

On some explicit local time stepping finite volume schemes for CFD

G. Jeanmasson^{*,**}, I. Mary^{*} and L. Mieussens^{**}

Corresponding author e-mail address: guillaume.jeanmasson@onera.fr

Corresponding author postal adress:

Onera Châtillon,
BP72 - 29 avenue de la Division Leclerc,
F-92322 CHATILLON CEDEX.

* ONERA, France.

** Université de Bordeaux, France.

Abstract: In this paper, two local time stepping schemes of order two and three in time are proposed. By construction, they are not mass conservative but a correction stage is added to make them conservative. These schemes are compared with some local time stepping schemes existing in the literature (schemes of Constantinescu and Sandu). The comparisons are carried out on various test cases. They prove that our schemes are efficient and our third order local time stepping has a higher time accuracy than the schemes based on the strategy of Constantinescu and Sandu. Our third order local time stepping scheme is used to perform an industrial-like test case: a 3D Large-Eddy Simulation over an airfoil.

Keywords: Explicit local time stepping schemes, Runge-Kutta methods, LES, turbulent flows.

Introduction

Numerical simulations are widely used to study fluid mechanics problems. Simulations are based on the discretized form of fluid mechanics equations (mainly Euler or Navier-Stokes equations). This discretized form is obtained by applying space and time integration methods. Finite volume approach is one of the most popular space integration method in CFD. For time integration, explicit Runge-Kutta methods have proven their efficiency for solving fluid mechanics problems. However, with these methods, the time step is strongly restricted by the numerical stability condition (CFL condition). This stability condition is defined in each cell. For each cell, it imposes that the time step is lower than the ratio between the grid size and the sum of the sound speed and the fluid velocity in the cell. The time step for the entire domain is chosen to be lower than the smallest

ratio in the mesh (generally imposed by the smallest cell in the mesh in our applications, since the fluid velocity is rather uniform). Consequently, the time step may be much smaller than necessary in the other cells, which is not efficient. It is particularly true in most CFD computations involving walls, which present some zones of mesh refinements where the stability condition is much more restrictive than in the other areas.

Implicit, unconditionally stable timestepping methods allow to use larger time steps, which reduces the computational time as compared to explicit methods. Note that the time step can not be too large in order to keep a good accuracy of the solution. For a Large Eddy Simulation of turbulent flows, the ratio between the time step and each cell characteristic length can not exceed 2-3 [1]. Moreover, implicit methods seem to have two drawbacks. First, an efficient parallel implementation of these methods may be difficult, especially if a strong scalability is required. In addition, implicit methods can be less accurate than explicit methods due to a residual numerical noise.

The use of an explicit local time stepping scheme seems to be a good approach to reduce the computational cost of explicit methods keeping their advantages with respect to implicit methods (solution accuracy and good compatibility with parallel programming). Indeed, an explicit local time stepping scheme allows a time step adaptation on the mesh to satisfy local stability conditions. Small time steps are used on cells which present the most severe CFL and larger time steps are applied on cells where CFL is less restrictive. This should result in a reduction of the computational cost with respect to explicit methods using a uniform time step.

Two approaches are possible to construct explicit local time stepping schemes. In the first approach, each cell of the computational domain is integrated with its own maximal allowable time step [2, 3, 4]. In the second approach, each time step of the computational domain is defined as a fraction of the largest time step and each time step is associated to a group of contiguous cells [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. In the literature, these schemes are frequently called "multirate schemes". All the local time stepping schemes presented in this article are constructed by following this second approach. Indeed, this approach seems well suited to High Performance Computations (HPC) since the memory accesses are much more efficient due to the regular predictable pattern of the cells groups. However, local time stepping schemes seem to have an important drawback: it is difficult to guarantee both high order time accuracy and mass conservation. In the 1980's, Osher and Sanders [6] proposed a locally varying time stepping scheme based on the forward Euler method. It is conservative but only first order accurate in time. Dawson and Kirby [13] extended the work of Osher and Sanders by performing a conservative local time stepping scheme based on a second order Runge-Kutta scheme. Tang and Warnecke [7] proved that the schemes in [6, 13] have a lack of consistency and proposed a new second order time accurate local time stepping scheme. However, their scheme is not mass-conservative. The local time stepping schemes presented in [17, 9, 10] are also second order time accurate but not mass-conservative. In [8], a comparison is made between different multirate Runge-Kutta schemes and a multirate time stepping scheme of order three in time is proposed. This third order local time stepping scheme borrows some ideas of implicit-explicit (IMEX) schemes [18]. However, it is not mass-conservative. Another third order non conservative explicit local time stepping scheme based on the theory of partitioned Runge-Kutta methods (PRK) is proposed in [14]. Recently, multirate Runge-Kutta schemes up to order four are developed by several authors [15, 16]. They are derived from the Multirate Infinitesimal Step method, introduced in [19]. These methods are very interesting since they can deal with several time step rates ("telescopic approach"), unlike [19]. The authors also provide rigorous stability analyses and propose numerical applications to several ordinary differential equations that confirm their theoretical findings. However, the authors don't perform numerical test on partial differential equations (PDEs), and their method seems to be not mass-conservative when applied to PDEs.

The strategy proposed by Constantinescu and Sandu [5] to construct local time stepping schemes seems promising. Indeed, their strategy allows to obtain second order accurate and mass conservative local time stepping schemes. Their method has been successfully applied to solve CFD equations [11]. Moreover, the overall stability of their original local time stepping scheme proposed in [5] can be improved using a higher order Runge-Kutta base method [8, 11]. Nevertheless, the overall method remains second order accurate in time [8, 11].

In this article, we propose two local time stepping schemes that are second order and third order time accurate. Consistent solutions are calculated at the interface between domains of different time steps like in [7, 8, 12, 9, 10] to obtain the desired orders of accuracy. Moreover, a correction stage is added to make these schemes mass-conservative. The first goal of this article is to make a comparison between our strategy to construct mass-conservative local time stepping schemes and the strategy of Constantinescu and Sandu. Moreover, in most of articles devoted to local time stepping methods, the schemes are validated on rather academic test cases (Burgers equation, advection equation...) [13, 7, 9, 10, 5, 8, 17]. The second goal of this article is then to show the ability of our third order local time stepping scheme to compute efficiently an industrial-like case: a 3D Large-Eddy Simulation (LES) over an airfoil.

In section 1, we present some theoretical concepts for the construction of local time stepping schemes and we introduce the strategy used by Constantinescu and Sandu. In section 2, we introduce our approach to construct mass-conservative local time stepping schemes of order two and three. In section 3, the comparison between the two strategies is performed with several academic test cases. The results of the LES over an airfoil computed with our third order local time stepping scheme are also presented in this final part.

1 Local time stepping scheme and Partitioned Runge-Kutta methods

1.1 Semi-discretization of a one-dimensional hyperbolic equation

We consider the one-dimensional hyperbolic equation :

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

with $y(0, x) = y^0(x)$, $x \in]-\infty, +\infty[$ and $t > 0$. We also consider a 1D computational domain divided into cells of variable length Δx_i . Cell i is bounded by points $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$. The finite volume method is applied to (1). After space integration over cell i , we obtain :

$$\frac{\partial y_i}{\partial t} = -\frac{1}{\Delta x_i}(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}), \quad (2)$$

where $y_i = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} y(t, x) dx$. The numerical flux at point $x_{i+\frac{1}{2}}$ is denoted by $F_{i+\frac{1}{2}}$, and depends on several values around y_i . For instance, we have: $F_{i+\frac{1}{2}}(y_{i-1}, y_i, y_{i+1}, y_{i+2})$ in the case of a flux function with a four cell stencil. For the sake of clarity for the reader, $F_{i+\frac{1}{2}}(y_{i-1}, y_i, y_{i+1}, y_{i+2})$ is denoted by $F_{i+\frac{1}{2}}$. For the sake of convenience, we introduce D_i , the flux balance on cell i :

$$D_i = \frac{1}{\Delta x_i}(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}). \quad (3)$$

The flux balance D_i is obviously a function of y_i , and several values around it ($y_{i-2}, y_{i-1}, y_{i+1} \dots$). The arguments of D_i are omitted for a better clarity. With this notation, (2) now reads:

$$\frac{\partial y_i}{\partial t} = -D_i. \quad (4)$$

Now (2) and (4) can be viewed as ordinary differential systems that can be solved with an explicit Runge-Kutta (RK) method, as described in the following section. This is the so called method of lines (MOL).

1.2 Explicit Runge-Kutta methods

We consider a non-autonomous ordinary differential equation:

$$\frac{dy}{dt} = f(t, y) \quad y(t=0) = y^0 \quad (5)$$

The solution of (5) at time t^n is y^n . A s -step explicit Runge-Kutta method allows to compute y^{n+1} (solution of (5) at time $t^{n+1} = t^n + \Delta t$) by using y^n and $s - 1$ intermediate values. In [20], a s -step explicit Runge-Kutta method is defined by the formulas :

$$t^{n,i} = t^n + c_i \Delta t, \quad (6)$$

$$y^{n,i} = y^n + \Delta t \sum_{j=1}^{i-1} a_{ij} K^{n,j}, \quad (7)$$

$$K^{n,i} = f(t^{n,i}, y^{n,i}), \quad (8)$$

where $1 \leq i \leq s$, and $K^{n,1} = f(t^{n,1}, y^{n,1}) = f(t^n, y^n)$. The $i - th$ intermediate time, the $i - th$ intermediate value, and the $i - th$ intermediate evaluation of the f function between t^n and $t^n + \Delta t$ are denoted by $t^{n,i}$, $y^{n,i}$ and $K^{n,i}$. The value at time $t^n + \Delta t$ is given by the following formula:

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^s b_i K^{n,i} \quad (9)$$

The method is defined by its coefficients $A = \{a_{ij}\}$, $b = \{b_i\}$ and $c = \{c_i\}$ that can be represented in a Butcher tableau (see [20]) :

$c_1 = 0$	0					
c_2	a_{21}					
c_3	a_{31}	a_{32}				
.	.					
.	.					
.	.					
c_s	a_{s1}	a_{s2}	.	.	.	$a_{s,s-1}$
	b_1	b_2	.	.	.	b_{s-1} b_s

Table 1: Butcher tableau of an s-stage explicit Runge-Kutta method

All Runge-Kutta methods in this article satisfy the property of consistency :

$$c_i = \sum_{j=1}^{i-1} a_{ij}. \tag{10}$$

1.3 Partitioned Runge-Kutta methods

In [5] and [17], Partitioned Runge-Kutta (PRK) methods are applied to temporal multiscale problems.

Let us consider a 1D computational domain composed of n cells (figure 1). The finite volume method applied to the one-dimensional hyperbolic equation (1) on the whole domain leads to the autonomous system of ordinary differential equations:

$$\frac{dy}{dt} = f(y), \quad y \in R^n. \tag{11}$$

As shown in figure 1, the computational domain is divided into two subdomains which have different characteristic times. The first subdomain has a slow characteristic time and contains cells from cell 1 to cell i_0 (large cells). In the following, this subdomain is referred to as the "slow subdomain". The second subdomain has a fast characteristic time and contains cells from cell i_0+1 to cell n (small cells). In the following, this subdomain is referred to as the "fast subdomain".

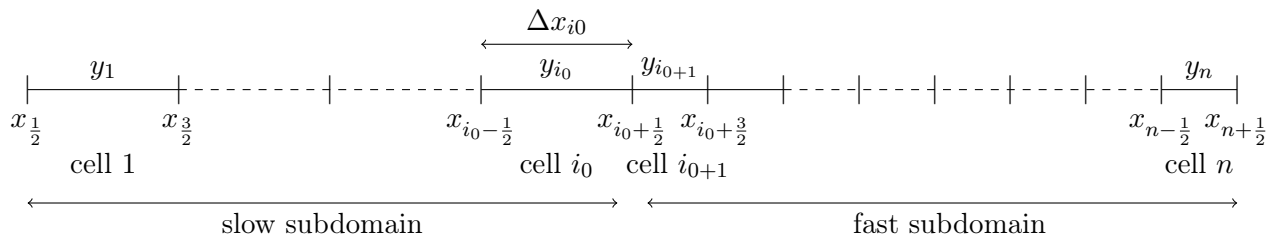


Figure 1: Partitioned computational domain

We denote by $Y_S = [y_1, y_2, \dots, y_{i_0}]$ the vector of unknowns in the slow subdomain and by $Y_F = [y_{i_0+1}, y_{i_0+2}, \dots, y_n]$ the vector of unknowns in the fast subdomain. System (11) can be rewritten using the variables Y_S and Y_F :

$$\begin{cases} \frac{dY_S}{dt} = f_S(Y_S, Y_F) \\ \frac{dY_F}{dt} = f_F(Y_S, Y_F) \end{cases} \quad (12)$$

The dependency between vectors Y_S and Y_F in each equation of (12) occurs at the interface between the subdomains. The dissimilar time scale for Y_S and Y_F is a motivation for solving these equations with two Runge-Kutta methods that use a different time step. In the following, we will show that these methods are, in fact, derived from the same base method. This is useful for the generalization of the scheme to more than two time steps. We denote by RKS the s -stage Runge-Kutta method for the slow subdomain. It is used to solve the first equation and its coefficients are : A^S, b^S, c^S . Similarly, we denote by RKF the s -stage Runge-Kutta method for the fast subdomain. It is used to solve the second equation and its coefficients are : A^F, b^F, c^F . According to [20], the PRK method using RKS and RKF is given by the following formulas:

$$\begin{aligned} Y_S^{n+1} &= Y_S^n + \Delta t \sum_{i=1}^s b_i^S K_S^{n,i} & Y_F^{n+1} &= Y_F^n + \Delta t \sum_{i=1}^s b_i^F K_F^{n,i} \\ Y_S^{n,i} &= Y_S^n + \Delta t \sum_{j=1}^{i-1} a_{ij}^S K_S^{n,j} & Y_F^{n,i} &= Y_F^n + \Delta t \sum_{j=1}^{i-1} a_{ij}^F K_F^{n,j} \end{aligned} \quad (13)$$

$$K_S^{n,i} = f_L(Y_S^{n,i}, Y_F^{n,i}) \quad K_F^{n,i} = f_F(Y_S^{n,i}, Y_F^{n,i})$$

where $1 \leq i \leq s$ with $K_S^{n,1} = f_S(Y_S^n, Y_F^n)$ and $K_F^{n,1} = f_F(Y_S^n, Y_F^n)$. We remind that Y_S^n and Y_F^n denote the vectors of unknowns at time t^n in the slow subdomain and in the fast subdomain, respectively. Y_S^{n+1} and Y_F^{n+1} are the vectors of unknowns at time $t^n + \Delta t$. The i -th intermediate slow solution and the i -th evaluation of the slow function f_S between t^n and $t^n + \Delta t$ are denoted by $Y_S^{n,i}$ and $K_S^{n,i}$. The i -th intermediate fast solution and the i -th evaluation of the fast function f_F between t^n and $t^n + \Delta t$ are denoted by $Y_F^{n,i}$ and $K_F^{n,i}$.

In most explicit local time stepping schemes (for instance [5], [7], [6] and [8]), RKS and RKF methods are based on the same RK method. RKF method is composed of two successive applications of the base method with the time step $\frac{\Delta t}{2}$, and some additional values to make the transition with the RKS method. RKS method is composed of the base method with the time step Δt , and some additional values to make the transition with the RKF method. With this strategy, an explicit local time stepping scheme with time steps Δt in the slow subdomain and $\frac{\Delta t}{2}$ in the fast subdomain is obtained. To have a Partitioned Runge-Kutta method of order p and therefore, a local time stepping scheme of order p , both RKS and RKF methods must be of order p but this is not sufficient. The coefficients of the methods have to satisfy additional coupling conditions. Following the methodology of [20] we can derive the following conditions on the Runge-Kutta coefficients to obtain a PRK method of order two and three, respectively (tables 2 and 3).

