MAIDESC
A review on multirate methods

E. Itam*, B. Koobus*

March 2015

*IMAG, Université de Montpellier, Montpellier, France
emmanuelle.itam@univ-montp2.fr, koobus@math.univ-montp2.fr

Abstract

A review on multirate methods is proposed. This starts from Rice's pioneering work on first order differential equations [1] to recent works dealing with hyperbolic conservation laws [16, 17, 18, 20].

Keywords: Multirate approach, differential equations, hyperlic conservation laws.

1 Introduction

For the solution of EDOs or EDPs, explicit integration schemes are still often used because of the accuracy they can provide and their simplicity of implementation. Nevertheless, these schemes can prove to be very expensive in some situations, for example stiff EDOs whose solution components exhibit different time scales, system of non-stiff EDOs characterized by different activity levels (fast/slow), or EDPs discretized on computational grids with very small elements. In order to overcome this efficiency problem, different strategies were developped, first in the field of EDOs, in order to propose an interesting alternative:

- Multi-method shemes: for systems of EDOs containing both non-stiff and stiff parts, an explicit scheme is used for the non-stiff subsystem and an implicit method for the stiff one [2, 3, 4].
- Multi-order schemes: for non-stiff system of ODEs, the same explicit method and step size are used, but the order of the method is selected according to the activity level (fast/slow) of the considered subsystem of EDOs [12].
- Multirate schemes: for stiff and non-stiff problems, the same explicit or implicit method with the same order is applied to all subsystems, but the step size is chosen according to the activity level. The first multirate time integration algorithm goes back to the work of Rice [1].

In this work we focus on the multirate approach. The application of such shemes was first limited to ODEs [1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] and restricted to a low number of industrial problems. In the last fifteen years, the developpment and application of such methods to the time integration of PDEs was also performed. In particular, a few works were conducted on the system of ODEs that arise after semidiscretization of hyperbolic conservation laws [16, 17, 18, 19, 20, 21], and rare applications were performed in Computational Fluid Dynamics (CFD) [20, 21] for which we are interested.

The remainder of this document is organized as follows.

In Section 2, some base integration methods to solve first order differential equations are recalled . A survey of some important works performed in the domain of multirate approaches is given in Section 3. It starts from Rice's pioneering work on first order differential equations [1] to recent works dealing with hyperlic conservation laws [16, 17, 18, 20].

2 Base integration methods to solve $\dot{y} = f(t, y)$

For the purpose of this survey, some base integration methods for the solution of ODE $\dot{y} = f(t, y)$ are recalled in this section. We focus on two large families of methods: (i) linear multistep methods and (ii) Runge Kutta methods. The Butcher representation of Runge Kutta methods is also given.

- (i) **Linear multistep methods** (including one-step methods as degenerate cases) which are written :

$$y_n = \sum_{i=1}^{K_1} \alpha_i y_{n-i} + h \sum_{i=0}^{K_2} \beta_i \dot{y}_{n-i}$$

  where y_n approximates $y(t_n)$, $h = t_n - t_{n-1}$ and $\dot{y}_j = f(t_j, y_j)$.
  The simplest examples of linear multistep methods are the Euler (forward Euler) method,

$$y_n = y_{n-1} + h\dot{y}_{n-1}$$

  and the backward Euler method,

$$y_n = y_{n-1} + h\dot{y}_n.$$

  Two classes of linear multistep methods are often used for the solution of EDOs:

  - **Backward Differentiation Formulas (BDF) methods** ($K_2 = 0, K_1 = q$):

$$y_n = \sum_{i=1}^{q} \alpha_i y_{n-i} + h\beta_0 \dot{y}_n$$

  - **Adams methods**:
    - **explicit** of order q ($K_1 = 1, \alpha_1 = 1, K_2 = q, \beta_0 = 0$):

$$y_n = y_{n-1} + h \sum_{i=1}^{q} \beta_i \dot{y}_{n-i}$$

    - **implicit** of order q ($K_1 = 1, \alpha_1 = 1, K_2 = q - 1$):

$$y_n = y_{n-1} + h \sum_{i=0}^{q-1} \beta_i \dot{y}_{n-i}$$

- (ii) **Runge Kutta (RK) methods** which concern :

- **r-stage explicit RK methods**:

$$y_n = y_{n-1} + \sum_{i=1}^{r} b_i k_i$$

with $\quad k_1 = hf(t_{n-1}, y_{n-1})\ \ ,\quad k_i = hf(t_{n-1} + c_i h, y_{n-1} + \sum_{j=1}^{i-1} a_{ij} k_j)$

and $c_i = \sum_{j=1}^{i-1} a_{ij} \quad (i = 2 \dots r)$

- **r-stage implicit RK methods**:

$$y_n = y_{n-1} + \sum_{i=1}^{r} b_i k_i$$

with $\quad k_i = hf(t_{n-1} + c_i h, y_{n-1} + \sum_{j=1}^{r} a_{ij} k_j) \quad (i = 1 \dots r)$

- **r-stage Rosenbrock and Rosenbrock-Wanner (ROW) methods**:

$$y_n = y_{n-1} + \sum_{i=1}^{r} b_i k_i$$

with

$$k_i = hf(t_{n-1}+c_i h, y_{n-1}+\sum_{j=1}^{i-1} a_{ij} k_j)+d_i h^2 \frac{\partial f}{\partial t}(t_{n-1}, y_{n-1})+h\frac{\partial f}{\partial y}(t_{n-1}, y_{n-1}) \sum_{j=1}^{i} d_{ij} k_j$$

where the coefficients d_{ij} are chosen to optimize order and stability properties.

