies. We also plan to expand the proposed explicit SSP second order multirate Runge-Kutta method MPRK2 to higher orders of accuracy using the same systematic approach. Previous studies showed large accuracy gains when refining the spatial domain in large scale scientific applications, however the cost of reducing the time-step was typically very large. This multirate approach alleviates this restriction while preserving a high order of accuracy. In this context, we intend to apply this multirate approach to a large scale application in a future study.



Fast method	Slow method
$y_F^0$	$y_S^0$
$y_F^A = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^0, y_S^0\right)$	$y_S^A = \mathcal{E}_S\left(\Delta t, y_F^0, y_S^0\right)$
$y_F^{A*} = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^A, y_S^A\right)$	$y_S^{A*} = \mathcal{E}_S\left(\Delta t, y_F^A, y_S^A\right)$
$y_F^B = \frac{1}{2}(y_F^0 + y_F^{A*})$	
$y_F^C = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^B, y_S^0\right)$	$y_S^C = \mathcal{E}_S\left(\Delta t, y_F^B, y_S^0\right)$
$y_F^{C*} = \mathcal{E}_F\left(\frac{\Delta t}{2}, y_F^C, y_S^C\right)$	$y_S^{C*} = \mathcal{E}_S\left(\Delta t, y_F^C, y_S^C\right)$
$y_F^1 = \frac{1}{2}(y_F^B + y_F^{C*})$	$y_S^1 = \frac{1}{4}(2y_S^0 + y_S^{A*} + y_S^{C*})$

Table 5: MPRK2 Euler steps for the fast and slow methods for  $m = 2$ .

Base method	Fast method							Slow method								
	0	0						0	0							
			1/3	1/3	0			1	1	0						
0	0	0	1/3	1/6	1/6	0		0	0	0	0					
1	1	0	2/3	1/6	1/6	1/3	0	1	0	0	1	0				
	1/2	1/2	2/3	1/6	1/6	1/6	1/6	0	0	0	0	0	0	0		
			1	1/6	1/6	1/6	1/6	1/3	0	1	0	0	0	0	1	0
				1/6	1/6	1/6	1/6	1/6	1/6		1/6	1/6	1/6	1/6	1/6	1/6

Table 6: MPRK order 2 Butcher tableau for  $m = 3$ .

## APPENDIX

### A Summary of MPRK2 Euler Steps for $m = 2$

The Euler steps for the fast and the slow method are summarized in Table 5.

### B An Order Two MPRK with $m = 3$

Here, we present the MPRK2 for a factor of three ( $m = 3$ ) between the fast and slow partitions using the same method as for  $m = 2$  (see sec. 3.4). The Butcher tableau for this method is shown in in Table 6.

The method can be written as follows:

$$\begin{aligned}
K_F^1 &= f_F(y_F^0, y_S^0) & K_S^1 &= f_S(y_F^0, y_S^0) \\
y_F^A &= y_F^0 + \frac{\Delta t}{3} K_F^1 & y_S^A &= y_S^0 + \Delta t K_S^1 \\
K_F^2 &= f_F(y_F^A, y_S^A) & K_S^2 &= f_S(y_F^A, y_S^A) \\
y_F^B &= y_F^0 + \frac{\Delta t}{6} K_F^1 + \frac{\Delta t}{6} K_F^2 & y_S^B &= y_S^0
\end{aligned}$$

$$\begin{aligned}
K_F^3 &= f_F(y_F^B, y_S^0) & K_S^3 &= f_S(y_F^B, y_S^0) \\
y_F^C &= y_F^B + \frac{\Delta t}{3} K_F^3 & y_S^C &= y_S^0 + \Delta t K_S^3 \\
K_F^4 &= f_F(y_F^C, y_S^C) & K_S^4 &= f_S(y_F^C, y_S^C) \\
y_F^D &= y_F^B + \frac{\Delta t}{6} K_F^3 + \frac{\Delta t}{6} K_F^4 & y_S^D &= y_S^0 \\
K_F^5 &= f_F(y_F^D, y_S^0) & K_S^5 &= f_S(y_F^D, y_S^0) \\
y_F^E &= y_F^D + \frac{\Delta t}{3} K_F^5 & y_S^E &= y_S^0 + \Delta t K_S^5 \\
K_F^6 &= f_F(y_F^E, y_S^E) & K_S^6 &= f_S(y_F^E, y_S^E) \\
y_F^1 &= y_F^0 + \frac{\Delta t}{6} \sum_{j=1}^6 K_F^j & y_S^1 &= y_S^0 + \frac{\Delta t}{6} \sum_{j=1}^6 K_S^j
\end{aligned} \tag{43}$$