RKS of order 2	RKF of order 2	Coupling conditions
$\sum_{i=1}^s b_i^S = 1$	$\sum_{i=1}^s b_i^F = 1$	$\sum_{i=1}^s b_i^S c_i^F = \frac{1}{2}$
$\sum_{i=1}^s b_i^S c_i^S = \frac{1}{2}$	$\sum_{i=1}^s b_i^F c_i^F = \frac{1}{2}$	$\sum_{i=1}^s b_i^F c_i^S = \frac{1}{2}$

Table 2: Conditions to obtain a second order accurate PRK method

RKS of order 3	RKF of order 3	Coupling conditions	
$\sum_{i=1}^s b_i^S = 1$	$\sum_{i=1}^s b_i^F = 1$	$\sum_{i=1}^s b_i^S c_i^F = \frac{1}{2}$	$\sum_{i=1}^s b_i^F c_i^S = \frac{1}{2}$
$\sum_{i=1}^s b_i^S c_i^S = \frac{1}{2}$	$\sum_{i=1}^s b_i^F c_i^F = \frac{1}{2}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^S a_{ij}^F c_j^S = \frac{1}{6}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^F a_{ij}^S c_j^F = \frac{1}{6}$
$\sum_{i=1}^s b_i^S (c_i^S)^2 = \frac{1}{3}$	$\sum_{i=1}^s b_i^F (c_i^F)^2 = \frac{1}{3}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^S a_{ij}^F c_j^F = \frac{1}{6}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^F a_{ij}^S c_j^S = \frac{1}{6}$
$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^S a_{ij}^S c_j^S = \frac{1}{6}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^F a_{ij}^F c_j^F = \frac{1}{6}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^S a_{ij}^S c_j^F = \frac{1}{6}$	$\sum_{i=1}^s \sum_{j=1}^{i-1} b_i^F a_{ij}^F c_j^S = \frac{1}{6}$
		$\sum_{i=1}^s b_i^S c_i^F c_i^S = \frac{1}{3}$	$\sum_{i=1}^s b_i^F c_i^F c_i^S = \frac{1}{3}$

Table 3: Conditions to obtain a third order accurate PRK method

The number of conditions quickly increases with the order of accuracy which makes the implementation of high order local time stepping schemes difficult.

1.4 Constantinescu and Sandu strategy

In [5], Constantinescu and Sandu have developed a strategy, based on PRK methods, to obtain conservative and second order accurate local time stepping schemes. In this subsection, we briefly describe two schemes based on this strategy. The first one is the original scheme, based on a RK2 method [5]. The second one follows the same strategy but is based on a RK3 method (see [8, 11]). A comparison between these schemes and the schemes we propose (that are based on a different strategy, see part 3) is performed in part 4.

1.4.1 First scheme : original scheme of Constantinescu and Sandu

The original scheme developed by Constantinescu and Sandu is based on the second order Heun method:

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Table 4: Base method for the original scheme of Constantinescu and Sandu

The RKF method (Table 6) is obtained by applying twice successively the base method with the time step $\frac{\Delta t}{2}$. The RKS method (Table 5) is obtained by applying twice successively the base method with the time step Δt and dividing all b_i coefficients by 2.

0	0			
1	1	0		
0	0	0	0	
1	0	0	1	0
<hr/>				
	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Table 5: RKS method

0	0			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	
1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	0
<hr/>				
	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Table 6: RKF method

Both RKS and RKF methods are second order accurate and we have $b_i^S = b_i^F$ for every i . These properties ensure that this local time stepping scheme is second order accurate [5]. The equality of coefficients b_i ensures that the scheme is mass conservative [5]. In [5], the authors demonstrate that the RKS method described in Table 5 is only applied in a buffer zone. It is an area of variable length (the length depends on the spatial stencil), adjacent to the interface. Outside this buffer zone, RKS reduces to the Heun method described in Table 4. Finally, our semi-discrete approximation (4) is solved by the algorithm shown in table 7. We introduce the following notations for this algorithm :

- for all n , the values at times t^n and t^{n+1} (with $t^{n+1} = t^n + \Delta t$, Δt corresponding to the largest time step) are denoted by y^n and y^{n+1} ,
- for all n , the k -th intermediate value between times t^n and t^{n+1} is denoted by $y^{n,k}$.

According to (3) and the notations given above :

- for all n , $D_i^n = \frac{1}{\Delta x}(F_{i+\frac{1}{2}}(y_i^n, y_{i+1}^n) - F_{i-\frac{1}{2}}(y_{i-1}^n, y_i^n))$,
- for all n , the k -th intermediate flux balance on cell i is given by : $D_i^{n,k} = \frac{1}{\Delta x}(F_{i+\frac{1}{2}}(y_i^{n,k}, y_{i+1}^{n,k}) - F_{i-\frac{1}{2}}(y_{i-1}^{n,k}, y_i^{n,k}))$.

	Slow subdomain		Fast subdomain
1	$y_i^{n,1} = y_i^n - \Delta t D_i^n$	$y_i^{n,1} = y_i^n - \Delta t D_i^n$	$y_i^{n,1} = y_i^n - \frac{\Delta t}{2} D_i^n$
2		$y_i^{n,2} = y_i^n$	$y_i^{n,2} = y_i^n - \frac{\Delta t}{4}[D_i^n + D_i^{n,1}]$
3		$y_i^{n,3} = y_i^n - \Delta t D_i^{n,2}$	$y_i^{n,3} = y_i^{n,2} - \frac{\Delta t}{2} D_i^{n,2}$
4	$y_i^{n+1} = y_i^n - \frac{\Delta t}{2}[D_i^n + D_i^{n,1}]$	$y_i^{n+1} = y_i^n - \frac{\Delta t}{4}[D_i^n + D_i^{n,1} + D_i^{n,2} + D_i^{n,3}]$	$y_i^{n+1} = y_i^{n,2} - \frac{\Delta t}{4}[D_i^{n,2} + D_i^{n,3}]$

Buffer zone

Table 7: Algorithm of the original scheme of Constantinescu and Sandu

We can notice that this scheme is internally inconsistent, since at the interface, some values estimated at different times are mixed in the flux balance D . For instance, at the interface, $D_i^{n,2}$ contains values estimated at time t^n ($y_i^{n,2}$ from the buffer zone) and values estimated at time $t^{n+\frac{1}{2}}$ ($y_i^{n,2}$ from the fast subdomain). Despite this inconsistency, the scheme is second order accurate and mass-conservative.

For each cell in the buffer zone, four flux balances have to be calculated, therefore the computational cost is the same as in the fast subdomain. In our applications, we use a flux function with a four cell stencil. In this case, it can be proven that the size of the buffer zone is equal to four cells.

1.4.2 Second scheme

In [8] and [11], a RK3 method is used as the base method and the strategy of Constantinescu and Sandu is applied. Here, we extend this approach using a RK3 low-storage scheme [21] as the base method. Low-storage Runge-Kutta schemes require less storage than "classical" Runge-Kutta schemes. The RK3 low-storage method applied to (11) reads :

Step 1	$K^n = \Delta t f(t^n, y^n)$ $y^{n,1} = y^n + \beta_1 K^n$	$\beta_1 = \frac{1}{2}$
Step 2	$K^{n,1} = \alpha_2 K^n + \Delta t f(t^n + c_2 \Delta t, y^{n,1})$ $y^{n,2} = y^{n,1} + \beta_2 K^{n,1}$	$\alpha_2 = -0.6830127$ $c_2 = \frac{1}{2}$ $\beta_2 = 0.9106836$
Step 3	$K^{n,2} = \alpha_3 K^{n,1} + \Delta t f(t^n + c_3 \Delta t, y^{n,2})$ $y^{n+1} = y^{n,2} + \beta_3 K^{n,2}$	$\alpha_3 = -\frac{4}{3}$ $\beta_3 = 0.3660254$ $c_3 = 0.7886751$

Table 8: The RK3 low-storage scheme written in "low-storage form"

The values of the coefficients α and β have been proposed by Lowery and Reynolds [22]. Successive values of K and y overwrite the previous ones so that at any stage only $2N$ storage locations are required. It is to be noticed that this scheme is not total variation diminishing (TVD). Indeed, in [23] the authors show that a Runge-Kutta method written as a combination of Euler steps is TVD if all coefficients are positive. It is possible to show that this scheme does not satisfy this condition (because of the negative value of α_2 , for instance). The same method can be written in the "classical Runge-Kutta form" and a Butcher tableau can be associated:

Step 1	$K^n = \Delta t f(t^n, y^n)$ $y^{n,1} = y^n + a_{21} K^n$	$a_{21} = \beta_1$	<table border="1"> <tbody> <tr> <td>0</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>c_2</td> <td>a_{21}</td> <td></td> <td></td> </tr> <tr> <td>c_3</td> <td>a_{31}</td> <td>a_{32}</td> <td></td> </tr> <tr> <td></td> <td>b_1</td> <td>b_2</td> <td>b_3</td> </tr> </tbody> </table>	0	0			c_2	a_{21}			c_3	a_{31}	a_{32}			b_1	b_2	b_3
0	0																		
c_2	a_{21}																		
c_3	a_{31}	a_{32}																	
	b_1	b_2	b_3																
Step 2	$K^{n,1} = \Delta t f(t^n + c_2 \Delta t, y^{n,1})$ $y^{n,2} = y^n + a_{31} K^n + a_{32} K^{n,1}$	$a_{31} = \beta_1 + \beta_2 \alpha_2$ $a_{32} = \beta_2$ $c_2 = \frac{1}{2}$																	
Step 3	$K^{n,3} = \Delta t f(t^n + c_3 \Delta t, y^{n,2})$ $y^{n+1} = y^n + b_1 K^n + b_2 K^{n,1} + b_3 K^{n,2}$	$b_1 = a_{31} + \beta_3 \alpha_2 \alpha_3$ $b_2 = \beta_2 + \beta_3 \alpha_3$ $b_3 = \beta_3$ $c_3 = 0.7886751$																	

Table 9: The RK3 low-storage scheme written in "classical form" and its associated Butcher tableau

In the following, we only use the "classical form" described in table 9 for our RK3 scheme. Indeed, its associated Butcher tableau is very useful for constructing and describing local time stepping schemes.

Following the strategy of Constantinescu and Sandu, the base method is duplicated to construct RKS (table 10) and RKF (table 11) methods.

0	0					
c_2	a_{21}					
c_3	a_{31}	a_{32}				
0	0	0	0			
c_2	0	0	0	a_{21}		
c_3	0	0	0	a_{31}	a_{32}	
	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$

Table 10: RKS method

0	0					
$\frac{c_2}{2}$	$\frac{a_{21}}{2}$					
$\frac{c_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$				
$\frac{1}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$			
$\frac{1+c_2}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{21}}{2}$		
$\frac{1+c_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$	
	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$

Table 11: RKF method

Even though both RKS and RKF methods are third order accurate, the local time stepping scheme is only second order accurate because some third order coupling conditions are not satisfied [5]. However, the local time stepping is mass conservative [11]. Like for the original scheme of Constantinescu and Sandu, there is a buffer zone at the interface, in the slow subdomain. Outside this buffer zone, the RKS method reduces to the base method described in table 9. Finally, our semi-discrete approximation (4) is solved by the algorithm shown in table 12, in which the notations introduced in subsection 1.4 are re-used.

	Slow subdomain		Fast subdomain
1	$y_i^{n,1} = y_i^n - a_{21}\Delta t D_i^n$	$y_i^{n,1} = y_i^n - a_{21}\Delta t D_i^n$	$y_i^{n,1} = y_i^n - \frac{a_{21}}{2}\Delta t D_i^n$
2	$y_i^{n,2} = y_i^n - a_{31}\Delta t D_i^n$ $-a_{32}\Delta t D_i^{n,1}$	$y_i^{n,2} = y_i^n - a_{31}\Delta t D_i^n$ $-a_{32}\Delta t D_i^{n,1}$	$y_i^{n,2} = y_i^n - \frac{a_{31}}{2}\Delta t D_i^n$ $-\frac{a_{32}}{2}\Delta t D_i^{n,1}$
3		$y_i^{n,3} = y_i^n$	$y_i^{n,3} = y_i^n - \frac{b_1}{2}\Delta t D_i^n$ $-\frac{b_2}{2}\Delta t D_i^{n,1} - \frac{b_3}{2}\Delta t D_i^{n,2}$
4		$y_i^{n,4} = y_i^n - a_{21}\Delta t D_i^{n,3}$	$y_i^{n,4} = y_i^{n,3} - \frac{a_{21}}{2}\Delta t D_i^{n,3}$
5		$y_i^{n,5} = y_i^n - a_{31}\Delta t D_i^{n,3}$ $-a_{32}\Delta t D_i^{n,4}$	$y_i^{n,5} = y_i^{n,3} - \frac{a_{31}}{2}\Delta t D_i^{n,3}$ $-\frac{a_{32}}{2}\Delta t D_i^{n,4}$
6	$y_i^{n+1} = y_i^n - b_1\Delta t D_i^n$ $-b_2\Delta t D_i^{n,1} - b_3\Delta t D_i^{n,2}$	$y_i^{n+1} = y_i^n - \frac{b_1}{2}\Delta t(D_i^n + D_i^{n,3})$ $-\frac{b_2}{2}\Delta t(D_i^{n,1} + D_i^{n,4}) - \frac{b_3}{2}\Delta t(D_i^{n,2} + D_i^{n,5})$	$y_i^{n+1} = y_i^{n,3} - \frac{b_1}{2}\Delta t D_i^{n,3}$ $-\frac{b_2}{2}\Delta t D_i^{n,4} - \frac{b_3}{2}\Delta t D_i^{n,5}$

Buffer zone

Table 12: Algorithm of the second local time stepping scheme

The algorithm is written in "classical form" but it can be easily implemented in the low storage form since these two formulations are equivalent. Note that with a flux function with a four cell stencil, the size of the buffer zone is equal to 6 cells.

2 New schemes

In this section, we present a different approach to construct local time stepping schemes. Like in the strategy of Constantinescu and Sandu, we use the theory of PRK methods but here we guarantee the internal consistency of the scheme. This allows to obtain higher order accuracy, since we are able to design a third order scheme. Note that our schemes require a correction step to make them mass conservative. In these new schemes, the ratio between two adjacent time steps is restricted to 2. Several authors [9, 10, 7, 5] develop local time stepping schemes that can deal with ratios between 1 and 2 (2, 3/2, 4/3...) and even superior to 2 (3, 4...). In most meshes dedicated to LES or CFD in general, the grid size variation is rather continuous to guarantee the accuracy of the solution. The use of time step ratios superior to 2 is not appropriate since such grid size variations are not present. The use of several time step ratios between 1 and 2 allows a better adaptability to all grid size variations than the use of ratios restricted to 2. However, it tends to increase the algorithmic complexity. Indeed, the algorithm must be able to handle all possible ratios, while it has only 3 possibilities when the time step ratio is limited to 2. In this case, the neighbouring time step can be identical, equal to twice or equal to half the current time step. The choice to limit the ratio between adjacent time steps to 2 is a compromise between efficiency and algorithmic complexity.