Hereafter, we give the representation in Butcher tableau of explicit and implicit RK methods :

- **Representation in Butcher tableau of r-stage explicit RK methods**:

$y_{n+1} = y_n + h\sum_{i=1}^{r} b_i k_i$ with $\quad k_i = f(t_n + c_i h, y_n + h\sum_{j=1}^{i-1} a_{ij} k_j)$

$$\begin{array}{c|ccccc} c_1 = 0 & 0 & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_r & a_{r1} & a_{r2} & \cdots & a_{r,r-1} & \\ \hline & b_1 & b_2 & \cdots & b_{r-1} & b_r \end{array} \qquad \text{or shorter} \qquad \begin{array}{c|c} c & A \\ \hline & b^T \end{array} \qquad \text{or} \qquad [A, b, c]$$

- **Representation in Butcher tableau of r-stage implicit RK methods,**:

$y_{n+1} = y_n + h\sum_{i=1}^{r} b_i k_i$ with $\quad k_i = f(t_n + c_i h, y_n + h\sum_{j=1}^{r} a_{ij} k_j)$

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1r} \\ \vdots & \vdots & & \vdots \\ c_r & a_{r1} & \cdots & a_{rr} \\ \hline & b_1 & \cdots & b_r \end{array} \quad \text{or shorter} \quad \begin{array}{c|c} c & A \\ \hline & b^T \end{array} \quad \text{or} \quad [A, b, c]$$

3 A review on multirate schemes

3.1 The pioneering work of Rice, 1960 [1]

This work, which is the first one in the field of multirate methods, considers the solution of the following system of EDOs

$\dot{x} = F(t, x, y), \quad x(t_0) = x_0$

$\dot{y} = G(t, x, y), \quad y(t_0) = y_0$

where $x(t)$ represents the latent component and $y(t)$ the active one (meaning $y(t)$ varies much more rapidly than $x(t)$).

The proposed time discretization of this system is schematically given in Figure 1.

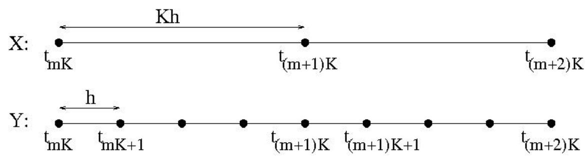

Figure 1: Time discretization of the latent and active component of the solution

In the work of Rice, the evaluation of the latent part of the solution $x_{(m+1)K}, x_{(m+2)K}, \ldots$ is performed with a time step Kh as follows:

$$x_{(m+1)K} = x_{mK} + \sum_{i=1}^{3} b_i k_i$$

with $k_i = hF(t_{mK} + c_i Kh, x_{mK} + \sum_{j=1}^{i-1} a_{ij} k_j, y_{mK} + \sum_{j=1}^{i-1} a_{ij} h_j) \quad (i = 1 \ldots 3)$

and $h_i = hG(t_{mK} + c_i Kh, x_{mK} + \sum_{j=1}^{i-1} a_{ij} k_j, y_{mK} + \sum_{j=1}^{i-1} a_{ij} h_j) \quad (i = 1 \ldots 2)$

Coefficients b_i, c_i, a_{ij} are given by any RK3 method (work also done with RK4).

The evaluation of the active part of the solution $y_{mK+j+1}, y_{mK+j+2}, \ldots$ is performed with a time step h as follows:

$$y_{mK+j+1} = y_{mK+j} + \sum_{i=1}^{3} \alpha_i d_i(j) \quad \text{for} \quad 0 \le j \le K-1.$$

with

$d_1(j) = hG(t_{mK+j}, \mathbf{x_{mK+j}}, y_{mK+j})$

$d_2(j) = hG(t_{mK+j} + \mu_2 h, \mathbf{x_{mK+j}} + \sum_{i=4}^{6} \lambda_i(j) k_{i-3}, y_{mK+j} + \gamma_{21} d_1(j))$

$d_3(j) = hG(t_{mK+j} + \mu_3 h, \mathbf{x_{mK+j}} + \sum_{i=7}^{9} \lambda_i(j) k_{i-6}, y_{mK+j} + \gamma_{31} d_1(j) + \gamma_{32} d_2(j))$

in which the following extrapolation, based on the previous k_i, is used :

$\mathbf{x_{mK+j}} = \mathbf{x_{mK}} + \sum_{\mathbf{i=1}}^{\mathbf{3}} \lambda_\mathbf{i}(\mathbf{j})\mathbf{k_i} \quad 1 \leq j \leq K-1,$

and where the several sets of parameters "$\alpha_i, \mu_i, \gamma_{ik}, \lambda_\mathbf{i}(\mathbf{j})$" are determined so that:
option 1: the local truncation error of integration formula for $y(t)$ is in $O(h^4)$
or
option 2: extrapolation parameters $\lambda_\mathbf{i}(\mathbf{j})$ lead to an extrapolation truncation error in $O(h^4)$ and the integration parameters are determined independantly.