The above MPRK2 can be written in Euler steps in the following way:

$$\begin{aligned}
y_F^1 &= \frac{1}{2} \left( y_F^0 + y_F^0 + \frac{\Delta t}{3} K_F^1 + \frac{\Delta t}{3} K_F^2 + \frac{\Delta t}{3} K_F^3 + \frac{\Delta t}{3} K_F^4 + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & y_F^A &= \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^0, y_S^0 \right) \\
&= \frac{1}{2} \left( y_F^0 + y_F^A + \frac{\Delta t}{2} K_F^2 + \frac{\Delta t}{3} K_F^3 + \frac{\Delta t}{3} K_F^4 + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & y_F^{A*} &= \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^A, y_S^A \right) \\
&= \frac{1}{2} \left( y_F^0 + y_F^{A*} + \frac{\Delta t}{3} K_F^3 + \frac{\Delta t}{3} K_F^4 + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & 2y_F^B &= y_F^0 + y_F^{A*} \\
&= \frac{1}{2} \left( y_F^B + y_F^B + \frac{\Delta t}{3} K_F^3 + \frac{\Delta t}{3} K_F^4 + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & y_F^C &= \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^B, y_S^0 \right) \\
&= \frac{1}{2} \left( y_F^B + y_F^C + \frac{\Delta t}{3} K_F^4 + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & y_F^{C*} &= \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^C, y_S^C \right) \\
&= \frac{1}{2} \left( y_F^B + y_F^{C*} + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & 2y_F^D &= y_F^B + y_F^{C*}, \quad y_F^E = \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^D, y_S^0 \right) \\
&= \frac{1}{2} \left( y_F^D + y_F^D + \frac{\Delta t}{3} K_F^5 + \frac{\Delta t}{3} K_F^6 \right), & y_F^E &= \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^D, y_S^0 \right), \quad y_F^{E*} = \mathcal{E}_F \left( \frac{\Delta t}{3}, y_F^E, y_S^E \right) \\
&= \frac{1}{2} \left( y_F^D + y_F^{E*} \right), & &
\end{aligned} \tag{44}$$

and

$$\begin{aligned}
y_S^1 &= \frac{1}{6} \left( 6y_S^0 + \Delta t K_S^1 + \Delta t K_S^2 + hK_S^3 + \Delta t K_S^4 + \Delta t K_S^5 + \Delta t K_S^6 \right), & y_S^{\{A,C,E\}} &= \mathcal{E}_S \left( \Delta t, y_F^{\{0,B,D\}}, y_S^0 \right) \\
&= \frac{1}{6} \left( 3y_S^0 + y_S^A + \Delta t K_S^2 + y_S^C + hK_S^4 + y_S^E + \Delta t K_S^6 \right), & y_S^{\{A^*,C^*,E^*\}} &= \mathcal{E}_S \left( \Delta t, y_F^{\{A,C,E\}}, y_S^{\{A,C,E\}} \right) \\
&= \frac{1}{6} \left( 3y_S^0 + y_S^{A*} + y_S^{C*} + y_S^{E*} \right). & &
\end{aligned} \tag{45}$$

The Euler steps for the fast and the slow methods are summarized in Table 7.

Fast method	Slow method
$y_F^0$	$y_S^0$
$y_F^A = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^0, y_S^0\right)$	$y_S^A = \mathcal{E}_S(\Delta t, y_F^0, y_S^0)$
$y_F^{A*} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^A, y_S^A\right)$	$y_S^{A*} = \mathcal{E}_S(\Delta t, y_F^A, y_S^A)$
$y_F^B = \frac{1}{2}(y_F^0 + y_F^{A*})$	
$y_F^C = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^B, y_S^0\right)$	$y_S^C = \mathcal{E}_S(\Delta t, y_F^B, y_S^0)$
$y_F^{C*} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^C, y_S^C\right)$	$y_S^{C*} = \mathcal{E}_S(\Delta t, y_F^C, y_S^C)$
$y_F^D = \frac{1}{2}(y_F^B + y_F^{C*})$	
$y_F^E = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^D, y_S^0\right)$	$y_S^E = \mathcal{E}_S(\Delta t, y_F^D, y_S^0)$
$y_F^{E*} = \mathcal{E}_F\left(\frac{\Delta t}{3}, y_F^E, y_S^E\right)$	$y_S^{E*} = \mathcal{E}_S(\Delta t, y_F^E, y_S^E)$
$y_F^1 = \frac{1}{2}(y_F^D + y_F^{E*})$	$y_S^1 = \frac{1}{6}(3y_S^0 + y_S^{A*} + y_S^{C*} + y_S^{E*})$

Table 7: MPRK2 Euler steps for the fast and slow methods for  $m = 3$ .