2.1 First scheme: a second order time accurate scheme

Again, we use the second order Heun method (table 4) as the base method. It is applied twice with the time step $\frac{\Delta t}{2}$ to obtain the RKF method (table 14). Our RKS method is composed of the base method and some additional values to perform the transition with the RKF method and

achieve the desired order of accuracy. Thus, both RKS and RKF methods are of order 2. Now, we impose internal consistency by the relation $c_i^S = c_i^F$, for every i . It can be easily seen that second order coupling conditions (see table 2) are satisfied with this condition. Note that the intermediate time coefficients of RKF are already fixed: we have $c^F = \{0; \frac{1}{2}; \frac{1}{2}; 1\}$. For RKS, we have $c_1^S = 0$ and $c_4^S = 1$, but c_2^S and c_3^S are still undetermined. Then the internal consistency is obtained with $c_2^S = c_3^S = \frac{1}{2}$. This requires to define additional intermediate values at time $t^n + \frac{1}{2}\Delta t$ at the interface, in the slow subdomain. The coefficients a_{ij} needed for the calculation of these additional value are determined by the consistency relation (10). We remind that these additional values are only calculated at the interface and are not used for the calculation of y^{n+1} in the slow subdomain. The Butcher tableau of RKS and RKF methods read :

0	0			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	$\frac{1}{2}$	0	0	
1	1	0	0	0
<hr/>				
	$\frac{1}{2}$	0	0	$\frac{1}{2}$

Table 13: RKS method

0	0			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	
1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	0
<hr/>				
	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Table 14: RKF method

However, this local time stepping scheme defined by RKS and RKF is not mass conservative. It can be shown by applying RKS and RKF methods to (2) on cells i_0 and i_{0+1} , respectively. We remind that cell i_0 is the last cell of the slow subdomain while cell i_{0+1} is the first cell of the fast subdomain (see figure 1). The application of RKS and RKF methods to (2) on cells i_0 and i_{0+1} is shown in table 15. Note that to clarify the presentation, we assume that the flux function has a 2 cell stencil: $F_{i+\frac{1}{2}} = F_{i+\frac{1}{2}}(y_i, y_{i+1})$. The following notations are introduced for this algorithm :

- for all n , the values at times t^n and t^{n+1} (with $t^{n+1} = t^n + \Delta t$, Δt corresponding to the largest time step) are denoted by y^n and y^{n+1} ,
- for all n , the k -th intermediate value between times t^n and t^{n+1} is denoted by $y^{n,k}$.
- for all n , $F_{i+\frac{1}{2}}^n = F_{i+\frac{1}{2}}^n(y_i^n, y_{i+1}^n)$,
- for all n , the k -th intermediate flux at the interface $x_{i+\frac{1}{2}}$ is given by : $F_{i+\frac{1}{2}}^{n,k} = F_{i+\frac{1}{2}}^{n,k}(y_i^{n,k}, y_{i+1}^{n,k})$.

step	Solution on cell i_0	Solution on cell i_{0+1}
1	$y_{i_0}^{n,1} = y_{i_0}^n - \frac{\Delta t}{2\Delta x_{i_0}}(F_{i_0+\frac{1}{2}}^n - F_{i_0-\frac{1}{2}}^n)$	$y_{i_{0+1}}^1 = y_{i_{0+1}}^{n,1} - \frac{\Delta t}{2\Delta x_{i_{0+1}}}(F_{i_0+\frac{3}{2}}^n - F_{i_0+\frac{1}{2}}^n)$
2	$y_{i_0}^{n,2} = y_{i_0}^{n,1}$	$y_{i_{0+1}}^{n,2} = y_{i_{0+1}}^n - \frac{\Delta t}{4\Delta x_{i_{0+1}}}(F_{i_0+\frac{3}{2}}^n + F_{i_0+\frac{3}{2}}^{n,1} - F_{i_0+\frac{1}{2}}^n - F_{i_0+\frac{1}{2}}^{n,1})$
3	$y_{i_0}^{n,3} = y_{i_0}^n - \frac{\Delta t}{\Delta x_{i_0}}(F_{i_0+\frac{1}{2}}^n - F_{i_0-\frac{1}{2}}^n)$	$y_{i_{0+1}}^{n,3} = y_{i_{0+1}}^{n,2} - \frac{\Delta t}{2\Delta x_{i_{0+1}}}(F_{i_0+\frac{3}{2}}^{n,2} - F_{i_0+\frac{1}{2}}^{n,2})$
4	$y_{i_0}^{n+1} = y_{i_0}^n - \frac{\Delta t}{2\Delta x_{i_0}}(F_{i_0+\frac{1}{2}}^n + F_{i_0+\frac{1}{2}}^{n,1} - F_{i_0-\frac{1}{2}}^n - F_{i_0-\frac{1}{2}}^{n,1})$	$y_{i_{0+1}}^{n+1} = y_{i_{0+1}}^2 - \frac{\Delta t}{4\Delta x_{i_{0+1}}}(F_{i_0+\frac{3}{2}}^{n,2} + F_{i_0+\frac{3}{2}}^{n,3} - F_{i_0+\frac{1}{2}}^{n,2} - F_{i_0+\frac{1}{2}}^{n,3})$

Table 15: RKS and RKF methods applied on cells i_0 and i_{0+1}

We analyze the expression of y_{i0}^{n+1} . We notice that, between t^n and $t^n + \Delta t$, the total flux computed with the RKS method through the interface $x_{i0+\frac{1}{2}}$ is :

$$tf_{i0+\frac{1}{2}}^{RKS} = \frac{1}{2}[F_{i0+\frac{1}{2}}^n + F_{i0+\frac{1}{2}}^{n,1}] \quad (14)$$

The same analysis can be made on the other side of the interface. ~~The Total Flux Received by the short time scale subdomain through the interface $x_{i0+\frac{1}{2}}$, denoted by $TFR_{i0+\frac{1}{2}}$~~ The total flux computed with the RKF method through the interface $x_{i0+\frac{1}{2}}$ is:

$$tf_{i0+\frac{1}{2}}^{RKF} = \frac{1}{4}[F_{i0+\frac{1}{2}}^n + F_{i0+\frac{1}{2}}^{n,1} + F_{i0+\frac{1}{2}}^{n,2} + F_{i0+\frac{1}{2}}^{n,3}] \quad (15)$$

It is clear that total fluxes ~~$TFR_{i0+\frac{1}{2}}$ and $TFR_{i0+\frac{1}{2}}$~~ $tf_{i0+\frac{1}{2}}^{RKS}$ and $tf_{i0+\frac{1}{2}}^{RKF}$ are different and consequently the local time stepping scheme is not mass-conservative at the interface between the subdomains.

We propose a correction step to make the scheme conservative. Following the notations given above, y_{i0}^{n+1} reads :

$$y_{i0}^{n+1} = y_{i0}^n - \frac{\Delta t}{\Delta x_{i0}} tf_{i0+\frac{1}{2}}^{RKS} + \frac{\Delta t}{2\Delta x_{i0}} (F_{i0-\frac{1}{2}}^n + F_{i0-\frac{1}{2}}^{n,1}) \quad (16)$$

This value is corrected by adding the total flux difference. This leads to the following expression for the corrected value $y_{i0}^{n+1,c}$:

$$y_{i0}^{n+1,c} = y_{i0}^{n+1} + \frac{\Delta t}{\Delta x_{i0}} tf_{i0+\frac{1}{2}}^{RKS} - \frac{\Delta t}{\Delta x_{i0}} tf_{i0+\frac{1}{2}}^{RKF} \quad (17)$$

We substitute y_{i0}^{n+1} by its expression given by (16). The corrected value $y_{i0}^{n+1,c}$ now reads:

$$y_{i0}^{n+1,c} = y_{i0}^n - \frac{\Delta t}{\Delta x_{i0}} tf_{i0+\frac{1}{2}}^{RKF} + \frac{\Delta t}{2\Delta x_{i0}} (F_{i0-\frac{1}{2}}^n + F_{i0-\frac{1}{2}}^{n,1}) \quad (18)$$

In (18), the total flux computed by the RKS method through the interface $x_{i0+\frac{1}{2}}$, $tf_{i0+\frac{1}{2}}^{RKS}$, has been replaced by the total flux computed by the RKF method through the interface $x_{i0+\frac{1}{2}}$, $tf_{i0+\frac{1}{2}}^{RKF}$. Consequently, the corrected scheme is mass conservative.

Time order and spatial order studies made on the different test cases in section 3 show that this correction step does not alter the time order of accuracy nor the spatial order of accuracy of the solution.

The algorithm of this local time stepping applied to our semi-discrete approximation (4) is shown in table 16. In this algorithm, the notations introduced in subsection 1.4 are re-used.

	Slow subdomain		Fast subdomain
1	$y_i^{n,3} = y_i^n - \Delta t D_i^n$	$y_i^{n,1} = y_i^n - \frac{\Delta t}{2} D_i^n$	$y_i^{n,1} = y_i^n - \frac{\Delta t}{2} D_i^n$
2		$y_i^{n,2} = y_i^{n,1}$	$y_i^{n,2} = y_i^n - \frac{\Delta t}{4} [D_i^n + D_i^{n,1}]$
3		$y_i^{n,3} = y_i^n - \Delta t D_i^n$	$y_i^{n,3} = y_i^{n,2} - \frac{\Delta t}{2} D_i^{n,2}$
4	$y_i^{n+1} = y_i^n - \frac{\Delta t}{2} [D_i^n + D_i^{n,3}]$	$y_i^{n+1} = y_i^n - \frac{\Delta t}{2} [D_i^n + D_i^{n,3}]$	$y_i^{n+1} = y_i^{n,2} - \frac{\Delta t}{4} [D_i^{n,2} + D_i^{n,3}]$
5		Correction step (last cell)	

Buffer zone

Table 16: Algorithm of the first proposed scheme (second order time accurate)

In our applications, with a four cell stencil flux function, the size of the buffer zone is equal to two cells, while it is equal to four cells for the original scheme of Constantinescu and Sandu. Moreover, only two flux evaluations (D_i^n and $D_i^{n,3}$) are needed in the buffer zone whereas the original scheme of Constantinescu and Sandu requires four flux evaluations in this zone. Although its buffer zone is larger, Constantinescu and Sandu scheme is mass-conservative and TVB (Total Variation Bounded) if both temporal and spatial scheme are TVD (Total Variation Diminishing) [5]. Our scheme requires a correction step to be mass conservative and we didn't design our scheme to be TVB or TVD.

2.2 Second scheme: a third order time accurate scheme

The base method used to construct this local time stepping scheme is the RK3 low storage method described in table 9. The construction of RKF method is performed with two successive applications of the base method with the time step $\frac{\Delta t}{2}$. Consequently, RKF method has the following intermediate time coefficients: $c^F = \{0; \frac{c_2}{2}; \frac{c_3}{2}; \frac{1}{2}; \frac{1+c_2}{2}; \frac{1+c_3}{2}\}$. For the construction of RKS method, the base method is applied once with a time step Δt , which leads to the following intermediate time coefficients: $c^S = \{0; c_2 = \frac{1}{2}; c_3\}$. As for the first proposed scheme, we impose internal consistency by the relations $c_i^S = c_i^F$, for every i . Indeed, this property implies that some of the third order coupling conditions of table 3 are necessarily satisfied (namely that of lines 1,4,5). To satisfy internal consistency, an additional value at time $t + c_3 \Delta t$ must be calculated in RKF method. In RKS method, additional values at times $t + \frac{c_2}{2} \Delta t$, $t + \frac{c_3}{2} \Delta t$, $t + \frac{1+c_2}{2} \Delta t$ and $t + \frac{1+c_3}{2} \Delta t$ must be computed. RKS and RKF methods are now two third order Runge-Kutta methods with 7 stages. Their intermediate time coefficients are the same and read: $\{0; \frac{c_2}{2}; \frac{c_3}{2}; \frac{1}{2}; \frac{1+c_2}{2}; c_3; \frac{1+c_3}{2}\}$. The Runge-Kutta coefficients for the calculation of additional values, that are still to be determined, are denoted by α^S in RKS method and by α^F in RKF method. The Butcher tableaux of RKS and RKF methods are shown in tables 17 and 18, respectively.

0	0						
$\frac{c_2}{2}$	α_1^S						
$\frac{c_3}{2}$	α_2^S	α_3^S					
$\frac{1}{2}$	a_{21}	0	0				
$\frac{1+c_2}{2}$	α_4^S	α_5^S	α_6^S	α_7^S			
c_3	a_{31}	0	0	a_{32}	0		
$\frac{1+c_3}{2}$	α_8^S	α_9^S	α_{10}^S	α_{11}^S	α_{12}^S	α_{13}^S	
	b_1	0	0	b_2	0	b_3	0

Table 17: RKS method

0	0						
$\frac{c_2}{2}$	$\frac{a_{21}}{2}$						
$\frac{c_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$					
$\frac{1}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$				
$\frac{1+c_2}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{21}}{2}$			
c_3	α_1^F	α_2^F	α_3^F	α_4^F	α_5^F		
$\frac{1+c_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$	0	
	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	0	$\frac{b_3}{2}$

Table 18: RKF method

The unknown coefficients α_i^S and α_i^F will be determined so that the coupling conditions of lines 2 and 3 of table 3 are satisfied. Some of them are equivalent, therefore the independent coupling conditions read:

$$\sum_{i=1}^s \sum_{j=1}^s b_i^S a_{ij}^F c_j^F = \frac{1}{6}, \quad (19)$$

$$\sum_{i=1}^s \sum_{j=1}^s b_i^F a_{ij}^S c_j^S = \frac{1}{6}, \quad (20)$$

The coefficients for the computation of additional values in RKS method have to satisfy (20) and four relations given by the consistency property (10). This gives 5 relations that can be used to determine 5 unknowns only. The other eight coefficients can be set to any arbitrary value. A simple and natural choice is to set: $\alpha_3^S = \alpha_5^S = \alpha_6^S = \alpha_7^S = \alpha_9^S = \alpha_{10}^S = \alpha_{12}^S = \alpha_{13}^S = 0$, while the resolution of the system leads to the following values for the other coefficients:

$$\left\{ \begin{array}{l} \alpha_1^S = \frac{c_2}{2} \\ \alpha_2^S = \frac{c_3}{2} \\ \alpha_4^S = \frac{1+c_2}{2} \\ \alpha_{11}^S = \frac{2}{3b_3} \approx 1.821 \\ \alpha_8^S = \frac{1+c_3}{2} - \frac{2}{3b_3} \approx -0.9270 \end{array} \right. \quad (21)$$

A similar analysis leads to the following values for the coefficients dedicated to the calculation

of additional values in RKF method:

$$\begin{cases} \alpha_2^F = \left(\frac{2}{3} - \frac{b_2}{2}\right) \frac{1}{b_3} \approx 1.2440 \\ \alpha_1^F = c_3 - \alpha_2^F \approx -0.45534 \\ \alpha_3^F = 0 \\ \alpha_4^F = 0 \\ \alpha_5^F = 0 \end{cases} \quad (22)$$

As for our previous second order scheme, the local time stepping scheme defined by RKS and RKF methods described in tables 17 and 18 is not mass conservative. The strategy based on the Total Flux Lost and the Total Flux Received at the interface between subdomains can be applied to make the scheme conservative. The algorithm of this local time stepping scheme applied to solve (4) is shown in table 19. In this algorithm, the notations introduced in subsection 1.4 are re-used.