Several numerical experiments were performed by applying the previous multirate approach on differential equations in order to investigate the integration errors as a function of K and h. A typical example is given by the 2 degrees of freedom problem:

$$\begin{cases} \dfrac{dx}{dt} = x/2, \quad x(0) = 1 \\[2ex] \dfrac{dy}{dt} = x\,\cos(25t), \quad y(0) = 1/1250.5 \end{cases}$$

for which the good features of the proposed multirate approach is shown.

With the objective to illustrate the saving in computation with the multirate option, the number of operations and function evaluations were given for several Runge-Kutta methods and their multirate counterparts. For example, for the third-order version (RK3), we obtain:

	Additions	Multiplications	F evaluations	G evaluations
RK3	22K	28K	3K	3K
Multirate RK3	14(K+1)	17(K+1)	3	3K+2

3.2 The work of Skelboe on multirate BDF methods, 1989 [7].

In this work, the following system of EDOs is considered:

$$\dot{y} = f(t, y, z), \quad y(t_0) = y_0, \quad \text{fast subsystem}$$

$$\dot{z} = g(t, y, z), \quad z(t_0) = y_0, \quad \text{slow subsystem}$$

In the proposed multirate strategy, the fast subsystem is integrated by a k-step BDF formula (BDF-k) with step length h (Figure 2):

$y_m = \sum_{i=1}^{k} \alpha_i y_{m-i} + h\beta_0 f(t_m, y_m, z_m)$

and the slow subsystem is integrated by the same BDF-k formula but with step length $H = qh$(Figure 2):

$z_n = \sum_{i=1}^{k} \alpha_i z_{n-qi} + qh\beta_0 f(t_n, y_n, z_n)$

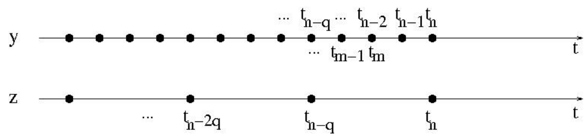

Figure 2: Time discretization of the fast and slow component of the solution

Different strategies for the sequence of computation are proposed:

- **Fastest first algorithm**
  step 1) Integration of the fast subsystem from t_{n-q} to t_n (q steps) with extrapolated values $\tilde{z}_m$ $(n-q < m \leq n)$ based on $z_{n-kq}, \ldots, z_{n-q}$ (following a Newton type formula, $\tilde{z}_m = \sum_{r=1}^{k} \tilde{\alpha}_{r,m-(n-q)} z_{n-rq}$).
  step 2) Integration of the slow subsystem from t_{n-q} to t_n (one step).
- **Slowest first algorithm**
  step 1) Integration of the slow subsystem from t_{n-q} to t_n (one step) with extrapolated value $\tilde{y}_n$ based on $y_{n-q-k+1}, \ldots, y_{n-q}$ (following a Newton type formula).
  step 2) Integration of the fast subsystem from t_{n-q} to t_n (q steps) with interpolated values $\tilde{z}_m$ $(n-q < m < n)$ based on $z_{n-(k-1)q}, \ldots, z_n$ (following a Newton type formula, $\tilde{z}_m = \sum_{r=0}^{k-1} \tilde{\alpha}_{r,m-(n-q)} z_{n-rq}$).

As for the application part, a 2×2 test problem is considered for investigating the stability properties of the previous multirate algorithms (BDF-1 and BDF-2, interpolation of order 0 and 1). From this application, it appears that the proposed multirate algorithms are not necessarily A-stable, limiting the use of such methods.

3.3 The work of Günther and Rentrop on multirate ROW methods, 1993 [10].

In this work, the following autonomous initial value problem is considered

$$\dot{y}(t) = f(y), \quad y(t_0) = y_0, \quad y \in \mathbb{R}^n$$

which can be split into active and latent components:

$$\dot{y_S} = f_S(y_S, y_L), \quad y_S(t_0) = y_{S0}, \quad y_S \in \mathbb{R}^{n_S}, \quad \text{active subsystem}$$

$$\dot{y_L} = f_L(y_S, y_L), \quad y_L(t_0) = y_{L0}, \quad y_L \in \mathbb{R}^{n_L}, \quad \text{latent subsystem.}$$

The proposed multirate strategy is the following:

- y_L is integrated with a ROW method on one large time step H :

  $y_L^H(t_0+H) = y_{L0} + \sum_{i=1}^{s} c_i k_i$

  $k_i = hf_L(\mathbf{\hat{y}_S(t_0+\alpha_i H)}, y_{L0} + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + HJ_L \sum_{j=1}^{i} \gamma_{ij} k_j \ , \quad J_L = \dfrac{\partial f_L}{\partial y_L}(y_{S0}, y_{L0})$

  where $\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}$ and $\mathbf{\hat{y}_S(t)}$ is an **extrapolated value for** $y_S(t)$.

- y_S is integrated with a ROW method and m time steps $h = H/m$:

  $y_S^H(t_0+(\lambda+1)h) = y_{S0}(t_0+\lambda h) + \sum_{i=1}^{s} c_i l_i$

  $l_i = hf_S(y_S(t_0+\lambda h) + \sum_{j=1}^{i-1} \alpha_{ij} l_j, \mathbf{\tilde{y}_L(t_0+\lambda h+\alpha_i)}) + hJ_S \sum_{j=1}^{i} \gamma_{ij} l_j \ ,$

  $J_S = \dfrac{\partial f_S}{\partial y_S}(y_S(t_0+\lambda h), \mathbf{\tilde{y}_L(t_0+\lambda h)}),$ for $\lambda = 0, 1, \ldots, m-1$

  where $\mathbf{\tilde{y}_L(t)}$ is an **extrapolated value for** $y_L(t)$.