## References

- J.F. Andrus. Numerical solution for ordinary differential equations separated into subsystems. *SIAM Journal of Numerical Analysis*, 16(4):605–611, 1979.
- J.F. Andrus. Stability of a multirate method for numerical integration of ODEs. *Journal of Computational and Applied Mathematics*, 25:3–14, 1993.
- A. Bartel and M. Günther. A multirate W-method for electrical networks in state-space formulation. *Journal of Computational Applied Mathematics*, 147(2):411–425, 2002. ISSN 0377-0427.
- P.J.F. Berkvens, M.A. Botchev, W.M. Lioen, and J.G. Verwer. A zooming technique for wind transport of air pollution. Technical report, Centrum voor Wiskunde en Informatica, 1999.
- J.P. Boris and D.L. Book. Flux-corrected transport I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 135(2):172–186, 1997.
- S. Chakravarthy and S. Osher. Numerical experiments with the Osher upwind scheme for the Euler equations. *AIAA Journal*, 21:241–248, 1983.
- S. Chakravarthy and S. Osher. Computing with high-resolution upwind schemes for hyperbolic equations. *Lectures in applied mathematics*, 22:57–86, 1985.
- C. Dawson and R. Kirby. High resolution schemes for conservation laws with locally varying time steps. *SIAM Journal on Scientific Computing*, 22(6):2256–2281, 2001.
- B. Engquist and R. Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Mathematics of Computation*, 74:1707–1742, 2005.
- C. Engstler and C. Lubich. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58(2):173–185, 1997. ISSN 0010-485X.
- C.W. Gear and D.R. Wells. Multirate linear multistep methods. *BIT*, 24:484–502, 1984.
- S.K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sborni*, 47:271–290, 1959.
- J.B. Goodman, R. LeVeque, and J. Randall. On the accuracy of stable schemes for 2D scalar conservation laws. *Mathematics of Computation*, 45:15–21, 1985.
- S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001.
- M. Günther and M. Hoschek. ROW methods adapted to electric circuit simulation packages. In *ICCAM '96: Proceedings of the seventh international congress on Computational and applied mathematics*, pages 159–170. Elsevier Science Publishers B. V., 1997.

- M. Günther, A. Kvaerno, and P. Rentrop. Multirate partitioned Runge-Kutta methods. *BIT*, 38(2):101–104, 1998.
- M. Günther and P. Rentrop. Multirate ROW-methods and latency of electric circuits. *Applied Numerical Mathematics*, 1993.
- E. Hairer. Order conditions for numerical methods for partitioned ordinary differential equations. *Numerische Mathematik*, 36(4):431 – 445, 1981.
- E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, chapter II.1. Springer, 1993a.
- E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, chapter II.15. Springer, 1993b.
- A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 135(2):260–278, 1997. ISSN 0021-9991.
- W. Hundsdorfer, B. Koren, and M. van Loon. A positive finite-difference advection scheme. *Journal of Computational Physics*, 117(1):35–46, 1995.
- W. Hundsdorfer, S.J. Ruuth, and R.J. Spiteri. Monotonicity-preserving linear multistep methods. *SIAM Journal on Numerical Analysis*, 41(2):605–623, 2003.
- T. Kato and T. Kataoka. Circuit analysis by a new multirate method. *Electrical Engineering in Japan*, 126(4):55–62, 1999.
- R. Kirby. On the convergence of high resolution methods with multiple time scales for hyperbolic conservation laws. *Mathematics of Computation*, 72(243):1239–1250, 2002.
- A. Kværnø. Stability of multirate Runge-Kutta schemes. *International Journal of Differential Equations and Applications*, 1(1):97–105, 2000.
- A. Kværnø and P. Rentrop. Low order multirate Runge-Kutta methods in electric circuit simulation, 1999.
- A. Logg. Multi-Adaptive Galerkin Methods for ODEs I. *SIAM Journal on Scientific Computing*, 24(6):1879–1902, 2003a.
- A. Logg. Multi-Adaptive Galerkin Methods for ODEs II: Implementation and Applications. *SIAM Journal on Scientific Computing*, 25(4):1119–1141, 2003b.
- A. Logg. Multi-Adaptive Galerkin Methods for ODEs III: Existence and Stability. *Chalmers Finite Element Center Preprint Series*, (2004-04), 2004.
- A. Logg. Multi-Adaptive Galerkin Methods for ODEs III: Apriory Estimates. *SIAM Journal of Numerical Analysis*, 43(6):2624–2646, 2006.
- S. Osher and S. Chakravarthy. High resolution schemes and the entropy condition. *SIAM Journal on Numerical Analysis*, 21(5):955–984, 1984.

- S. Osher and S. Chakravarthy. Very high order accurate TVD schemes. *Oscillation Theory, Computation, and Methods of Compensated Compactness, IMA Vol. Math. Appl.*, 2:229–274, 1986.
- J.R. Rice. Split Runge-Kutta methods for simultaneous equations. *Journal of Research of the National Institute of Standards and Technology*, 60(B), 1960.
- P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- J. Sand and K. Burrage. A jacobi waveform relaxation method for ODEs. *SIAM Journal on Scientific Computing*, 20(2):534–552, 1998. ISSN 1064-8275.
- C-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988. ISSN 0021-9991.
- C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes,II. *Journal of Computational Physics*, 83(1):32–78, 1989. ISSN 0021-9991.
- R.J. Spiteri and S.J. Ruuth. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM Journal on Numerical Analysis*, 40(2):469–491, 2002.
- P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 21(5):995–1011, 1984.
- H.-Z. Tang and G. Warnecke. A class of high resolution schemes for hyperbolic conservation laws and convect ion-diffusion equations with varying time and space grids. submitted, 2003.
- C.B. Vreugdenhil and B. Koren, editors. *Numerical Methods for Advection–Diffusion Problems*, volume 45. Vieweg, 1993.
- S.T. Zalesak. Fully multidimensional flux corrected transport algorithms for fluids. *Journal of Computational Physics*, 31:335–362, 1979.