Zonal time-stepping method for multi-scale unsteady simulations within an industrial CFD code

Clement Le Touze

ZONAL TIME-STEPPING METHOD FOR MULTI-SCALE UNSTEADY SIMULATIONS WITHIN AN INDUSTRIAL CFD CODE

CLEMENT LE TOUZE[1]

[1] DMPE, ONERA, Université Paris Saclay, F-91123 Palaiseau - France,
clement.le_touze@onera.fr

Key words: Time Integration, Zonal Time-stepping, Multi-scale Unsteady Simulations, Computational Efficiency

Summary. A conservative zonal time-stepping method is developed within ONERA's Cedre platform to reduce the computational cost of unsteady simulations, with a special attention to make the method as flexible as possible in terms of time steps ratio and evolution. The method is tested on different cases to assess its efficiency and robustness, and perspectives of improvement are discussed.

1 INTRODUCTION

The numerical simulations conducted in computational fluid dynamics (CFD) often exhibit strong multi-scale features. This can be due to the use of unstructured grids with highly heterogeneous refinement, and/or to a wide range of physical characteristic times or lengths. This is especially true in the field of energetics, where it is common to deal with compressible multiphase flows, chemical reactions and turbulence. As a result, significant spatial variations in the CFL (Courant-Friedrichs-Lewy) number can be encountered. The stability of temporal integration methods, particularly explicit ones, mandates a maximum CFL number not to be exceeded, thereby imposing a maximum stability time step, theoretically different for each mesh cell. However in practice, the simulation time step must be uniform for unsteady calculations to preserve conservation and temporal consistency. This uniform time step is then dictated by the stability of the most restrictive cell. Hence, there is a clear interest in developing conservative and consistent local time-stepping methods to relax the constraint of a uniform time step for unsteady simulations. This can significantly optimize computation times, with a greater benefit for applications in which the distribution of CFL values is broad.

In this work we present a zonal time-stepping method applicable to explicit Runge-Kutta methods, implemented within Cedre [1], a multi-physics platform for industrial applications in energetics, using general unstructured meshes in the finite volume framework. The method follows the principles of some methods proposed in the literature [2–6]. However, a specific effort has been made to allow the time step in each zone to evolve over time to match the CFL condition as closely as possible throughout the computation, keeping some flexibility in the timing of the different domains. Conservation is ensured through the application of corrective terms. After describing the method, it is illustrated on representative cases. Its benefits in term of computational efficiency are evaluated, as well as the impact on both accuracy and robustness. Finally, some improvement perspectives are discussed.

2 ZONAL TIME-STEPPING METHOD

2.1 Cell-centered finite volume framework and explicit SSPRK methods

In the framework of the cell-centered finite volume method, the computational domain Ω is split into N_c general polyhedral elements K_i and polygonal faces S_{ij} of barycenters $\boldsymbol{B}_i$ and $\boldsymbol{M}_{ij}$ (see figure 1). We also define several neighborhoods for a given cell K_i, namely $\mathcal{V}_1(i) = \{K_j | K_i \cap K_j = S_{ij}\}$ the set of face-neighbor cells of K_i, $\mathcal{V}_2(i)$ the reconstruction stencil of cell K_i, e.g the vertex-neighbor cells in the multislope MUSCL method [7], and $\mathcal{V}_3(i)$ the set of all neighbor cells potentially involved in the computation of the right-hand-side (RHS) during an iteration of the Runge-Kutta (RK) method (which depends on the reconstruction stencil $\mathcal{V}_2(i)$ and on the number of steps s of the RK method).

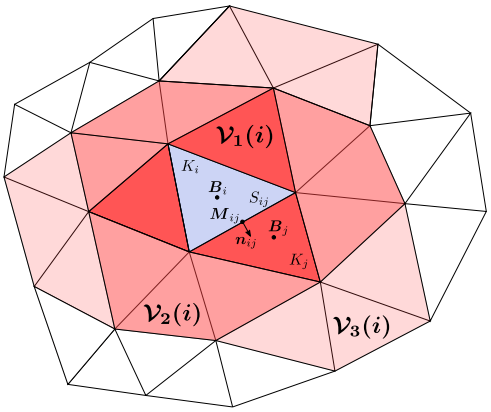


Figure 1: Geometric notations for the cell-centered finite volume method.