	Slow subdomain		Fast subdomain	
1		$y_i^{n,1} = y_i^n - \alpha_1^S \Delta t D_i^n$	$y_i^{n,1} = y_i^n - \frac{a_{21} \Delta t}{2} D_i^n$	$y_i^{n,1} = y_i^n - \frac{a_{21} \Delta t}{2} D_i^n$
2		$y_i^{n,2} = y_i^n - \alpha_2^S \Delta t D_i^n$	$y_i^{n,2} = y_i^n - \frac{a_{31} \Delta t}{2} D_i^n$ $- \frac{a_{32} \Delta t}{2} D_i^{n,1}$	$y_i^{n,2} = y_i^n - \frac{a_{31} \Delta t}{2} D_i^n$ $- \frac{a_{32} \Delta t}{2} D_i^{n,1}$
3	$y_i^{n,3} = y_i^n - a_{21} \Delta t D_i^n$	$y_i^{n,3} = y_i^n - a_{21} \Delta t D_i^n$	$y_i^{n,3} = y_i^n - \frac{b_1 \Delta t}{2} D_i^n$ $- \frac{b_2 \Delta t}{2} D_i^{n,1} - \frac{b_3 \Delta t}{2} D_i^{n,2}$	$y_i^{n,3} = y_i^n - \frac{b_1 \Delta t}{2} D_i^n$ $- \frac{b_2 \Delta t}{2} D_i^{n,1} - \frac{b_3 \Delta t}{2} D_i^{n,2}$
4		$y_i^{n,4} = y_i^n - \alpha_4^S \Delta t D_i^n$	$y_i^{n,4} = y_i^{n,3} - \frac{a_{21} \Delta t}{2} D_i^{n,3}$	$y_i^{n,4} = y_i^{n,3} - \frac{a_{21} \Delta t}{2} D_i^{n,3}$
5	$y_i^{n,5} = y_i^n - a_{31} \Delta t D_i^n$ $- a_{32} \Delta t D_i^{n,3}$	$y_i^{n,5} = y_i^n - a_{31} \Delta t D_i^n$ $- a_{32} \Delta t D_i^{n,3}$	$y_i^{n,5} = y_i^n - \alpha_1^F \Delta t D_i^n$ $- \alpha_2^F \Delta t D_i^{n,1}$	
6		$y_i^{n,6} = y_i^n - \alpha_8^S \Delta t D_i^n$ $- \alpha_{11}^S \Delta t D_i^4$	$y_i^{n,6} = y_i^{n,3} - \frac{a_{31} \Delta t}{2} D_i^{n,3}$ $- \frac{a_{32} \Delta t}{2} D_i^4$	$y_i^{n,6} = y_i^{n,3} - \frac{a_{31} \Delta t}{2} D_i^{n,3}$ $- \frac{a_{32} \Delta t}{2} D_i^4$
7	$y_i^{n+1} = y_i^n - b_1 \Delta t D_i^n$ $- b_2 \Delta t D_i^{n,3} - b_3 \Delta t D_i^{n,5}$	$y_i^{n+1} = y_i^n - b_1 \Delta t D_i^n$ $- b_2 \Delta t D_i^{n,3} - b_3 \Delta t D_i^{n,5}$	$y_i^{n+1} = y_i^{n,3} - \frac{b_1 \Delta t}{2} D_i^{n,3}$ $- \frac{b_2 \Delta t}{2} D_i^{n,4} - \frac{b_3 \Delta t}{2} D_i^{n,6}$	$y_i^{n+1} = y_i^{n,3} - \frac{b_1 \Delta t}{2} D_i^{n,3}$ $- \frac{b_2 \Delta t}{2} D_i^{n,4} - \frac{b_3 \Delta t}{2} D_i^{n,6}$
8		Correction step		

Buffer zone

Buffer zone

Table 19: Algorithm of the second proposed scheme (third order time accurate)

This algorithm is written in the Runge-Kutta "classical form" but it can be implemented in the "low-storage form", since these forms are equivalent. In this scheme, there are two buffer zones because both RKS and RKF methods require one or several additional values. The size of each buffer zone is equal to two cells in our applications (flux function with a four cell stencil). We remind that for the scheme of Constantinescu and Sandu based on RK3, there is only one buffer

zone of six cells and no correction step since this scheme is mass conservative. Moreover, if both temporal and spatial scheme are TVD, the scheme of Constantinescu and Sandu is TVB, which is not the case for our scheme.

3 Numerical tests

In this section, some numerical test cases are performed to compare the new proposed schemes and the schemes proposed in the literature that are based on the strategy of Constantinescu and Sandu. The schemes based on RK2 and RK3 which follow the strategy of Constantinescu and Sandu are denoted by CSRK2 and CSRK3, respectively. The schemes that we propose, based on RK2 and RK3, are denoted by Prop. RK2 and Prop. RK3, respectively.

Two different schemes are used for the Euler flux discretization. The Roe scheme [24] with a minmod limiter [25] is retained for the Sod tube, whereas an hybrid centered/upwind version of the scheme AUSM+(P) proposed by Mary and Sagaut [26] is employed in the others test cases. Viscous fluxes are discretized with a classical second order scheme.

3.1 Two-dimensional vortex advection

This test case is based on 2D compressible Euler equations for a perfect gas:

$$\begin{cases} \partial_t \rho + \partial_{x_j}(\rho u_j) = 0 \\ \partial_t \rho u_i + \partial_{x_j}(\rho u_i u_j + p \delta_{ij}) = 0 \\ \partial_t(\rho e) + \partial_{x_j}(\rho e u_j + p u_j) = 0 \end{cases} \quad (23)$$

with $1 \leq i, j \leq 2$. The variables ρ, u_1, u_2, p are the fluid density, the fluid velocity in directions \vec{x} and \vec{y} and the fluid pressure, respectively. Total energy is denoted by e and reads : $e = \frac{1}{\gamma-1} \frac{p}{\rho} + \frac{u_1^2 + u_2^2}{2}$, with $\gamma = 1.40$.

The initial state is a 2D uniform flow ($\rho = 1, u_1 = 1, u_2 = 0, p = 1$) with the superposition of perturbations $(\delta\rho, \delta u_1, \delta u_2, \delta p)$ which simulate a vortex. The perturbations read:

$$\begin{cases} \delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} \\ \delta u_1 = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(y_c - y) \\ \delta u_2 = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(x - x_c) \\ \delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} \end{cases} \quad (24)$$

where $\delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}$, is the temperature perturbation around the value $T = 1$. Note that the corresponding initial value and perturbation for the entropy are $S = 1$ and $\delta S = 0$, respectively. The coordinates (x_c, y_c) are the coordinates of the the vortex center, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the vortex radius and $\epsilon = 5$ is the vortex strength. The initial solution of this problem reads:

$$\begin{cases} \rho^0 = 1 + \delta\rho \\ u_1^0 = 1 + \delta u_1 \\ u_2^0 = 0 + \delta u_2 \\ p^0 = 1 + \delta p \end{cases} \quad (25)$$

The solution of this problem is a simple advection of the vortex in the \vec{x} direction. For more information about this test case, see [27].

The computational domain is rectangular (dimensions : 30×20) with periodic boundary conditions. It is divided into three zones. In the first and last zones, the grid spacings in directions \vec{x} and \vec{y} are $\Delta x = \Delta y = 0.05$. In the middle zone, the grid spacing in the \vec{x} direction is refined by a factor 2 ($\Delta x = 0.025$ and $\Delta y = 0.05$). Thus, the time step $\frac{\Delta t}{2}$ is used in this zone and the time step Δt is applied in the first and last zones. The mesh and the density field at initial time are shown in figure 2, while the density profile at initial time and $y = 10$ is plotted in Figure 3. The reference solution mentioned in figures 4 and 5 is obtained with a fourth order Runge-Kutta scheme (RK4) with $\Delta t = 0.0035$. The *CFL* number in the middle zone (the maximum *CFL* number over the mesh) is $CFL = 0.28$. For each local time stepping scheme, the time step used in the first and last zones (Δt) is the maximum allowable time step shown in the row "*CFL*max (Δt_{max})" of table 20.

Figures 4 and 5 show the density profile along $y = 10$ for each local time stepping scheme and for the reference solution at the final time 43.5. At this time, the vortex has returned approximately to its initial position (we remind that we are using periodic boundary conditions). A very good agreement between each scheme and the reference solution is observed.

Some characteristics studied at the final time are listed in table 20. The "speedup" refers to the relative difference between the CPU time of an explicit global time stepping scheme based on RK2 (which uses a uniform time step satisfying the most restrictive *CFL* condition) and the CPU time of the local time stepping schemes. The row "Conservation defect" is the relative difference between the total fluid mass in the computational domain at final time $m(t_f)$ and at initial time $m(t_0)$:

$$\text{Conservation defect} = \frac{m(t_f) - m(t_0)}{m(t_0)}$$

The local time stepping schemes based on RK3 have a better stability than the local time stepping schemes based on RK2, which allows to use larger time steps. It is the reason why the local time stepping schemes based on RK3 (CS RK3 and Prop. RK3) provide a better speedup than the local time stepping schemes based on RK2 (CS RK2 and Prop. RK2). The conservation defects obtained with our local time stepping schemes are similar to the ones obtained with the schemes of Constantinescu that are conservative. It shows that our flux-correction strategy to make our schemes conservative is efficient.

Figure 6 shows the results of a time order study. For each local time stepping scheme, a finest solution is computed with the time step $\Delta t = 10^{-4}$ in the first and last zones. We remind that in the middle zone the grid spacing and the time step are refined by a factor 2, therefore the *CFL* number is uniform over the mesh ($CFL = 0.008$). Three other simulations, with time steps equal to $2\Delta t$, $4\Delta t$ and $8\Delta t$ for the first and last zones are also run. The L_2 errors on the density field between the finest solution and the three other solutions are computed. The slope of each convergence curve is in agreement with the theoretical temporal order of all the local time stepping schemes. We can notice that the L_2 errors of the original scheme of Constantinescu and Sandu and our proposed

scheme based on RK2 are identical and show second order accuracy. For the schemes based on RK3, our proposed approach is really third order accurate, while CS RK3 is only a second-order accurate scheme leading to a higher level of L_2 error. Figure 7 shows the results of a space order study, for each local time stepping scheme. For this study, the spatial steps Δx and Δy are uniform (no space refinement in the middle zone). There is only a temporal refinement in the middle zone. The finest solution is computed with $\Delta x = \Delta y = 0.013$. The time step is taken sufficiently small in order to compute only the spatial error. The time step $\Delta t = 9 \cdot 10^{-4}$ ($CFL = 0.1$) is used in the first and last zones, while the time step $\Delta t = 4,5 \cdot 10^{-4}$ is used in the middle zone. Three other solutions are computed keeping the same time steps and with the spacial steps $(2\Delta x, 2\Delta y)$, $(3\Delta x, 3\Delta y)$, $(4\Delta x, 4\Delta y)$. The L_2 errors on the density field between the finest solution and the three other solutions are computed. The results show that the four local time stepping schemes have the same spatial order of accuracy. This order of accuracy is close to 2, the theoretical spatial order of accuracy for the AUSM+P scheme. Figures 6 and 7 demonstrate that our flux correction stage used in schemes "Prop. RK2" and "Prop. RK3" does not alter the time order of accuracy nor the spatial order of accuracy.

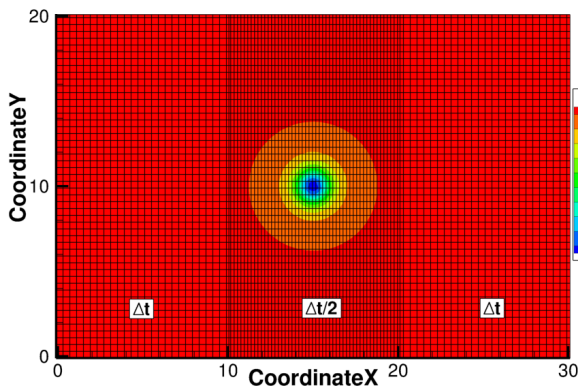


Figure 2: Density field at initial time and view of the mesh partitioning for the local time stepping computation.

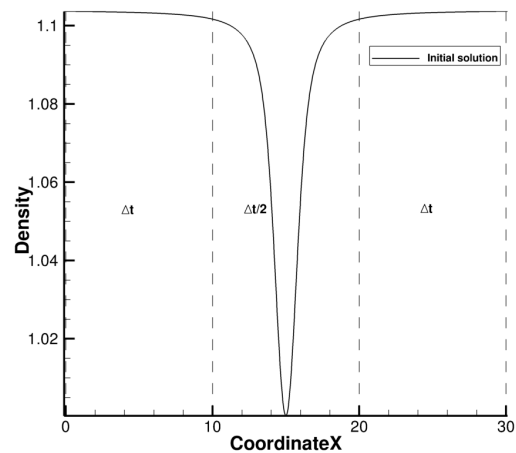


Figure 3: Initial density profile at $y = 10$

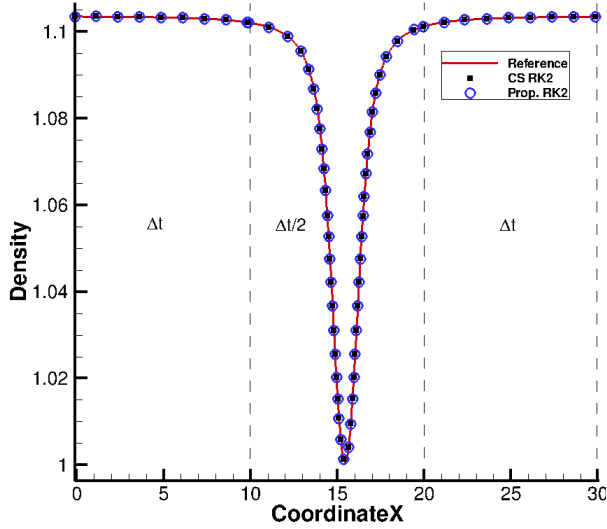


Figure 4: Density profiles at $y = 10$ and final time for the schemes Prop.RK2 and CS RK2 and the reference solution

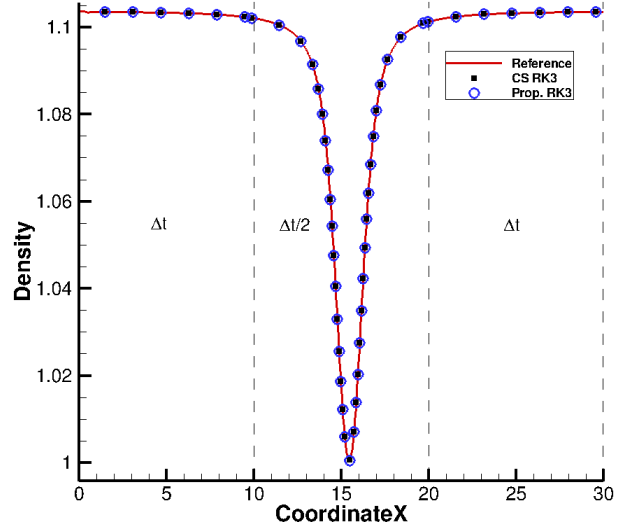


Figure 5: Density profiles at $y = 10$ and final time for Prop.RK3, CS RK3 schemes and the reference solution

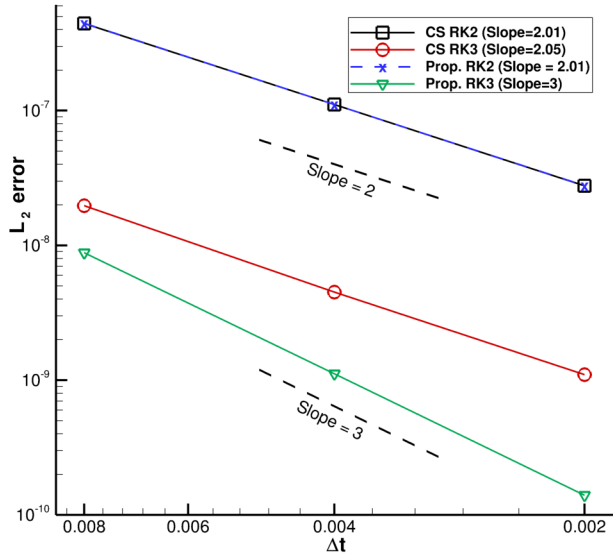


Figure 6: L_2 error as a function of the time step Δt for each local time stepping scheme