The extrapolation formulas (Padé approximation of order (1,1)) are given by:

$$\hat{y}_{Si}(t_0+\bar{h}) = y_{Si}(t_0) + \frac{2\bar{h}\, f_{Si}(t_0)^2}{2f_{Si}(t_0) - \bar{h} \sum_{j=1}^{n_S} \frac{\partial f_{Si}}{\partial y_{Sj}}(y(t_0)) f_{Sj}(y(t_0)) - \bar{h} \sum_{j=n_S+1}^{n} \frac{\partial f_{Si}}{\partial y_{Lj}}(y(t_0)) f_{Lj}(y(t_0))}$$

$$\tilde{y}_{Li}(t_0+\bar{h}) = y_{Li}(t_0) + \frac{2\bar{h}\, f_{Li}(t_0)^2}{2f_{Li}(t_0) - \bar{h} \sum_{j=1}^{n_S} \frac{\partial f_{Li}}{\partial y_{Sj}}(y(t_0)) f_{Sj}(y(t_0)) - \bar{h} \sum_{j=n_S+1}^{n} \frac{\partial f_{Li}}{\partial y_{Lj}}(y(t_0)) f_{Lj}(y(t_0))}$$

The application part concerns the simulation of electric circuits (inverter chain) leading to the solution of stiff EDOs (system of 250-4000 differential equations). A multirate 4-steps ROW method was implemented, leading to a A-stable algorithm, and a speedup up to 2.8 compared to the classical RK4 method.

3.4 The work of Löhner-Morgan-Zienkiewicz on explicit multirate schemes for hyperbolic problems, 1984 [21].

To our knowledge, this work is the first one on multirate methods which deals with applications in CFD.
In this study, the problem of interest is given by

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0 \quad \text{in } \Omega = \Omega_1 \cup \Omega_2,$$

where Ω_1 and Ω_2 are two subregions with different grid resolution (see Figure 3 in 1D), and with, for a given explicit scheme, an allowable time step Δt_1 in Ω_1 and $\Delta t_2 = \Delta t_1/n$ in Ω_2.

One global time step of the proposed multirate explicit scheme (for 2 subdomains) is performed as follows:

- Add to Ω_2 two grid points of Ω_1, let Ω_2' be the new subdomain obtained (see Figure 3).

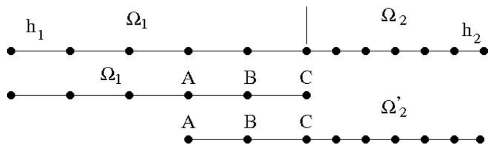

Figure 3: 1D model, 2 subdomains splitting

- Specify a boundary condition for U (free or fixed) at point C (see Figure 3) and advance one global time step Δt_1 in Ω_1.
- Specify a boundary condition for U (free or fixed) at point A (see Figure 3) and advance n small time steps $\Delta t_2 = \Delta t_1/n$ in Ω_2'.
- U_A is obtained from Ω_1, U_C from Ω_2', U_B = mean values obtained from Ω_1 and Ω_2', where B is the point between A and C (see Figure 3).

The same procedure can be performed with more than 2 subdomains splitting, and in the multidimensional case.

The above multirate scheme was implemented with a second order explicit finite element scheme (Taylor-Galerkin method of Donea).
Three test-cases were considered:

- A transient solution of a 1D shock tube problem (Sod).
- A transient solution of a 2D supersonic inviscid flow around a circular cylinder.
- A steady-state solution of a 2D supersonic inviscid flow past a wedge.

The first two test-cases were chosen in order to show that shocks can be handled by the method without problems. The third test-case illustrates the gain in CPU-time that can be obtained with the multirate approach. A speedup of 2 between the multirate scheme and its single-rate counterpart is achieved.

3.5 The work of Kirby on a multirate forward Euler scheme for hyperbolic conservation laws, 2002 [19]

This work deals with one-dimensional hyberbolic conservation laws

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

After semi-discretization by a finite volume MUSCL scheme, a system of EDOs is obtained

$$\dot{y}_i(t) = f_i(y_1(t), \dots, y_n(t)), \quad i = 1, \dots, n$$

that is partitioned in fast and slow subsystems

$$\dot{y}_F = f_F(y_F, y_S) \quad \text{(fast solution subsystem, explicit Euler time step } \Delta t/m)$$

$$\dot{y}_S = f_S(y_F, y_S) \quad \text{(slow solution subsystem, explicit Euler time step } \Delta t).$$

A multirate scheme based on forward Euler steps is proposed for the solution of these subsystems:

- For y_F : m steps integration from t^n to t^{n+1}
  $y_F^{n+\eta_k} = y_F^{n+\eta_{k-1}} + \sigma_k \Delta t\, f_F(y_F^{n+\eta_{k-1}}, y_S^n), \quad k = 1, \ldots, m-1$
  $y_F^{n+1} = y_F^{n+\eta_{m-1}} + \sigma_m \Delta t\, f_F(y_F^{n+\eta_{m-1}}, y_S^n)$

- For y_S : 1 step integration from t^n to t^{n+1}
  $y_S^{n+1} = y_S^n + \Delta t\, f_S(y_F^n, y_S^n)$

where $\sum_{k=1}^{m} \sigma_k = 1$ with $0 < \sigma_k \leq 1, \eta_l = \sum_{k=1}^{l} \sigma_k,\ \eta_0 = 0$ and $t^{n+\eta_k} = t^n + \eta_k \Delta t$.