Now if we consider any hyperbolic system of conservation laws written as

$$\partial_t \boldsymbol{Q}(t,\boldsymbol{x}) + \nabla \cdot \boldsymbol{f}[\boldsymbol{Q}(t,\boldsymbol{x})] = 0,\ \boldsymbol{x} \in \Omega,\ t > 0, \tag{1}$$

where $\boldsymbol{Q}$ is the vector of the conserved quantities, then the finite volume discretization of this system reads:

$$\forall K_i \in \Omega,\ \frac{d\boldsymbol{Q}_i(t)}{dt} = -\sum_{j \in \mathcal{V}_1(i)} \frac{|S_{ij}|}{|K_i|} \boldsymbol{\phi}_{ij}\left(\boldsymbol{Q}_{ij}(t), \boldsymbol{Q}_{ji}(t), \boldsymbol{n}_{ij}\right). \tag{2}$$

Here $\boldsymbol{\phi}_{ij}$ is the numerical flux function depending on some MUSCL reconstructions $\boldsymbol{Q}_{ij}(t) = f_{MUSCL}\left(\overline{\boldsymbol{\mathcal{Q}}_i}(t)\right)$, with $\overline{\boldsymbol{\mathcal{Q}}_i}(t) = \bigcup_{k \in \mathcal{V}_2(i)} \{\boldsymbol{Q}_k(t)\}$ the set of all cell states involved in the reconstruction stencil. The finite volume scheme can be recast in a more compact form as follows:

$$\forall K_i \in \Omega,\quad \frac{d\boldsymbol{Q}_i(t)}{dt} = \boldsymbol{F}\left(\boldsymbol{\mathcal{Q}}_i(t)\right), \tag{3}$$

where $\boldsymbol{F}\left(\boldsymbol{\mathcal{Q}}_i(t)\right) = -\sum_{j \in \mathcal{V}_1(i)} \frac{|S_{ij}|}{|K_i|} \boldsymbol{\phi}_{ij}\left(\boldsymbol{Q}_{ij}(t), \boldsymbol{Q}_{ji}(t), \boldsymbol{n}_{ij}\right)$ is the right-hand-side (RHS) of the ordinary differential equation (ODE), and $\boldsymbol{\mathcal{Q}}_i(t) = \bigcup_{k \in \mathcal{V}_3(i)} \{\boldsymbol{Q}_k(t)\}$ is the set of all cell states

involved in the RHS. Then the time discretization of the ODE is performed using the following Runge-Kutta formalism (written here in the Butcher form):

$$\left\{\begin{array}{lcl} \boldsymbol{Q}_i^{n+1} & = & \boldsymbol{Q}_i^n + (\Delta t)_i^n \sum_{\alpha=1}^{s} b_\alpha \boldsymbol{F}\left(\boldsymbol{\mathcal{Q}}_i^{n,\alpha}\right) \\ \boldsymbol{Q}_i^{n,\alpha} & = & \boldsymbol{Q}_i^n + (\Delta t)_i^n \sum_{\beta=1}^{s} a_{\alpha\beta} \boldsymbol{F}\left(\boldsymbol{\mathcal{Q}}_i^{n,\beta}\right), \quad 1 \leq \alpha \leq s \end{array}\right. , \tag{4}$$

with $a_{\alpha\beta}$, b_α, the coefficients of the Butcher table and s the number of steps of the RK method. We are more specifically interested in the Strong Stability Preserving Runge-Kutta methods (SSPRK), which are a subset of the RK methods that can be written as a sequence of explicit Euler steps (Shu-Osher form) [8, 9], thereby ensuring some good stability properties for the overall scheme.

2.2 Specifications for a zonal time step method in Cedre

We want a method that is **conservative** and which preserves as much as possible the space and time **accuracy** of the scheme, compatible with **any explicit SSPRK method** and compatible with the code **architecture** (not too intrusive as Cedre is a huge legacy code). This means that we exclude asynchronous methods [10] that employ a local time step at the cell level, because they basically require to write a new code from scratch. Instead we skip to a zonal approach, i.e we use the same time step for all cells of a given zone [2–6]. To do so we make use of an already existing feature in Cedre which enables the user to split the computational domain in zones in which different physical models or numerical methods can be used.