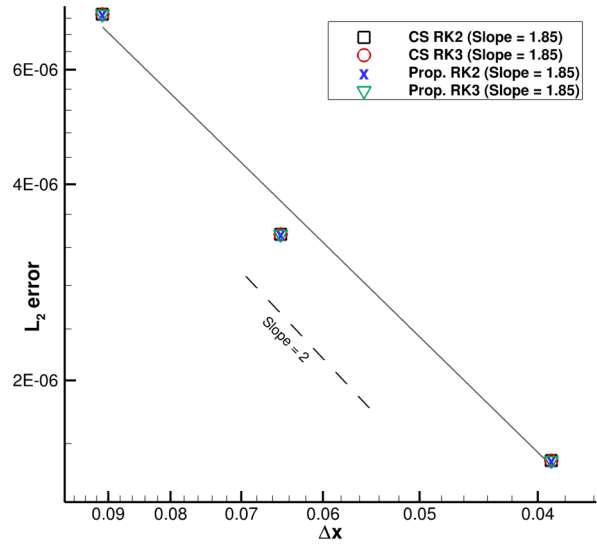


Figure 7: L_2 error as a function of the spatial step Δx for each local time stepping scheme

	$CFL_{\max}(\Delta t_{\max})$	Speedup as compared to a RK2 global time stepping scheme	Conservation defect
CS RK2	0.25 ($\Delta t_{\max} = 0.007$)	21%	$3.4 \cdot 10^{-16}$
Prop. RK2	0.25 ($\Delta t_{\max} = 0.007$)	25%	$1.7 \cdot 10^{-16}$
CS RK3	1.1 ($\Delta t_{\max} = 0.029$)	73%	$1.7 \cdot 10^{-16}$
Prop. RK3	1.1 ($\Delta t_{\max} = 0.029$)	74%	$9.1 \cdot 10^{-16}$

Table 20: Comparison of efficiency and conservation property indicators for the local time stepping schemes

3.2 Sod shock tube

The Sod tube problem is used to test if the proposed schemes are able to predict accurate and stable solution for a flow containing discontinuities. Note that this 1D problem is solved here in 2D. At the initial time the velocity field is equal to zero, while the density and pressure profiles are given by :

- $\rho^0(x, y) = 1, p^0(x, y) = 1$ for $x \leq 9$,
- $\rho^0(x, y) = 0.125, p^0(x, y) = 0.1$ for $x > 9$,

The rectangular computational domain (18×0.6) is divided into three zones of length 6. In the first and last zones, the grid spacings in the \vec{x} direction (Δx) and in the \vec{y} direction (Δy) are: $\Delta x = \Delta y = 0.01$. In the middle zone, $\Delta x = 0.005$ and $\Delta y = 0.01$. Thus, the time step $\frac{\Delta t}{2}$ is used in the middle zone and the time step Δt is applied in the first and last zones. The mesh partitioning and the initial density profile at $y = 0.3$ are plotted in figure (8).

Figures 9 and 10 show the density profile at $y = 0.3$ for each local time stepping scheme and for the reference solution at the final time, $t = 4.8$. The reference solution mentioned in these figures is obtained by using the classical RK4 explicit scheme with the time step $\Delta t = 5.10^{-4}$. The CFL number in the middle zone (the maximum CFL number over the mesh) is $CFL = 0.2$. The time step Δt of the local time stepping schemes (the time step in the first and last zones) is their maximal allowable time step shown in table 21 below. Figures 9 and 10 show a very good agreement between the reference solution and the solutions obtained with the local time stepping schemes. Table 21 also shows the speedup as compared to a RK2 global time stepping scheme, which uses a uniform time step satisfying the most restrictive CFL condition. Our local time stepping schemes have roughly the same speedup as their homologous schemes of Constantinescu and Sandu based on the same RK method.

The results of a time order study are plotted in figure 11. For each local time stepping scheme, a finest solution is computed with the time step $\Delta t = 2.5 \cdot 10^{-4}$ in the first and last zones. We remind that in the middle zone the grid spacing and the time step are refined by a factor 2, therefore the CFL number is uniform over the mesh ($CFL = 0.05$). Three other simulations, with time steps equal to $2\Delta t, 4\Delta t$ and $8\Delta t$ for the first and last zones are also run. The L_2 errors committed on the density field between the finest solution and the three other solutions are computed. The slope of each convergence curve is in agreement with the theoretical temporal order of the local time stepping schemes. Our local time stepping scheme "Prop. RK3" is third-order accurate and leads

to the lowest error. Figure 12 shows the results of a space order study, for each local time stepping scheme. This study is performed by employing the same method as in subsection 3.1. The finest solution is computed with $\Delta x = \Delta y = 0.0057$. The time step $\Delta t = 5 \cdot 10^{-4}$ ($CFL = 0.1$) is used in the first and last zones, while the time step $\Delta t = 2,5 \cdot 10^{-4}$ is used in the middle zone. The results show that the four local time stepping schemes have the same spatial order of accuracy. This order of accuracy is slightly superior to 1. This result is logical since our third order accurate Roe scheme uses a slope limiter which reduces the order of accuracy at discontinuities. Figures 11 and 12 demonstrate that our flux correction stage used in schemes "Prop. RK2" and "Prop. RK3" does not alter the time order of accuracy nor the spatial order of accuracy.

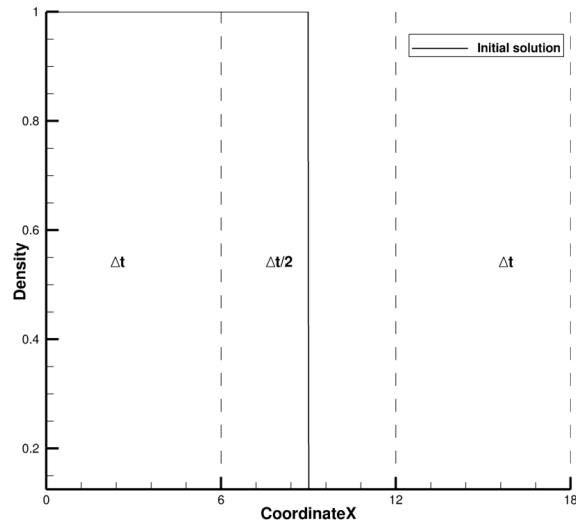


Figure 8: Initial density profile and view of the mesh partitioning for local time stepping scheme.

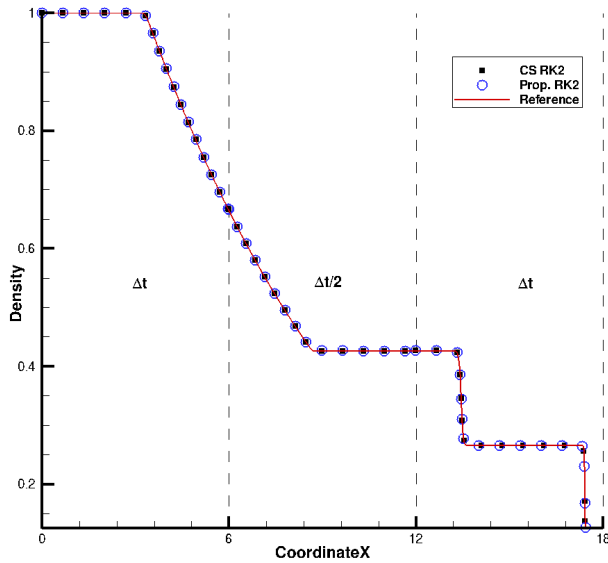


Figure 9: Density profile at the final time for the CS RK2 and Prop. RK2 schemes

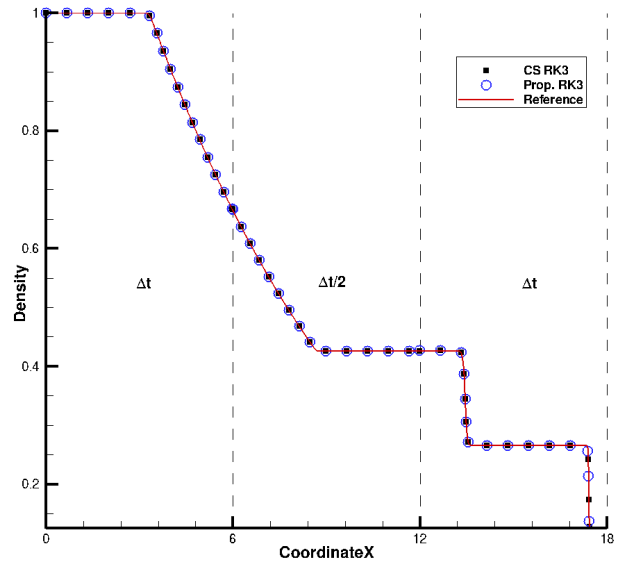


Figure 10: Density profile at the final time for the CS RK3 and Prop. RK3 schemes.

	CFLmax (Δt_{max})	Speedup as compared to a RK2 global time stepping scheme
CS RK2	0.7 ($\Delta t_{max} = 0.0032$)	19%
Prop. RK2	0.7 ($\Delta t_{max} = 0.0032$)	19%
CS RK3	0.9 ($\Delta t_{max} = 0.004$)	28%
Prop. RK3	0.9 ($\Delta t_{max} = 0.004$)	26%

Table 21: Comparison of the maximum CFL and the speedup of the local time stepping schemes for the Sod shock tube.

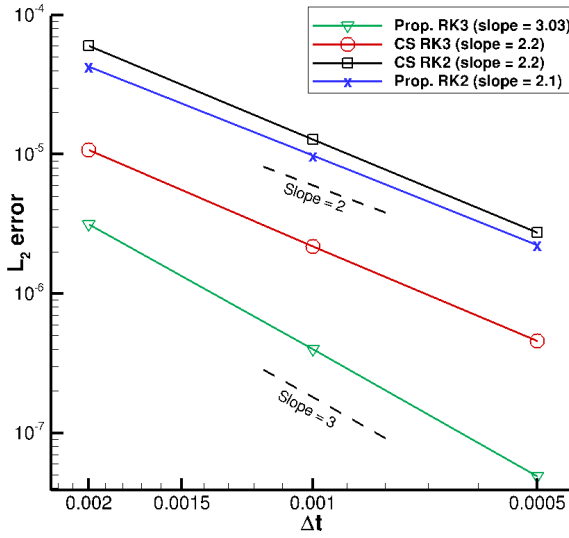


Figure 11: L_2 error as a function of the time step Δt for each local time stepping scheme

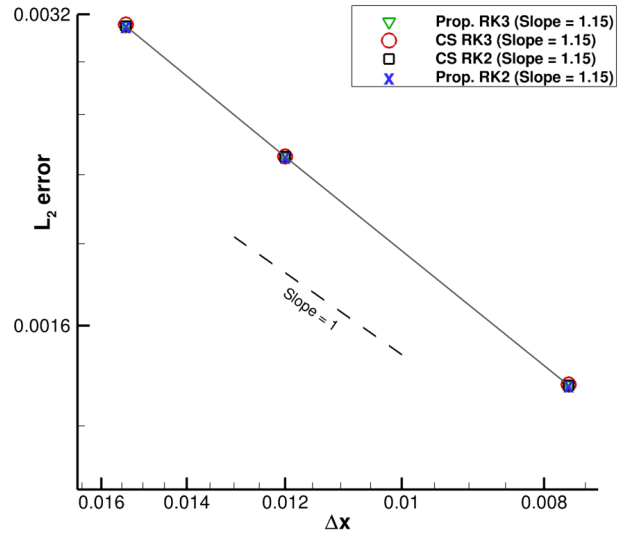


Figure 12: L_2 error as a function of the spatial step Δx for each local time stepping scheme

3.3 Taylor-Green Vortex

This test case is used to validate the proposed schemes in the context of DNS or LES simulation of turbulent flows. At the initial time, large vortices are present in the computational box and during the simulation an energy transfer occurs from the large scales to smaller ones. When the energy cascade is sufficient to reach the Kolmogorov scale, the kinetic energy is dissipated. For that case, the compressible 3D Navier-Stokes equations are solved for a viscous fluid :

$$\begin{cases} \partial_t \rho + \partial_{x_j}(\rho u_j) = 0 \\ \partial_t \rho + \partial_{x_j}(\rho u_i u_j + p \delta_{ij} - \tau_{ij}) = 0 \\ \partial_t(\rho e) + \partial_{x_j}(\rho e u_j + p u_j - \tau_{jk} u_k + q_j) = 0 \end{cases} \quad (26)$$

with $1 \leq i, j \leq 3$. The variable ρ is the fluid density, u_i are the fluid velocity components, e is the total specific energy of the fluid, p is the fluid pressure (ideal gas law), q_j is the heat flux, and τ_{ij}

is the tensor of viscous constraints.

The computational domain is a cube of dimension $2\pi L_0$ ($L_0 = 1$ is a reference length), with periodic boundary conditions. The initial conditions for variables u_i , p and ρ read :

$$\begin{cases} u_1^0 = U_0 \sin(x/L_0) \cos(y/L_0) \cos(z/L_0) \\ u_2^0 = -U_0 \cos(x/L_0) \sin(y/L_0) \cos(z/L_0) \\ u_3^0 = 0 \\ p^0 = P_0 + \rho_0 U_0^2 / 16 (\cos(x/L_0) + \cos(y/L_0)) (\cos(2z/L_0) + 2) \\ \rho^0 = p / (RT_0) \end{cases} \quad (27)$$

where $P_0 = 101183 Pa$, $\rho_0 = 1.2 kg.m^{-3}$, $T_0 = 294 K$ and $U_0 = 34.38 m.s^{-1}$. The simulation is performed during a period of $14L_0/U_0$.

The computational domain is divided into three zones of equal length, as shown in figure 13. In the first and last zones, the grid spacings in the \vec{x} direction (Δx), \vec{y} direction (Δy) and \vec{z} direction (Δz) are $\Delta x = \Delta y = \Delta z = 0.025$, respectively. In the middle zone, the mesh is refined by a factor two in the \vec{x} direction ($\Delta x = 0.0125$, $\Delta y = \Delta z = 0.025$). In this zone, the time step $\frac{\Delta t}{2}$ is used while the time step Δt is applied in the other zones. For each scheme, we compute at each time step the enstrophy ϵ , defined by

$$\epsilon = \frac{L_0}{U_0^2 \rho_0} \int_{\Omega} \rho \vec{w} \cdot \vec{w} d\Omega, \quad (28)$$

where \vec{w} denotes the vorticity, defined by : $\vec{w} = \vec{\nabla} \wedge \vec{V}$, with $\vec{V} = (u_1, u_2, u_3)$. The enstrophy is known to be highly sensitive to the numerical method [28].

Figures 14 and 15 show the time evolution of the enstrophy for each scheme. The reference solution mentioned in these figures is obtained with a classical RK4 explicit scheme used with the time step $\Delta t = 0.00015$. The CFL number in the middle zone (the maximum CFL number over the mesh) is $CFL = 0.03$. The time step Δt in the first and last zones of each local time stepping scheme is the maximal allowable time step, shown in the first row of table 22. A very good agreement is observed between the reference solution and the solutions obtained with the local time stepping schemes.