It is shown that the proposed multirate scheme satisfies the TVD property and a maximum principle under local CFL conditions, but it is only first order time accurate. No application has been presented in this theoritical work.

3.6 The work of Constantinescu and Sandu on multirate RK methods for hyperbolic conservation laws, 2007 [16]

One-dimensional scalar hyperbolic equations are considered in this study:

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

The objective of this work is to develop a second-order accurate multirate scheme that inherits stability properties (maximum principle, TVD, TVB, monotocity-preservation, positivity) of the single rate integrator.
After a semi-discrete finite volume approximation (which satisfies some of the above stability properties), a system of EDOs is obtained

$$\dot{y}_i(t) = f_i(y_1(t), \ldots, y_n(t)), \quad i = 1, \ldots, n$$

which is partitioned into slow and fast subsystems

$$\dot{y}_F = f_F(y_F, y_S), \quad \text{fast subsystem}$$

$$\dot{y}_S = f_S(y_F, y_S), \quad \text{slow subsystem.}$$

This partioning can be represented schematically by Figure 4:

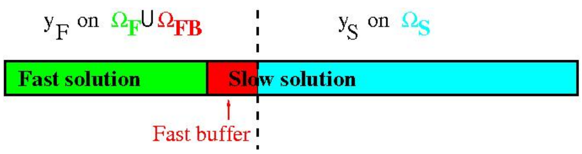

Figure 4: Partitioning into fast and slow subsytems

where

$\mathbf{\Omega_F}$ is the subdomain corresponding to a fast characteristic time and where a small time step $\Delta t/m$ is used in the multirate scheme (m = integer corresponding to the number of small time steps per large time step Δt),

$\mathbf{\Omega_S}$ denotes the subdomain with a slow characteristic time and where a large time step Δt is used in the multirate scheme,
$\mathbf{\Omega_{FB}}$ (fast buffer)is the subdomain with a slow characteristic time but a small time step $\Delta t/m$ is used in the multirate scheme (the size of the fast buffer is equal to half of the stencil size),
the **Fast solution** is the part of the solution which corresponds to a fast characteristic time,
the **Slow solution** denotes the part of the solution which corresponds to a slow characteristic time,
the **Fast buffer solution** is the solution defined on the fast buffer Ω_{FB},
y_F denotes the **fast solution** $\cup$ **fast buffer solution** for which the time integration is based on the small time step $\Delta t/m$,
y_S is the **slow solution** $\setminus$ **fast buffer solution** for which the time integration uses the large time step Δt.

Note that y_F is different from the fast solution, and that y_S is different from the slow solution. The fast buffer $\mathbf{\Omega_{FB}}$, which bridges the transition between Ω_F and Ω_S, is introduced for the purpose that the multirate scheme satisfies the stabilitiy properties of the single rate scheme.

The following general multirate partitioned RK scheme, associated with a base RK method noted RK_B, is proposed (the Butcher notation is used, see Section 2):

Hereafter, RK_F denotes the RK method which applies on y_F defined on $\Omega_F \cup \Omega_{FB}$, and RK_S the RK method which applies on y_S defined on $\Omega_S \backslash \Omega_{FB}$ (see Figure 5).

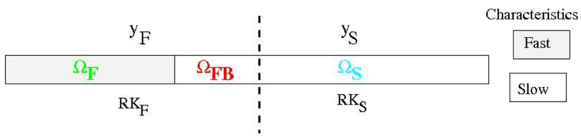

Figure 5: RK_F applies on y_F, and RK_S applies on y_S

Base method (RK_B) :

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

Fast method (RK_F) : $\dot{y}_F = f_F(y_F, y_S)$ **Slow method** (RK_S) : $\dot{y}_S = f_S(y_F, y_S)$

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

One can notice that **the same weight coefficients are taken for** RK_F **and** RK_S $\left(b_{F_i} = b_{S_i} = \frac{b_i}{m}\right)$, for the purpose of second order accuracy and conservation properties of the multirate scheme.

For the case $RK_B = RK2$ and $m = 2$ (2 small time steps $\Delta t/2$ per large time step Δt), the previous multirate scheme based on RK_B, RK_F and RK_F becomes:

$$
\begin{array}{c|cc}
0 & 0 & 0 \\
1 & 1 & 0 \\
\hline
 & 1/2 & 1/2
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
1/2 & 1/2 & 0 & & \\
1/2 & 1/4 & 1/4 & 0 & \\
1 & 1/4 & 1/4 & 1/2 & 0 \\
\hline
 & 1/4 & 1/4 & 1/4 & 1/4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
1 & 1 & 0 & & \\
0 & 0 & 0 & 0 & \\
1 & 0 & 0 & 1 & 0 \\
\hline
 & 1/4 & 1/4 & 1/4 & 1/4
\end{array}
$$