Another important specification is that we want the method to exhibit some **flexibility** when dealing with the zones. In particular, we do not want to prescribe the number of zones or the time steps ratio between the zones, and we want to allow the time step of each zone to evolve during the simulation, typically in order to follow a CFL criterion. This has led to the development of the evolutive zonal time step method in Cedre, which we will now describe.

2.3 Description of the evolutive zonal time step method

First of all, let us a consider a first simple approach in which the time steps in each zone would be prescribed, as illustrated in figure 2. In this method, the ratio between the time steps of any two neighboring zones Ω_i and Ω_j must be an integer:

$$\forall \left(\Omega_i, \Omega_j\right), \quad \partial\Omega_i \cap \partial\Omega_j \neq \emptyset, \quad \frac{\max\left[(\Delta t)_i, (\Delta t)_j\right]}{\min\left[(\Delta t)_i, (\Delta t)_j\right]} \in \mathbb{N}^\star. \tag{5}$$

This enables synchronization points between both zones, during which exchanges and conservation corrections are carried out at the boundaries, with a minimal period of time equal to

$$\tau_{ij} = \max\left[(\Delta t)_i, (\Delta t)_j\right]. \tag{6}$$

Now the principle ot the method allowing an evolutive zonal time step is slightly more complicated. It is based on the following procedure (see figure 3):

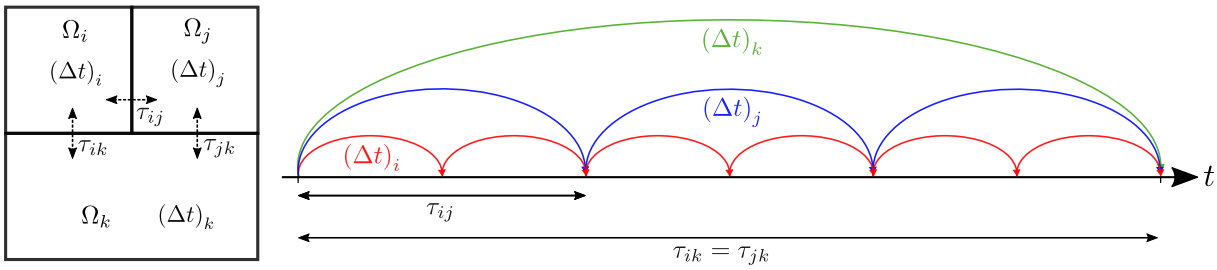


Figure 2: Principle of the prescribed zonal time step method.

1. Compute the **maximum time step allowed** for each zone Ω_i (e.g. based on a CFL criterion, l is the index of the cells in zone Ω_i):

$$\overline{(\Delta t)}_i = \min_{l \in \Omega_i} \overline{(\Delta t)}_l. \tag{7}$$

2. Compute an arbitrary **reference time step** $(\Delta t)_0$ and the set $\mathcal{T}$ of all possible time steps:

$$\mathcal{T} = \left\{ (\Delta t)_p = 2^p \, (\Delta t)_0 \ , \quad p \in \mathbb{Z} \right\}, \tag{8}$$

   where p is the number of the **time level**.

3. Compute the **effective time step** for each zone Ω_i, which is chosen as the highest of these quantified time steps that is inferior to the maximum value constrained by the CFL condition:

$$(\Delta t)_i = \left\{ 2^{p_i} \, (\Delta t)_0 \ | \ 2^{p_i} \leq \frac{\overline{(\Delta t)}_i}{(\Delta t)_0} < 2^{p_i+1} \right\}, \ p_i = \mathrm{E}\left[\ln\left(\frac{\overline{(\Delta t)}_i}{(\Delta t)_0} \right) / \ln 2 \right]. \tag{9}$$

4. Each zone is time-advanced with its effective time step $(\Delta t)_i$ using any SSPRK explicit method. An important feature is that $(\Delta t)_i$ can be **reduced anytime** if required by the evolution of $\overline{(\Delta t)}_i$ (the zone "jumps" to a lower time level, see figure 3). On the other hand it can be **increased only** during a general synchronization point, otherwise synchronization points could be missed.

Like in the prescribed time steps approach, synchronization between neighboring zones has to be performed with a minimal period of time $\tau_{ij} = \max\left[(\Delta t)_i, (\Delta t)_j \right]$. This ensures that exchange of information and conservation corrections at the boundaries are employed as frequently as possible. On the HPC side, each zone is split in an equal number of subdomains, which are then dispatched on all MPI processes. This way the **load balancing** is guaranteed throughout the computation even when the time levels of each zone evolve.