Table 22 also shows the speedup obtained with the local time stepping schemes as compared to an RK2 explicit global time stepping scheme, and the conservation defects. Our local time stepping schemes provide the same speedup as their homologous schemes based on the strategy of Constantinescu and Sandu. The conservation defects of our schemes are very close to the ones obtained with the schemes of Constantinescu and Sandu. It confirms that our flux-correction strategy is efficient.

Finally, figure 16 shows the results of a time order study. For each local time stepping scheme, the "finest" solution is computed with the time step $\Delta t = 7.10^{-5}$ in the first and last zones. We remind that in the middle zone the grid spacing and the time step are refined by a factor 2, therefore the CFL number is uniform over the mesh ($CFL = 0.007$). In this figure, each mark is associated to the use of the time steps : $2\Delta t$, $4\Delta t$ and $8\Delta t$, for the first and last zones. Once again the slope of each curve is in agreement with the theoretical temporal order of each local time stepping scheme. Our proposed scheme based on RK3 has the lowest L_2 error. Figure 17 shows the results of a space order study, for each local time stepping scheme. This study is performed by employing the same method as in subsection 3.1. The finest solution is computed with $\Delta x = \Delta y = 0.022$. The time step

$\Delta t = 15 \cdot 10^{-4}$ ($CFL = 0.1$) is used in the first and last zones, while the time step $\Delta t = 7,5 \cdot 10^{-4}$ is used in the middle zone. The results show that the four local time stepping schemes have the same spatial order of accuracy. This order of accuracy is slightly superior to 2, the theoretical spatial order of accuracy for the AUSM+P scheme. Again, Figures 16 and 17 demonstrate that our flux correction stage used in schemes "Prop. RK2" and "Prop. RK3" does not alter the time order of accuracy nor the spatial order of accuracy.

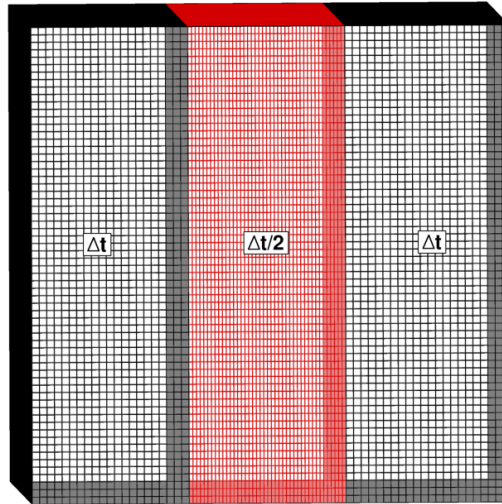


Figure 13: Mesh used for the Taylor-Green Vortex test case

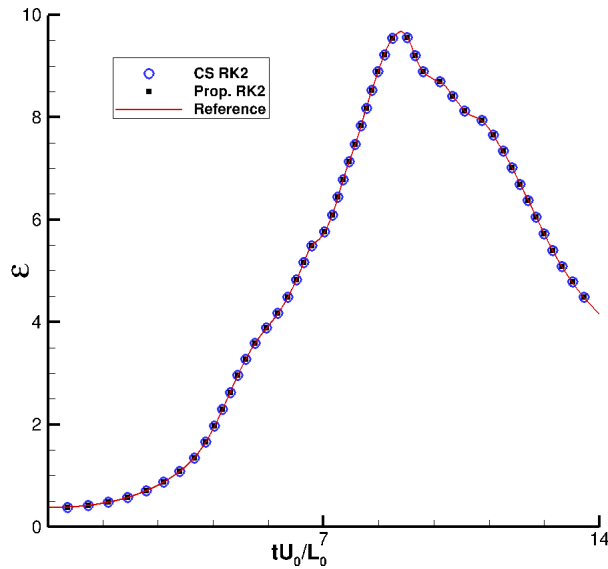


Figure 14: Enstrophy obtained with the schemes "CS RK2" and "Prop. RK2"

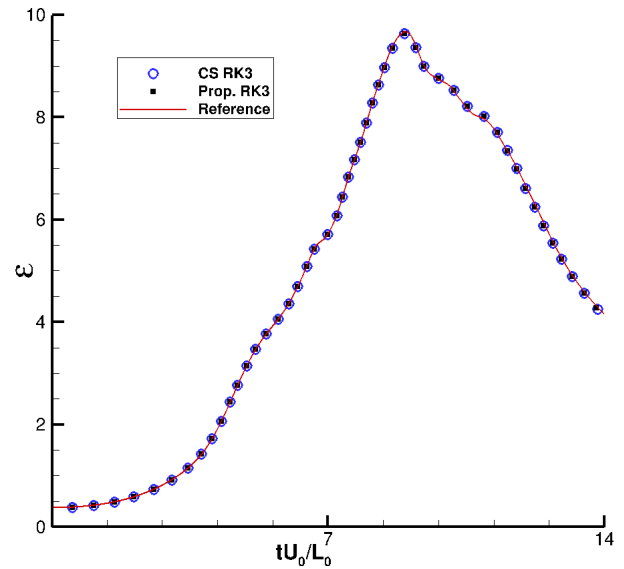


Figure 15: Enstrophy obtained with the schemes "CS RK3" and "Prop. RK3"

	CFLmax(Δt_{max})	Speedup as compared to a RK2 global time stepping scheme	Conservation defect
CS RK2	0.3 ($\Delta t_{max} = 0.0006$)	20%	$1, 8 \cdot 10^{-15}$
Prop. RK2	0.3 ($\Delta t_{max} = 0.0006$)	27%	$2, 2 \cdot 10^{-15}$
CS RK3	0.9 ($\Delta t_{max} = 0.0019$)	57%	$2, 5 \cdot 10^{-15}$
Prop. RK3	0.9 ($\Delta t_{max} = 0.0019$)	59%	$3, 8 \cdot 10^{-15}$

Table 22: Taylor-Green Vortex: comparison of some indicators for the different local time stepping schemes

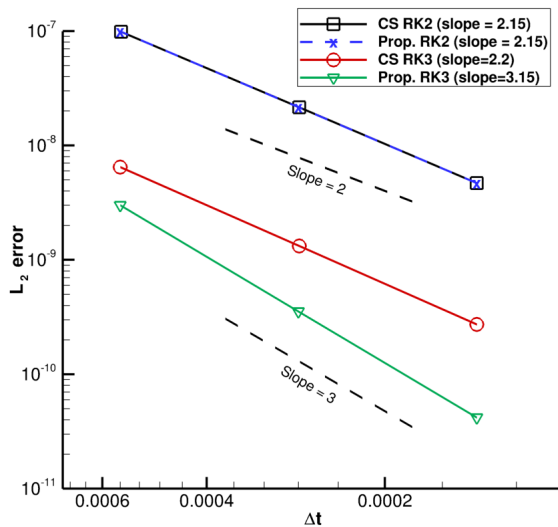


Figure 16: L_2 error as a function of the time step Δt for each local time stepping scheme

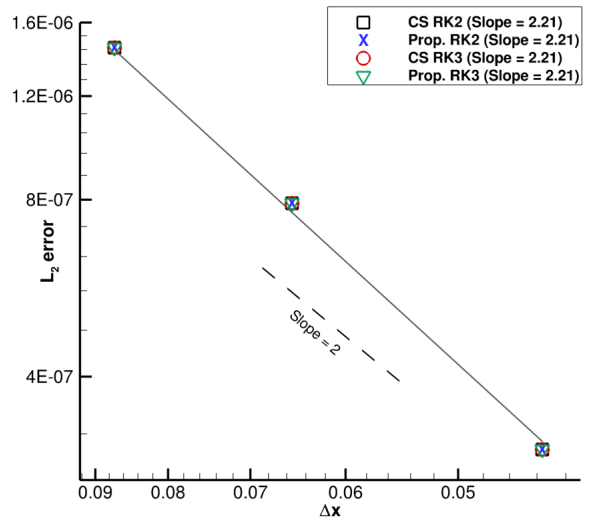


Figure 17: L_2 error as a function of the space step Δx for each local time stepping scheme

3.4 Turbulent flow over an airfoil

This test case is chosen to assess the interest of the local time stepping method for a flow configuration representative of current challenging configurations in the aerospace context. For that purpose a Large Eddy Simulation (LES) over a SD7003 profile is performed. Due to the angle of attack and the Reynolds number, a laminar-turbulent transition occurs in a laminar separation bubble (LSB). It is followed by a turbulent reattachment and a classical turbulent boundary layer subjected to an adverse pressure gradient. As this flow configuration is very sensitive to physical perturbations due to the laminar-turbulent transition process [29], it represents a relevant test case to highlight possible unwanted numerical perturbations introduced by our proposed temporal scheme. This test case is based on 3D Navier-Stokes equations (26). The Mach number, M , and the Reynolds number Re based on the airfoil chord c and the infinite velocity U_∞ are set to $M = 0.1$ and $Re = 60000$, respectively. The angle of attack α is set to $\alpha = 8^\circ$. This Reynolds number is not so challenging for nowadays simulations, but it is sufficient to test the numerical method with a reasonable CPU cost for a comparison between different schemes.

The computational domain is a 3D structured mesh with approximately $90 \cdot 10^6$ cells. The spanwise extent is equal to $0.2c$, while the outer boundaries are located at $100c$ from the airfoil. Characteristic non reflective boundary conditions are used for the farfield, an adiabatic wall is considered for the profile, and periodic boundary conditions are used in the spanwise z direction. To trigger the transition process in a deterministic way, the perturbation proposed in [29] is imposed on the suction side at $x/c = 0.03$ with an amplitude $A_v = 0.001U_\infty$. Figure 18 shows a mesh section in the (\vec{x}, \vec{y}) map with a reduced number of points (one point over 10). An extruded 2D RANS solution is used to initialize the 3D flow solution. No reliable experimental data are available to validate the simulation since the size of the LSB is strongly influenced by the level of the freestream turbulence and also by sidewall effects of the windtunnel. For a computational cost reason, taking into account properly these phenomena is out of the scope of the present simulation. The present wall mesh resolution, $\Delta x^+ \approx 25$, $\Delta y^+ \approx 1$, $\Delta z^+ \approx 8$, is finer than that used in [26] and equivalent to that used in [30, 31], in which grid converged results are presented for the same kind of flow: airfoil with laminar separation bubble and turbulent reattachment. As the numerical method employed in the present paper is the same as the one validated in [26, 32, 30, 31, 33, 34] thanks to windtunnel experiment comparisons, we are confident that our results obtained with the classical RK3 scheme and a global timestep are physically correct for the specific set of Reynolds number and boundary conditions. Therefore the solution obtained with the classical RK3 explicit scheme is considered as a reference to assess the accuracy and efficiency of the proposed temporal scheme.

Our most accurate local time stepping scheme "Prop. RK3" is used to compute this test case. It is compared to a classical explicit scheme (RK3) and one implicit scheme in terms of accuracy (see figures 23 - 28) and CPU cost (see table 24). The implicit scheme is based on the second order accurate Gear scheme [35] for the approximation of the time derivative. A self-adaptive Newton method is used to solve the non linear problem [33]. This method takes advantage of Newton convergence rate heterogeneities by using a spatially varying number of iterations for the Newton process. A user defined convergence criterion ϵ drives the spatially varying number of iterations needed to solve the Newton process. Two implicit simulations are performed with different values for the convergence criterion of the Newton process: $\epsilon = 0.1$, and $\epsilon = 0.06$ (see [33] for more information about this convergence criterion). Note that for this test case, all the schemes are implemented with parallel OpenMP directives and run on 64 threads on one Intel Xeon Phi 7230 node.

The initial 2D RANS solution is used for an automatic mesh partitioning based on the local value of the CFL number. Figure 19 shows a zoom on the airfoil to illustrate the spatial repartition of the different time steps for our local time stepping scheme. Only four time levels have been considered for this validation and the largest time step is such that $\Delta t \times U_\infty/c = 1 \cdot 10^{-5}$. This table 23 shows the relative number of cells for each time step. This table 23 illustrates the interest of using a local time stepping scheme for this case. Indeed, it seems inefficient to restrict the time step for the whole domain to $\frac{\Delta t}{8}$, like in a classical explicit scheme, while only 4% cells require to use this time step.

Time step	Δt	$\frac{\Delta t}{2}$	$\frac{\Delta t}{4}$	$\frac{\Delta t}{8}$
Relative number of cells (%)	61	23	12	4

Table 23: Relative number of cells for each time step

The time step used for the classical RK3 scheme is such that $\Delta t \times U_\infty/c = 1.25 \cdot 10^{-6}$ and the time step of the implicit simulations is such that $\Delta t \times U_\infty/c = 6 \cdot 10^{-5}$ (largest CFL number of

roughly 30). The simulations with the different schemes are performed until the final time t_f such that $t_f \times U_\infty/c = 2$ is reached.

Figure 20 shows the iso-surface of the Q-criterion ($Q = 1.5$) coloured with the fluid velocity magnitude, obtained with our explicit local time stepping scheme at the final time t_f . The Q-criterion is an indicator of turbulent structures, defined by the formula:

$$Q = \frac{1}{2} [(tr(\nabla U))^2 - tr(\nabla U \cdot \nabla U)], \quad (29)$$

where $U = (u_1, u_2, u_3)$ is the fluid velocity vector. This figure shows that the flow is laminar at the leading edge, then a laminar-turbulent transition zone is observed. Finally, the flow is fully turbulent. Figures 21 and 22 show instantaneous isolines of the streamwise velocity component in the plane $z/c=0$. Figure 21 shows that the transition occurs in a laminar separation bubble, characterized by a fluid recirculation (negative streamwise velocity component). The turbulent structures created in the transition zone allow to reattach the fluid to the airfoil (reattachment zone). Figure 22 shows the location of the mid-chord zone, where the flow is fully turbulent. Pressure values are recorded at points 1 (end of the transition zone), 2 (reattachment zone) and 3 (mid-chord zone) to compute Power Spectral Density (PSD) spectrum. Figures 23 - 25 show the mean streamwise velocity profile $\langle u \rangle / u_e$ and two components of the Reynolds stress profiles ($\langle \rho u' u' \rangle / \rho u_e^2$ and $\langle \rho u' v' \rangle / \rho u_e^2$) for each scheme in the transition, reattachment and mid-chord zones. The velocity u_e refers to the streamwise velocity at the edge of the boundary layer. Pressure spectrums at points 1,2 and 3 are plotted in figures 26 - 28.

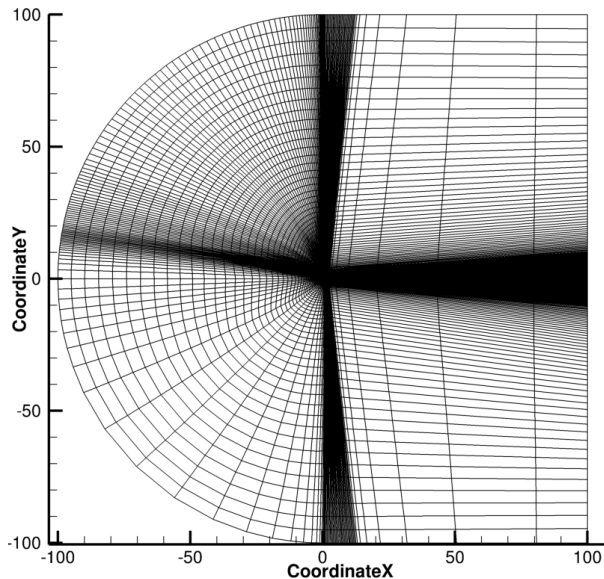


Figure 18: 2D illustration of the structured mesh used for the computation of a flow over a SD7003 Wing. One point over 10 is shown.