Base method ($\mathbf{RK_B}$) Fast method ($\mathbf{RK_F}$) Slow method ($\mathbf{RK_S}$)

which corresponds to the following $\mathbf{RK_B}$, $\mathbf{RK_F}$ and $\mathbf{RK_S}$ stages :

$\mathbf{RK_B}$ $(\dot{y} = f(y))$:

$$
\begin{aligned}
&k^1 = f(y^n) \\
&y^{(1)} = y^n + \Delta t\, k^1 \\
&k^2 = f(y^{(1)}) \\
&y^{n+1} = y^n + \tfrac{\Delta t}{2}(k^1 + k^2)
\end{aligned}
$$

$\mathbf{RK_F}$ $(\dot{y}_F = f_F(y_F, y_S))$:

$$
\begin{aligned}
&k_F^1 = f_F(\mathbf{y_F^n}, \mathbf{y_S^n}) \\
&y_F^{(1)} = y_F^n + \tfrac{\Delta t}{2} k_F^1 \\
&k_F^2 = f_F(\mathbf{y_F^{(1)}}, \mathbf{y_S^{(1)}}) \\
&y_F^{(2)} = y_F^n + \tfrac{\Delta t}{4} k_F^1 + \tfrac{\Delta t}{4} k_F^2 \\
&k_F^3 = f_F(\mathbf{y_F^{(2)}}, \mathbf{y_S^n}) \\
&y_F^{(3)} = y_F^{(2)} + \tfrac{\Delta t}{2} k_F^3 \\
&k_F^4 = f_F(\mathbf{y_F^{(3)}}, \mathbf{y_S^{(3)}}) \\
&y_F^{n+1} = y_F^n + \tfrac{\Delta t}{4}(k_F^1 + k_F^2 + k_F^3 + k_F^4)
\end{aligned}
$$

$\mathbf{RK_S}$ $(\dot{y}_S = f_S(y_F, y_S))$:

$$
\begin{aligned}
&k_S^1 = f_S(\mathbf{y_F^n}, \mathbf{y_S^n}) \\
&y_S^{(1)} = y_S^n + \Delta t\, k_S^1 \\
&k_S^2 = f_S(\mathbf{y_F^{(1)}}, \mathbf{y_S^{(1)}}) \\
&y_S^{(2)} = y_S^n \\
&k_S^3 = f_S(\mathbf{y_F^{(2)}}, \mathbf{y_S^n}) \\
&y_S^{(3)} = y_S^n + \Delta t\, k_S^3 \\
&k_S^4 = f_S(\mathbf{y_F^{(3)}}, \mathbf{y_S^{(3)}}) \\
&y_S^{n+1} = y_S^n + \tfrac{\Delta t}{4}(k_S^1 + k_S^2 + k_S^3 + k_S^4)
\end{aligned}
$$

Note that at each stage of the multirate formula, the flux functions are evaluated at the same argument values so that the conservation properties are satisfied by the multirate scheme.

In order to decouple y_S from y_F and to show that the major part of y_S is advanced in time using the large time step Δt, a slow buffer region Ω_{SB} of size $m\times$ half of stencil size is introduced (see Figure 6) so that f_S depends only on y_S in $\Omega_S \backslash \Omega_{SB}$ (we recall that m denotes the number of small time steps per large time step Δt) .
The previous multirate scheme (case $RK_B = RK2$ and $m = 2$) becomes

$\mathbf{\Omega_F \cup \Omega_{FB}}$, $\mathbf{RK_F}$:
$(\dot{y}_F = f_F(y_F, y_S))$

$$
\begin{aligned}
&k_F^1 = f_F(\mathbf{y_F^n}, \mathbf{y_S^n}) \\
&y_F^{(1)} = y_F^n + \tfrac{\Delta t}{2} k_F^1 \\
&k_F^2 = f_F(\mathbf{y_F^{(1)}}, \mathbf{y_S^{(1)}}) \\
&y_F^{(2)} = y_F^n + \tfrac{\Delta t}{4} k_F^1 + \tfrac{\Delta t}{4} k_F^2 \\
&k_F^3 = f_F(\mathbf{y_F^{(2)}}, \mathbf{y_S^n}) \\
&y_F^{(3)} = y_F^{(2)} + \tfrac{\Delta t}{2} k_F^3 \\
&k_F^4 = f_F(\mathbf{y_F^{(3)}}, \mathbf{y_S^{(3)}}) \\
&y_F^{n+1} = y_F^n + \tfrac{\Delta t}{4}(k_F^1 + k_F^2 + k_F^3 + k_F^4)
\end{aligned}
$$

$\mathbf{\Omega_{SB}}$, $\mathbf{RK_S}$:
$(\dot{y}_S = f_S(y_F, y_S))$

$$
\begin{aligned}
&k_S^1 = f_S(\mathbf{y_F^n}, \mathbf{y_S^n}) \\
&y_S^{(1)} = y_S^n + \Delta t\, k_S^1 \\
&k_S^2 = f_S(\mathbf{y_F^{(1)}}, \mathbf{y_S^{(1)}}) \\
&y_S^{(2)} = y_S^n \\
&k_S^3 = f_S(\mathbf{y_F^{(2)}}, \mathbf{y_S^n}) \\
&y_S^{(3)} = y_S^n + \Delta t\, k_S^3 \\
&k_S^4 = f_S(\mathbf{y_F^{(3)}}, \mathbf{y_S^{(3)}}) \\
&y_S^{n+1} = y_S^n + \tfrac{\Delta t}{4}(k_S^1 + k_S^2 + k_S^3 + k_S^4)
\end{aligned}
$$