2.4 Conservation error and correction

Let us consider two cells K_i and K_j that belong to adjacent zones Ω_i and Ω_j and that evolve between times t^0 and t^∞ with their own sequence of time steps as depicted in figure 4. Then we

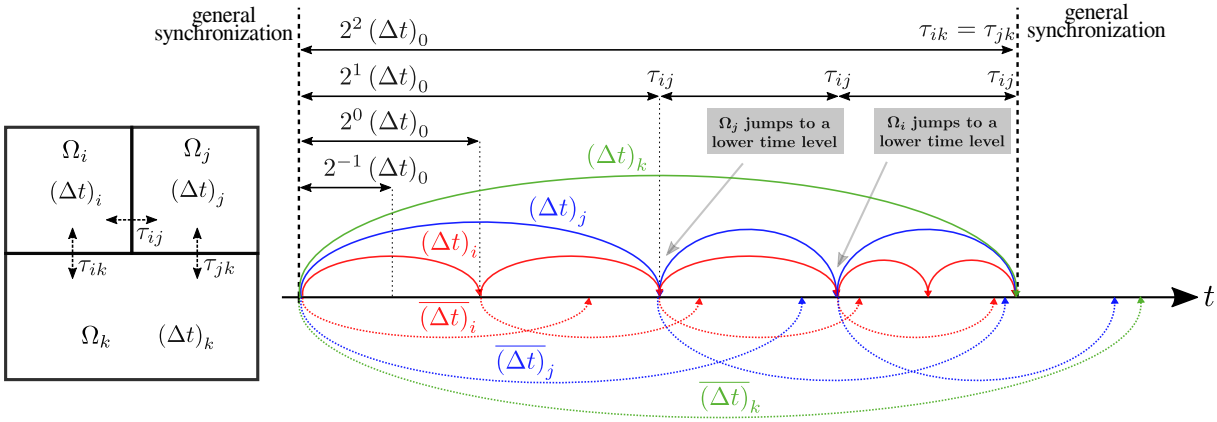


Figure 3: Principle of the evolutive zonal time step method.

can write the evolution of the state $\boldsymbol{Q}_i$ between the initial and final times as:

$$\left\{\begin{array}{rcl} \boldsymbol{Q}_i^{\infty} & = & \boldsymbol{Q}_i^0 + \displaystyle\sum_{n_i=0}^{N_i-1}\left[(\Delta t)^{n_i}\sum_{\alpha_i=1}^{s_i} b_{\alpha_i}\boldsymbol{F}\left(\boldsymbol{\mathcal{Q}}_i^{n_i,\alpha_i}\right)\right] \\ \boldsymbol{Q}_i^{n_i,\alpha_i} & = & \boldsymbol{Q}_i^{n_i} + (\Delta t)^{n_i}\displaystyle\sum_{\beta_i=1}^{\alpha_i-1} a_{\alpha_i\beta_i}\boldsymbol{F}\left(\boldsymbol{\mathcal{Q}}_i^{n_i,\beta_i}\right) \end{array}\right. , \tag{10}$$

and a similar expression can be written for the state $\boldsymbol{Q}_j$. After some manipulations to extract from the RHS $\boldsymbol{F}$ the sum of the fluxes across the faces of K_i and K_j, we get the time integral of the fluxes seen from each side of their common face S_{ij}, namely

$$\mathcal{F}_i = |S_{ij}|\sum_{n_i=0}^{N_i-1}\left[(\Delta t)^{n_i}\sum_{\alpha_i=1}^{s_i} b_{\alpha_i}\phi_{ij}\left(\boldsymbol{Q}_{ij}^{n_i,\alpha_i}, \boldsymbol{Q}_{ji}^{n_i,\alpha_i}, \boldsymbol{n}_{ij}\right)\right] \tag{11}$$

from the side of cell K_i, and

$$\mathcal{F}_j = -|S_{ij}|\sum_{n_j=0}^{N_j-1}\left[(\Delta t)^{n_j}\sum_{\alpha_j=1}^{s_j} b_{\alpha_j}\phi_{ij}\left(\boldsymbol{Q}_{ij}^{n_j,\alpha_j}, \boldsymbol{Q}_{ji}^{n_j,\alpha_j}, \boldsymbol{n}_{ij}\right)\right] \tag{12}$$