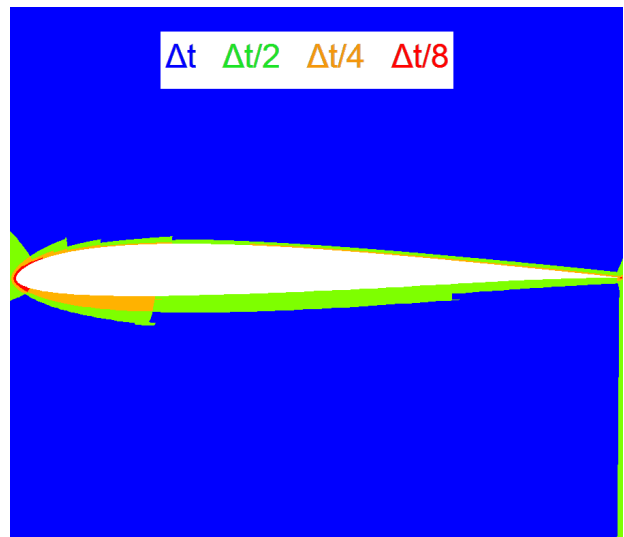


Figure 19: View of the mesh partitioning near the airfoil for our "Prop. RK3" local time stepping scheme.

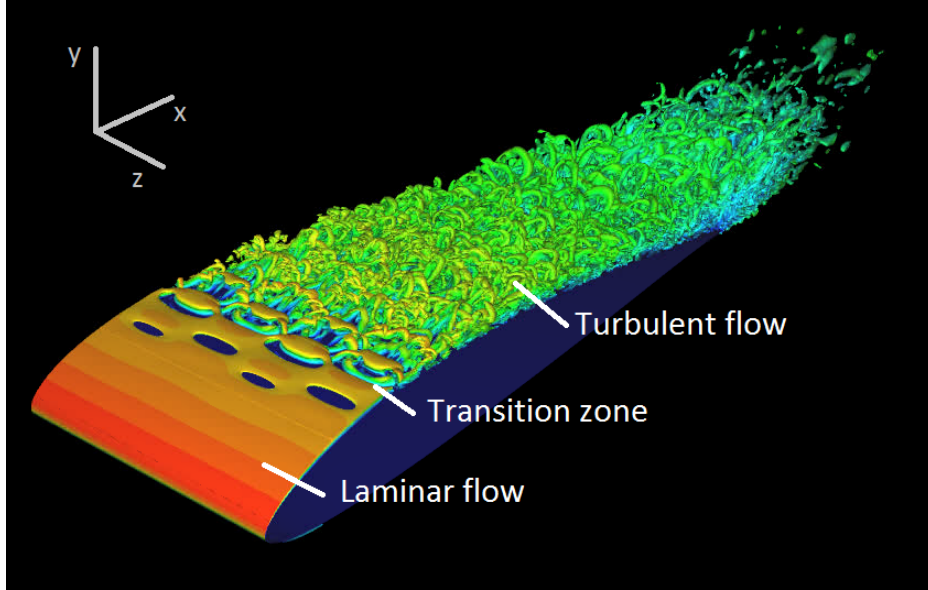


Figure 20: Q-criterion iso-surface ($Q = 1.5$) coloured by the fluid velocity magnitude

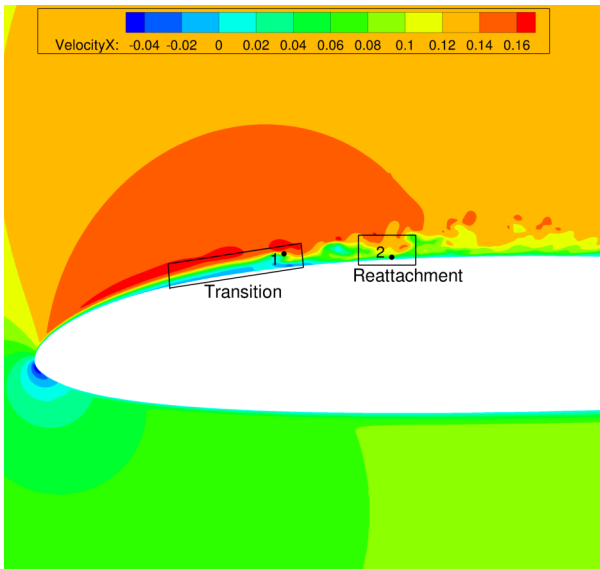


Figure 21: Location of transition and reattachment zones with their pressure recording points

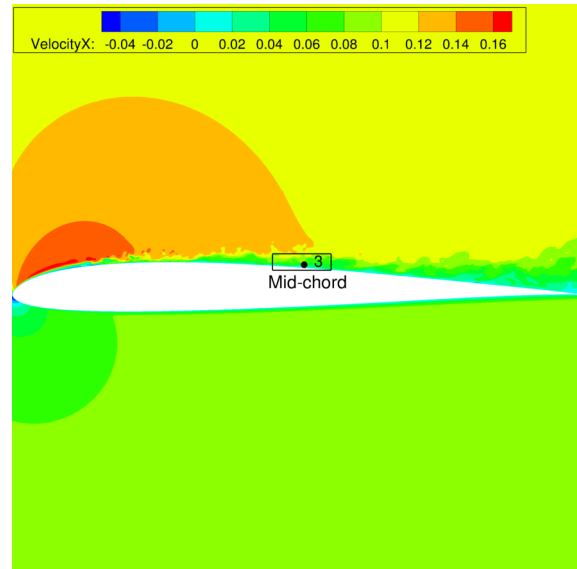


Figure 22: Location of the mid-chord zone with its pressure recording point

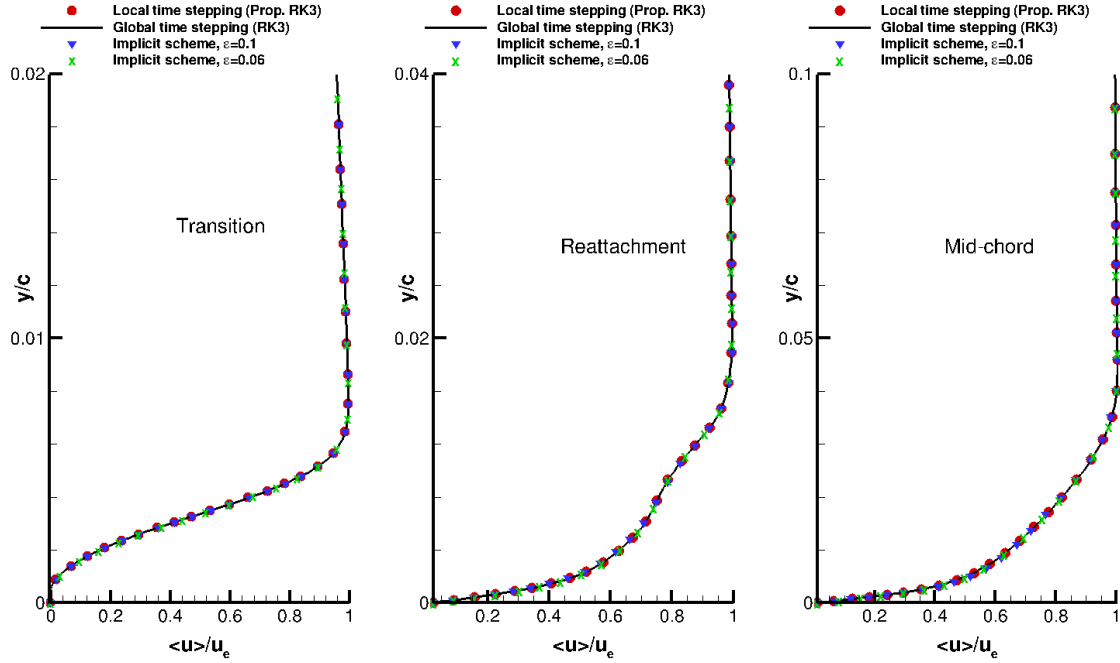


Figure 23: Mean streamwise velocity profiles for each scheme in the transition, reattachment and mid-chord zones.

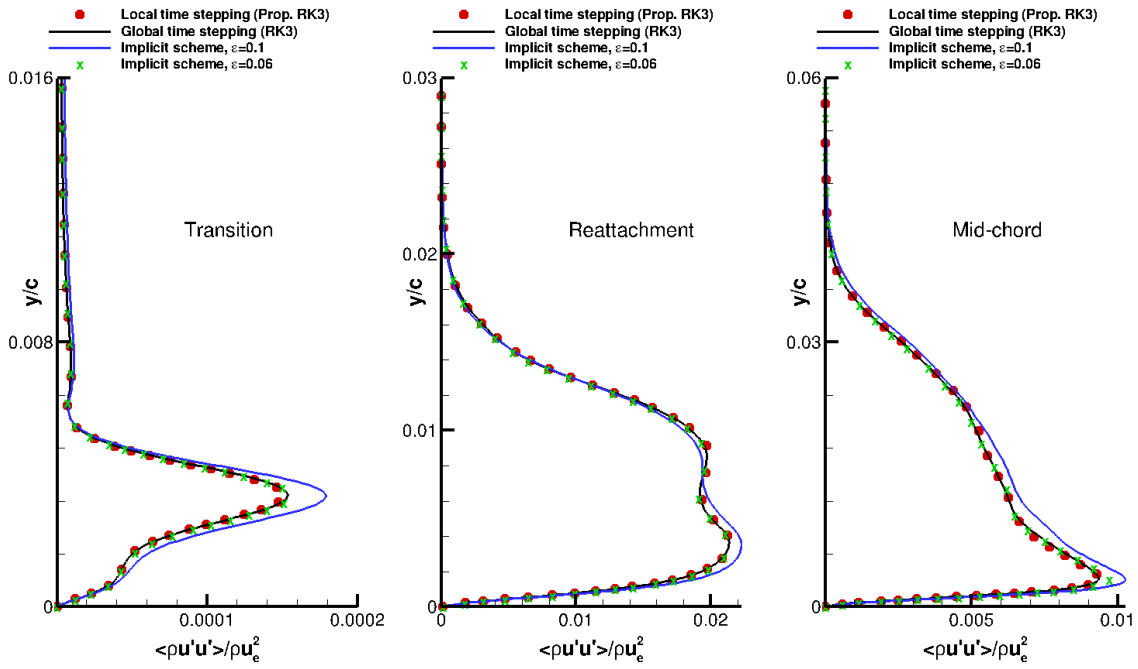


Figure 24: Mean Reynolds stress profiles $\langle \rho u' u' \rangle / \rho u_e^2$ for each scheme in the transition, reattachment and mid-chord zones.

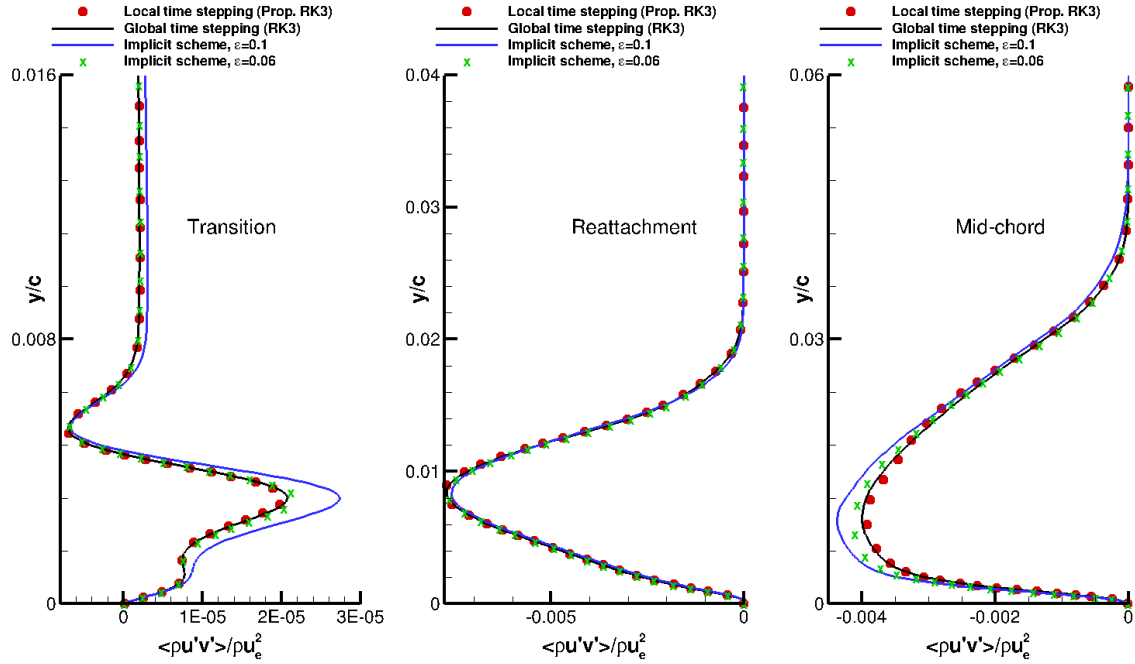


Figure 25: Mean Reynolds stress profiles $\langle \rho u'v' \rangle / \rho u_e^2$ for each scheme in the transition, reattachment and mid-chord zones.

Figure 23 shows a perfect agreement between all schemes for the mean streamwise velocity profiles. The same perfect agreement is observed for the Reynolds stress profiles between the local time stepping scheme, the implicit scheme with $\epsilon = 0.06$ and the reference solution (global time stepping scheme). Small discrepancies between the reference solution and the implicit scheme run with $\epsilon = 0.1$ are present for the Reynolds stresses profiles. It shows that this value of ϵ is too large to prevent an interaction between the flow physics and the numerical error due to the incomplete convergence of the Newton process.