$\mathbf{\Omega_S \backslash \Omega_{SB}}$, $\mathbf{RK_S \to RK_B}$:
$(\dot{y}_S = f_S(y_S))$

$$
\begin{aligned}
&k_S^1 = f_S(\mathbf{y_S^n}) \\
&y_S^{(1)} = y_S^n + \Delta t\, k_S^1 \\
&k_S^2 = f_S(\mathbf{y_S^{(1)}}) \\
&\left\{ y_S^{(2)} = y_S^n \right\} \\
&\left\{ k_S^3 = f_S(\mathbf{y_S^n}) = \mathbf{k_S^1} \right\} \\
&\left\{ y_S^{(3)} = y_S^n + \Delta t\, k_S^3 = y_S^{(1)} \right\} \\
&\left\{ k_S^4 = f_S(\mathbf{y_S^{(3)}}) = \mathbf{k_S^2} \right\} \\
&y_S^{n+1} = y_S^n + \tfrac{\Delta t}{2}(k_S^1 + k_S^2)
\end{aligned}
$$

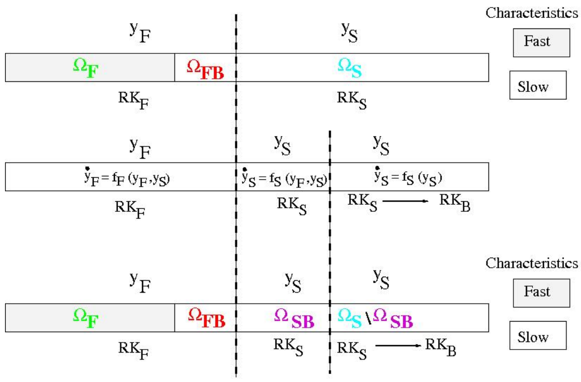

Figure 6: Slow buffer Ω_{SB}

Expressions in braces (last column) indicate that they are not evaluated.

The proposed multirate partitioned RK scheme satisfies the following properties:

- second order accurate
- conservative
- nonlinear stable (positivity, maximum principle preserving, TVB)

The theoritical speedup (single rate/multirate) is given by:

$$\text{Speedup} = \frac{m(N_{\Omega_F}+N_{\Omega_{FB}}+N_{\Omega_S})}{m(N_{\Omega_F}+N_{\Omega_{FB}}+N_{\Omega_{SB}})+N_{\Omega_S}-N_{\Omega_{SB}}} = \frac{m(N_{\Omega_F}+N_{\Omega_{FB}}+N_{\Omega_S})}{m(N_{\Omega_F}+N_{\Omega_{FB}}+N_{int}m\Delta)+N_{\Omega_S}-N_{int}m\Delta}$$

where N_{Ω_X} denotes the number of nodes in Ω_X, Δ is the half of stencil size, and N_{int} the number of interface nodes between $N_{\Omega_{FB}}$ and N_{Ω_S}.

It is deduced that:

- the speedup depreciates as m grows, but in practical applications large m can be avoided through nested partitioning.
- in practice, $N_{\Omega_{SB}} << min(N_{\Omega_F} + N_{\Omega_{FB}}, N_{\Omega_S})$
  $\Rightarrow$ Speedup $\simeq \frac{m(N_{\Omega_F}+N_{\Omega_{FB}}+N_{\Omega_S})}{m(N_{\Omega_F}+N_{\Omega_{FB}})+N_{\Omega_S}}$
  $\Rightarrow$ Speedup close to the ideal value of m if $N_{\Omega_F} + N_{\Omega_{FB}} << N_{\Omega_S}$.

The applications presented in this work concern the following points:

- a multirate RK2 scheme with $m = 2$ ($\Delta t/2$) and $m = 3$ ($\Delta t/3$), and 2 levels of partitioning, is used,
- the problem of a 1D advection equation (initial solutions : step, triangular and exponential shapes), on fixed and moving grids, discretized with a 2nd order limited FV scheme, is considered,

- the case of a 1D burger equation (initial solutions : step and exponential shapes), on fixed grids, discretized with a 3rd order TVD FV scheme, is also presented.

It was checked that the numerical solutions are second order accurate, positive, obey the maximum principle, TVD, wiggle free, and the integration is conservative.