from the side of cell K_j. For the scheme to be conservative the sum of these two integrals should be zero. When this is not the case we have a conservation error $\mathcal{E}_{ij}$ that we need to compensate and which reads:

$$\mathcal{E}_{ij} = \mathcal{F}_i + \mathcal{F}_j. \tag{13}$$

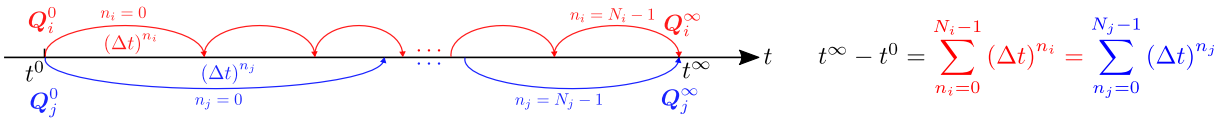


Figure 4: Two adjacent cells evolving between times t^0 and t^{∞} with their own sequence of time steps.

To do so, we eventually compute corrected states $\overline{\boldsymbol{Q}_i^\infty}$ and $\overline{\boldsymbol{Q}_j^\infty}$ as:

$$\left\{ \begin{array}{rcl} \overline{\boldsymbol{Q}_i^\infty} & = & \boldsymbol{Q}_i^\infty - \gamma_i \dfrac{\mathcal{E}_{ij}}{|K_i|} \\ \overline{\boldsymbol{Q}_j^\infty} & = & \boldsymbol{Q}_j^\infty - \gamma_j \dfrac{\mathcal{E}_{ij}}{|K_j|} \end{array} \right. , \quad \gamma_i \geq 0, \quad \gamma_j \geq 0, \quad \gamma_i + \gamma_j = 1, \tag{14}$$

where the weighting coefficients γ_i and γ_j can be based for example on the cell volumes:

$$\gamma_i = \frac{|K_i|}{|K_i| + |K_j|}, \quad \gamma_j = \frac{|K_j|}{|K_i| + |K_j|}. \tag{15}$$

Note that the conservation error increases with the synchronization period between the zones, this is why it is important to have the lowest synchronization period (exchange as frequently as possible). The error is also higher when strongly non-linear solutions come across the boundaries (shocks, liquid-gas interface in two-phase flows...).

2.5 Potential speedups on typical unsteady CEDRE simulations

Now the interesting question is: what are the potential speedups that we can expect on typical unsteady simulations carried out with CEDRE ? To answer this question, we can compute for example the expected speedup S with a fully local time step method:

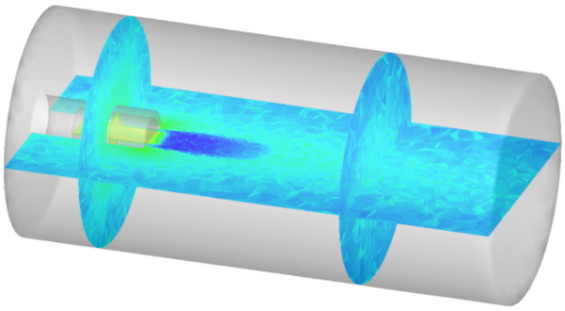
$$S = \frac{\text{global time step cost}}{\text{local time step cost}} = \frac{N_c / \min\limits_{K_i \in \Omega} \overline{(\Delta t)}_i}{\sum_{i=1}^{N_c} 1/\overline{(\Delta t)}_i}. \tag{16}$$

This speedup is therefore defined as the ratio between the cost of a global time step method and a local time step method, when each cell has itw own optimal time step. This gives us an upper bound value for our zonal approach, for which the effective speedup will be lower due to the zonal partitionning, the quantification in 2^p and the cost of the algorithm itself. Speedup estimates are provided for two typical cases in figure 5, in particular for the case of the liquid jet in cross flow on which the zonal time step method will be applied in section 3.1. This shows that it can lead to potentially very high speedups. Actually, it all depends on the heterogeneity of the CFL field for each case.