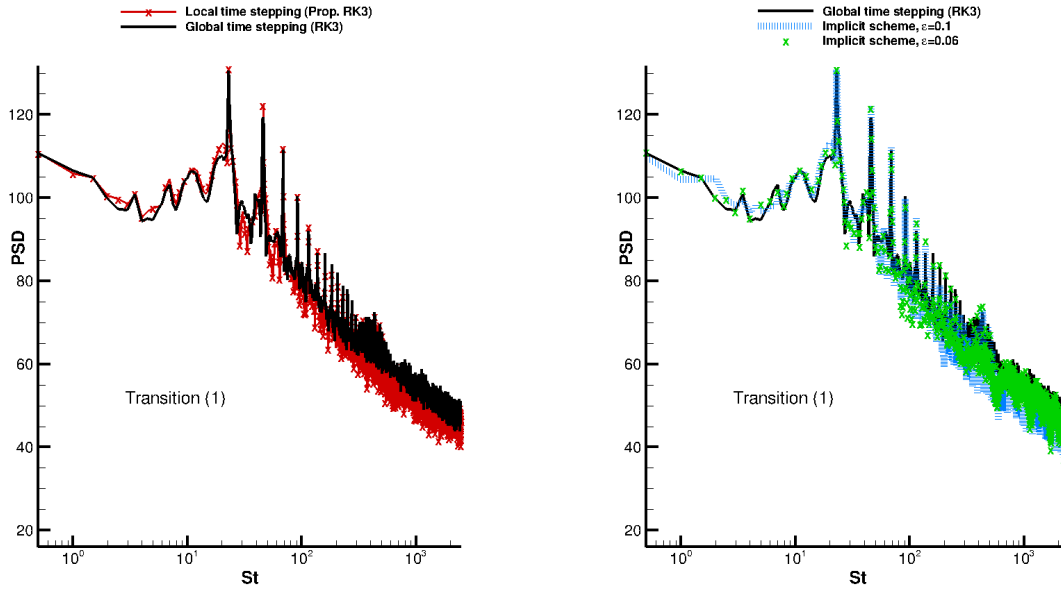


Figure 26: PSD spectrum as a function of the Strouhal number St (based on U_∞) for each scheme in the transition zone (point 1)

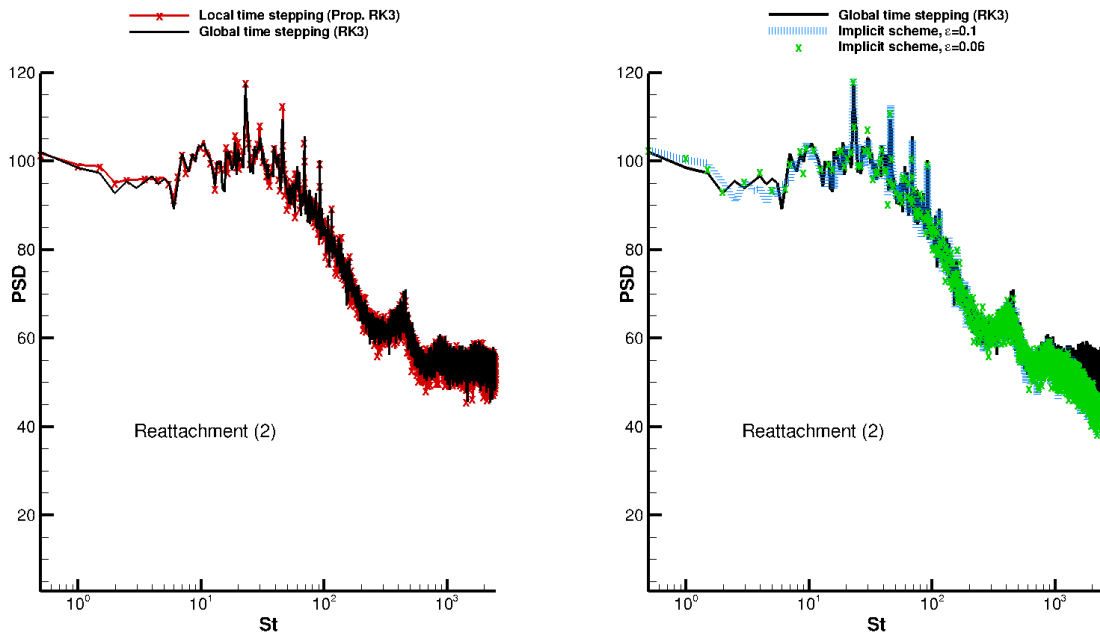


Figure 27: PSD spectrum as a function of the Strouhal number St (based on U_∞) for each scheme in the reattachment zone (point 2)

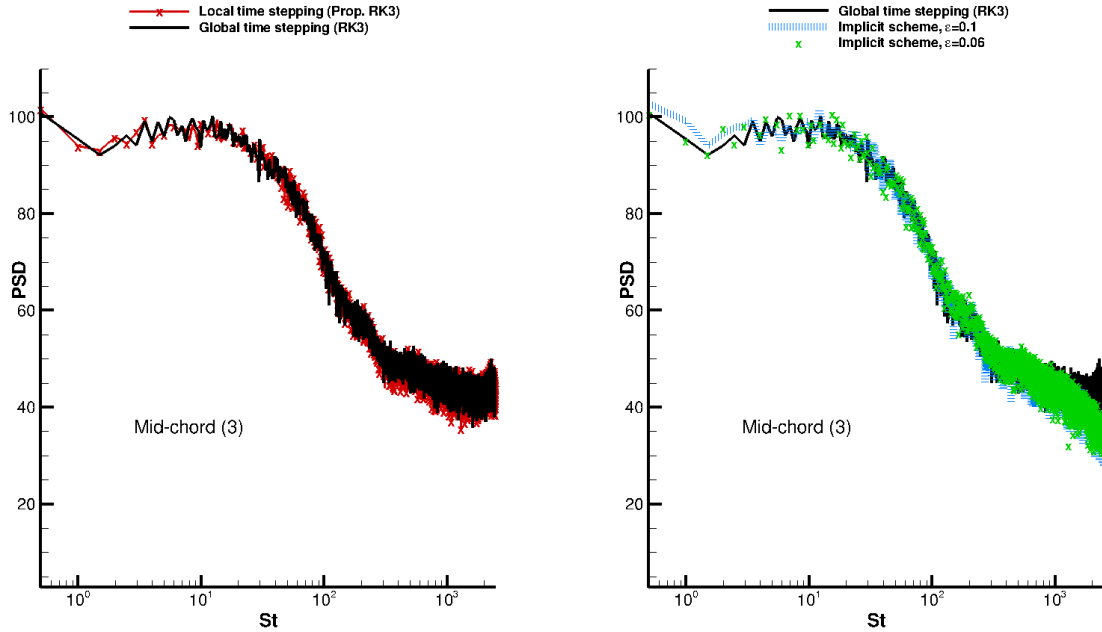


Figure 28: PSD spectrum as a function of the Strouhal number St (based on U_∞) for each scheme in the mid-chord zone (point 3)

No significant influence of the temporal scheme is observed on the pressure spectrums of figures 26 - 28. A slight offset between the implicit schemes spectrums and the explicit schemes spectrums is observed for high Strouhal numbers ($St > 10^3$) at points (2) and (3). The implicit schemes seem slightly less accurate than the explicit schemes to compute high frequencies, certainly because of the use of a larger time step and the dissipative nature of the implicit Gear scheme.

Table 24 shows the CPU time obtained for each scheme and the time reduction as compared to the explicit global time stepping scheme. It appears that our local time stepping scheme "Prop. RK3" provides a significant CPU time reduction (69%), without altering the accuracy of the solution. Same order of CPU time reduction is obtained with the implicit scheme using $\epsilon = 0.1$ (73%), whereas the implicit simulation using $\epsilon = 0.06$ only provides a 47% CPU time reduction.

	Global time stepping (RK3)	Local time stepping (Prop. RK3)	Imp. scheme $\epsilon = 0.1$	Imp. scheme $\epsilon = 0.06$
CPU time (hours)	653	197	175	347
Time reduction (%)	0	69	73	47

Table 24: Temporal scheme influence on the CPU cost of the SD7003 airfoil simulation on a single Intel Xeon Phi 7230 node

This test case demonstrates that our local time stepping scheme "Prop RK3" is able to compute a challenging, industrial-like test case with a very good accuracy and for a competitive CPU time. In order to reach the accuracy of our local time stepping scheme, the implicit scheme convergence criterion ϵ has to be reduced from $\epsilon = 0.1$ to $\epsilon = 0.06$ (see figures 24 - 25). It leads to an important increase in the CPU time, which makes the implicit scheme less competitive than our explicit local time stepping scheme in this field.

4 Conclusion

The diversity of spatial scales in fluid mechanics problems leads to the use of several cell sizes in CFD meshes. It leads to a wide range of numerical stability conditions that must be satisfied by explicit time integration methods. Local time stepping schemes are well suited to these problems because the time step can vary across the mesh to satisfy several local stability conditions. It results in a computational speedup as compared to classical explicit schemes, where the time step for the entire domain is restricted by the most severe stability condition. However, the construction of local time stepping schemes requires a particular attention to ensure high order time accuracy and mass conservation.

Constantinescu and Sandu [5] have developed an interesting strategy which allows to construct mass conservative and second order accurate local time stepping schemes. Their strategy is based on the theory of Partitioned Runge-Kutta (PRK) methods. We also use the theory of PRK methods as well as the computation of consistent additional values at the interfaces to develop two new local time stepping schemes of order two and three. Our schemes are made conservative with an additional flux-correction step. The academic tests show that very similar results are obtained with our local time stepping schemes and the local time stepping schemes of [5]. Moreover, the tests confirm that our third order local time stepping scheme has a lower temporal error than our second order local time stepping scheme and than the local time stepping schemes of [5]. The final test case shows that our third order local time stepping scheme is able to compute complex, industrial-like test cases with a very good accuracy and for a CPU time close to the one obtained with our implicit scheme. Our strategy seems to be a good approach to construct efficient, accurate and conservative local time stepping schemes.

We believe that our method will be more efficient than implicit schemes on a larger number of CPU cores, thanks to MPI or hybrid MPI/OpenMP parallelization. This requires a specific work on load balancing for our local time stepping scheme, like in [36] and [37]. This is a work in progress that will be the subject of a future publication.

Acknowledgment

This work has been supported by ONERA (Office Nationale d'Etudes et de Recherches Aéronautiques) and DGA (Direction Générale de l'Armement)

References

- [1] Haecheon Choi and Parviz Moin. Effects of the computational time step on numerical solutions of turbulent flow. *Journal of Computational Physics*, 113(1):1 – 4, 1994.
- [2] T. Unfer. An asynchronous framework for the simulation of the plasma/flow interaction. *Journal of Computational Physics*, 236:229 – 246, 2013.
- [3] A. Toumi, G. Dufour, R. Perrussel, and T. Unfer. Asynchronous numerical scheme for modeling hyperbolic systems. *Comptes Rendus Mathematique*, 353(9):843 – 847, 2015.
- [4] V.A. Semiletov and S.A. Karabasov. Cabaret scheme for computational aero acoustics: Extension to asynchronous time stepping and 3d flow modelling. *International Journal of Aeroacoustics*, 13(3-4):321–336, 2014.
- [5] E. M. Constantinescu and A. Sandu. Multirate timestepping methods for hyperbolic conservation laws. *Journal of Scientific Computing*, 33(3):239–278, Dec 2007.
- [6] S. Osher and R. Sanders. Numerical approximations to nonlinear conservation laws with locally varying time and space grids. *Mathematics of computation*, 41(164), October 1983.
- [7] H. Tang and G. Warnecke. A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and space grids. *Journal of Scientific Computing*, 24(2):121–140, 2005.
- [8] M. Schlegel, O. Knoth, M. Arnold, and R. Wolke. Multirate runge–kutta schemes for advection equations. *Journal of Computational and Applied Mathematics*, 226(2):345 – 357, 2009. Special Issue: Large scale scientific computations.
- [9] L. Liu, X. Li, and F. Q. Hu. Nonuniform time-step runge–kutta discontinuous galerkin method for computational aeroacoustics. *Journal of Computational Physics*, 229(19):6874 – 6897, 2010.
- [10] L. Liu, X. Li, and F. Q. Hu. Nonuniform-time-step explicit runge–kutta scheme for high-order finite difference method. *Computers and Fluids*, 105:166 – 178, 2014.
- [11] B. Seny, J. Lambrechts, R. Comblen, V. Legat, and J.-F. Remacle. Multirate time stepping for accelerating explicit discontinuous galerkin computations with application to geophysical flows. *International Journal For Numerical Methods In Fluids*, 71(41):41–64, 2007.
- [12] V. Savcenco, W. Hundsdorfer, and J. G. Verwer. A multirate time stepping strategy for stiff ordinary differential equations. *BIT numerical mathematics*, 2006.
- [13] C. Dawson and R. Kirby. High resolution schemes for conservation laws with locally varying time steps. *SIAM J. Sci. Comput.*, 22(6):2256–2281, June 2000.
- [14] A. Kvaerno and P. Rentrop. Low order multirate runge-kutta methods in electric circuit simulation. *Preprint No. 99/1, IWRMM, Universität Karlsruhe (TH)*, 1999.
- [15] Jean M. Sexton and D. Reynolds. Relaxed multirate infinitesimal step methods. *Preprint submitted to Journal of Computational and Applied Mathematics*, 08 2018.
- [16] A. Sandu. A class of multirate infinitesimal gark methods. *Technical report of the computational science laboratory (Virginia Tech)*, 08 2019.

- [17] W. Hundsdorfer, A. Mozartova, and V. Savcenko. *Analysis of explicit multirate and partitioned Runge-Kutta schemes for conservation laws*. CWI report. MAS-E. Centrum voor Wiskunde en Informatica, 2007.
- [18] O. Knoth and R. Wolke. Implicit-explicit runge-kutta methods for computing atmospheric reactive flows. *Applied Numerical Mathematics*, 28(2):327 – 341, 1998.
- [19] J. Wensch, K. Oswald, and G. Alexander. Multirate infinitesimal step methods for atmospheric flow simulation. *BIT Numerical Mathematics*, 49(2):449–473, Jun 2009.
- [20] E. Hairer, G. Wanner, and S.P. Norsett. *Solving ordinary differential equations I : Nonstiff Problems*. Springer, Berlin, 1993.
- [21] J.H. Williamson. Low-storage runge-kutta schemes. *Journal of Computational Physics*, 35(1):48 – 56, 1980.
- [22] P.S. Lowery and W. C. Reynolds. Numerical simulation of a spatially-developping, forced, plane mixing layer. *Report of the Stanford University, TF26*, 1986.
- [23] S. Gottlieb and C.-W. Shu. Total variation diminishing runge-kutta schemes. *Math. Comput.*, 67(221):73–85, January 1998.
- [24] P.L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357 – 372, 1981.
- [25] P. L. Roe. Characteristic-based schemes for the euler equations. *Annual Review of Fluid Mechanics*, 18(1):337–365, 1986.
- [26] I. Mary and P. Sagaut. Large eddy simulation of flow around an airfoil near stall. *AIAA Journal*, 40(6):1139–1145, 2002.
- [27] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97 – 127, 1999.
- [28] J. R. DeBonis. Solutions of the taylor-green vortex problem using high-resolution explicit finite difference methods. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.
- [29] O. Marxen and Dan S. Henningson. The effect of small-amplitude convective disturbances on the size and bursting of a laminar separation bubble. *Journal of Fluid Mechanics*, 671:1–33, 2011.
- [30] N. Alferez, I. Mary, and E. Lamballais. Study of stall development around an airfoil by means of high fidelity large eddy simulation. *Flow, Turbulence and Combustion*, 91(3):623–641, 2013.
- [31] C. Laurent, I. Mary, V. Gleize, A. Lerat, and D. Arnal. Dns database of a transitional separation bubble on a flat plate and application to rans modeling validation. *Computers and Fluids*, 61:21–30, 2012.
- [32] L. Larchevêque, P. Sagaut, I. Mary, O. Labbé, and P. Comte. Large-eddy simulation of a compressible flow past a deep cavity. *Physics of Fluids*, 15(1):193–210, 2003.
- [33] F. Daude, I. Mary, and P. Comte. Self-adaptive newton-based iteration strategy for the les of turbulent multi-scale flows. *Computers and Fluids*, 100:278 – 290, 2014.

- [34] J. Dandois, I. Mary, and V. Brion. Large-eddy simulation of laminar transonic buffet. *Journal of Fluid Mechanics*, 850:156–178, 2018.
- [35] C. W. Gear. *Numerical initial value problems in ordinary differential equations*. Prentice Hall PTR, Upper Saddle River, New Jersey, USA, 1971.
- [36] B. Seny, J. Lambrechts, T. Toulorge, V. Legat, and J.-F. Remacle. An efficient parallel implementation of explicit multirate runge–kutta schemes for discontinuous galerkin computations. *Journal of Computational Physics*, 256:135 – 160, 2014.
- [37] M. Rietmann, D. Peter, O. Schenk, B. Uçar, and M.J. Grote. Load-balanced local time stepping for large scale wave propagation. *2015 IEEE International Parallel and Distributed Processing Symposium, IEEE CPS*, 2015.