By way of example, for the case of the Burger equation (fast region $\simeq 10$ % entire domain), the speedup (single rate/multirate) is given by the following Table :

Time ratio	Single rate time (sec)	Multirate time (sec)	Experimental Speedup	Theoritical Speedup
$m = 2$	25.28	13.71	1.84	1.80
$m = 3$	36.73	15.07	2.43	2.45

3.7 The work of Seny *et al.* on a parallel implementation of multirate RK methods, 2014 [20]

The work of Seny *et al.* focuses on the efficient parallel implementation of explicit multirate RK schemes in the framework of discontinuous Galerkin methods. The multirate RK scheme used is the approach proposed by Constantinescu [16] and introduced in the previous subsection.
In order to optimize the parallel efficiency of the multirate scheme, they propose a solution based on multi-constraint mesh partitioning. The objective is to ensure that the workload, for each stage of the multirate algorithm, is almost equally shared by each computer core i.e. the same number of elements are active on each core, while minimizing inter-processor communications. The METIS software is used for the mesh decomposition, and the parallel programming is performed with the Message Passing Interface.
The efficiency of the parallel multirate strategy is evaluated on three test cases: the wind driven circulation in a square basin and the propagation of a tsunami wave using a shallow water model (two-dimensional), and the acoustic propagation in a turbofan engine intake using the linearized Euler equations (three-dimensional). It is shown that the multi-constraint partitioning strategy increases the efficiency of the parallel multirate scheme compared to the classical single-constraint partitioning. However, they observe that strong scalability is achieved with more difficulty with the multirate algorithm than with its singlerate counterpart, especially when the number of processors becomes important compared to the number of mesh elements. The possible low number of elements per multirate group and per processor is a limiting factor for the proposed approach.

4 Conclusion

A review on multirate schemes is performed in this document. Some important works have been reviewed, ranging from that of Rice [1] in 1960 on first order differential equations to recent works on time integration of PDEs associated with hyperbolic conservation laws [16, 20]. On the basis of this review, it can be said that few works on multirate methods were conducted for the solution of hyperbolic conservation laws and rare applications in CFD were performed [21, 20]. There is therefore a need to developp such methods in the field of CFD.

References

[1] J. R. Rice, Split Runge-Kutta Method for Simultaneous Equations, *Journal of Research of the National Bureau of Standards - B. Mathematics and Mathematical Physics*, Vol. 64B, No3, 1960.

[2] E. Hofer, A partially implicit method for large stiff systems of ODEs with only few equations introducing small time-constants, *SIAM J. Numerical Analysis*, pp. 645-666, 1976.

[3] P. Rentrop, Partitioned Runge-Kutta methods with stiffness detection and step-size control, *Numer. Mathematik*, 47, pp. 545-564, 1985.

[4] R. Weiner, M. Arnold, P. rentrop, K. Strehmel, Partioning strategies in Runge-Kutta type methods, *IMA J. of Numer. Analysis*, 13, pp. 303-319, 1993.

[5] J.F. Andrus, Numerical solution of systems of ordinary differential equations separated into subsystems, *SIAM J. Numerical Analysis*, Vol.16, No 4, 1979.

[6] C.W. Gear and D.R. Wells, Multirate linear multistep methods, *BIT*, 484-502, 1984.

[7] S. Skelboe, Stability properties of backward differentiation multirate formulas, *Applied Numerical Mathematics*, 5, 151-160, 1989.

[8] J. Sand and S. Skelboe, Stability of backward Euler multirate methods and convergence of waveform relaxation, *BIT*, 32, 350-366, 1992.

[9] J.F. Andrus, Stability of a multi-rate method for numerical integration of ODE'S, *Computers Math. Applic.*, Vol. 25, No 2, pp. 3-14, 1993.

[10] M. Günther and P. Rentrop, Multirate ROW methods and latency of electric circuits, *Applied Numarical Mathematics*, 13, 83-102, 1993.

[11] C. Engstler and C. Lubich, MUR8: a multirate extension of the eight-order Dormer-Prince method, *Applied Numerical Mathematics*, 25, 185-192, 1997.

[12] C. Engstler and C. Lubich, Multirate Extrapolation Methods for Differential Equations with Different Time Scales, *Computing*, 58, 173-185, 1997.

[13] M. Günther, A. Kvaerno and P. Rentrop, Multirate partitioned Runge-Kutta methods, *BIT*, Vol. 38, No 2, pp. 101-104, 1998.

[14] M. Günther, A. Kvaerno and P. Rentrop, Multirate partitioned Runge-Kutta methods, *BIT*, Vol. 41, No 3, pp. 504-514, 2001.

[15] V. Savcenco, W. Hundsdorfer and J.G. Verwer, A multirate time stepping strategy for stiff ordinary differential equations, *BIT*, Vol. 47, pp. 137-155, 2007.

[16] E. Constantinescu and A. Sandu, Multirate Timestepping Methods for Hyperbolic Conservation Laws, *J. sci. Comp*, 33(3), 239-278, 2007.

[17] A. Sandu and E. Constantinescu, Multirate Explicit Adams Methods for Time Integration of Conservation Laws, *J. Sci. Comp*, 38: 229-249, 2009.

[18] P.R. Mugg, Construction and analysis of multi-rat partitioned Runge-Kutta methods, *Thesis*, Naval Postgraduate School, Monterey, California, June 2012.

[19] R. Kirby, On the convergence of high resolutin methods with multiple time scales for hyperbolic laws, *Mathematics of Computation*, Vol. 72, Number 243, 129-1250, 2002.

[20] B. Seny, J. Lambrechts, T. Toulorge, V. Legat, J.-F. Remacle, An efficient parallel implementation of explicit multirate Runge-Kutta schemes for discontinuous Galerkin computations, *JCP*, Vol. 256, pp. 135-160, 2014.

[21] R. Löhner, K. Morgan and O.C. Zienkiewicz, The use of doamin splitting with an explicit hyperbolic solver, *Comput. Methods Appl. Mech. Engrg.*, 45, 313-329, 1984.