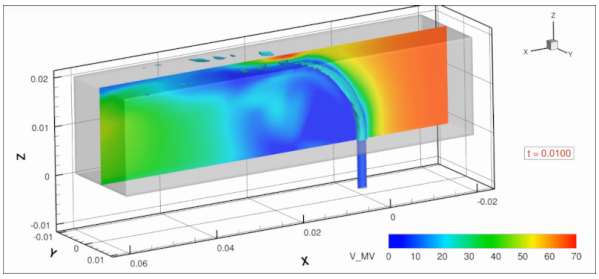
3 APPLICATION CASES

3.1 Liquid jet in cross flow

We now apply the zonal time-stepping method to the case of the liquid jet in cross flow mentioned in section 2.5. The multi-species compressible Navier-Stokes equations are solved to simulate the impact of a liquid jet by a cross flow of air. The water is injected from the small round pipe visible in figure 6 with a velocity $V_{water} = 8$ m/s, while the air is injected from the right boundary with a velocity $V_{air} = 66$ m/s. We employ the explicit SSPRK(6,2) method (see [9] for details) with a maximum CFL value of 2. And we use a 3D structured mesh with 5.6M cells, which is also refined near the injector wall to capture boundary layer effects ($y^+ \approx 5$). The domain is split into four different zones in order to run the evolutive zonal time stepping method, and the computation is performed on 144 MPI processes.



(a) Liquid-gas coaxial injector: $S = 17.5$.



(b) Liquid jet in cross flow: $S = 7.5$.

Figure 5: Speedup estimates for typical unsteady CEDRE simulations.

Figure 7 shows a snapshot of the solution at time $t = 10$ ms, obtained with both a global time step method and the evolutive zonal time step method. No significant difference can be seen between both simulations, however the one using the zonal time step is 6.4 times faster than the reference simulation. As expected, this is slightly less than the optimal speedup estimate of $S = 7.5$ (see section 2.5), but this is still a very interesting speedup.

3.2 Inviscid strong vortex - shock wave interaction

We now consider the case of the inviscid strong vortex - shock wave interaction, which is a classical test case used in the HiOCFD workshops [11]. In this case, performed with the 2D Euler equations for an ideal gas, a strong vortex is travelling through a stationnary shock, which entails a complex structure of acoustic and shock waves. Figure 8 shows the setup of this case, as well as the reference solution obtained at the final time $t = 0.7$ s on a very fine 16-million cell cartesian grid and with a global time step method. The objective here is to assess the impact of the zonal time step approach on the **accuracy** and **robustness** of the computation. To do so, we build a relatively coarse mesh with around 300k cells that we partition on purpose in 450 temporal zones, and with a very heterogeneous mesh refinement (see figure 9). This way

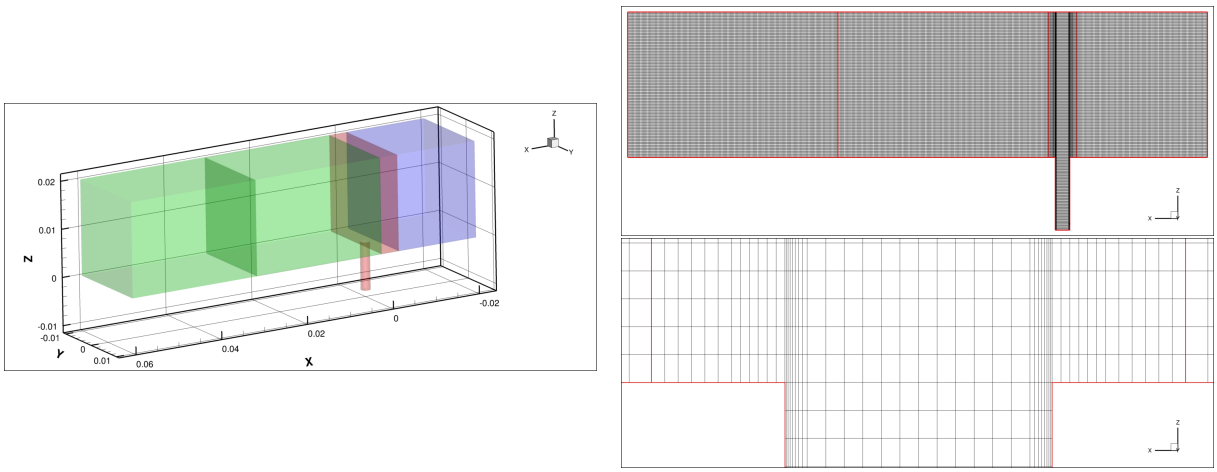
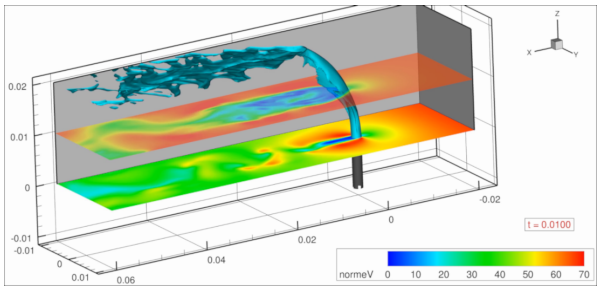
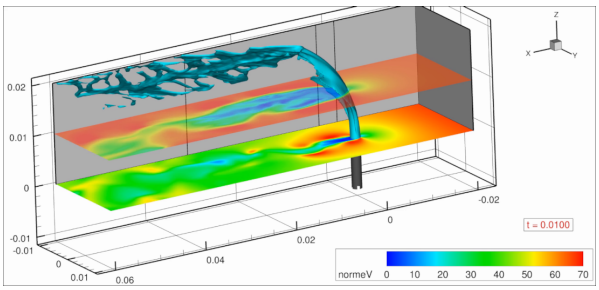


Figure 6: Setup used for the case of the liquid jet in cross flow.



(a) Global time step: elapsed time = 353.9 h.

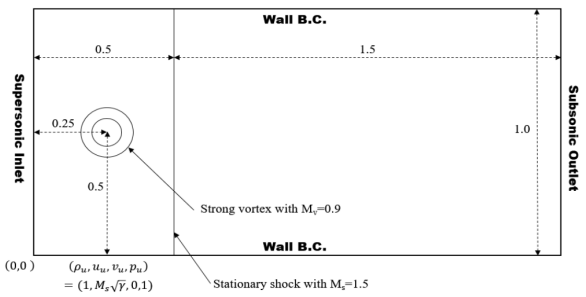


(b) Evolutive zonal time step (4 zones): elapsed time = 55.1 h.

Figure 7: Velocity field and isosurface of liquid volume fraction $\alpha = 0.5$ at time $t = 10$ ms.

the number of boundaries between the zones is magnified, and we also get four consecutive time levels activated in the zonal time step method (which means a ratio of $2^{4-1} = 8$ between the lowest and the highest time steps). Then three different simulations are performed with this set-up: the first one with a global time step (the same for all the zones), and the second and third ones with a zonal time step either with or without the conservation correction at the boundaries. Each simulation is run with the RK2 method and an evolutive time step driven by a maximum CFL value of 0.4.

Figure 10 shows the density field at the final time for each simulation (compared with that of the reference simulation on the very fine cartesian grid and the global time step method), while figure 11 gathers the density profiles extracted along the vertical line $x = 1.05$ for all these simulations (corresponding to the position of the vortex center at the final time). The interesting thing to notice is that the zonal time step method without the conservation correction is less accurate than the global time step method. In particular we can see that the error made in the conservation of the energy results in a bad propagation speed of the different waves, which are slightly mispositioned, as if the solution was not at the right time. This phenomenon disappears when we apply the conservation correction, but in top of that we also notice a slightly lower numerical diffusion in that case, which is due to the fact that with the zonal approach the time integration is performed more often with a higher CFL number. Finally, we can say that the zonal approach is also robust despite the shocks and discontinuities, but it is also worth



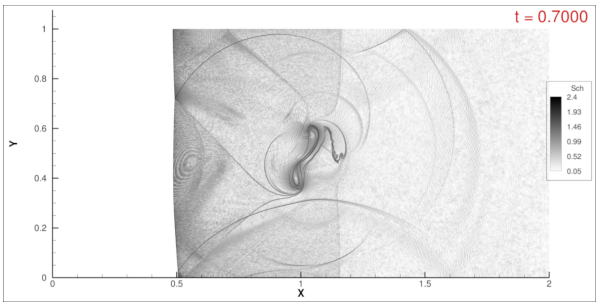


Figure 8: Setup of the vortex-shock interaction case (left) and reference solution obtained at time $t = 10$ ms on a 16-million cell cartesian grid (right).

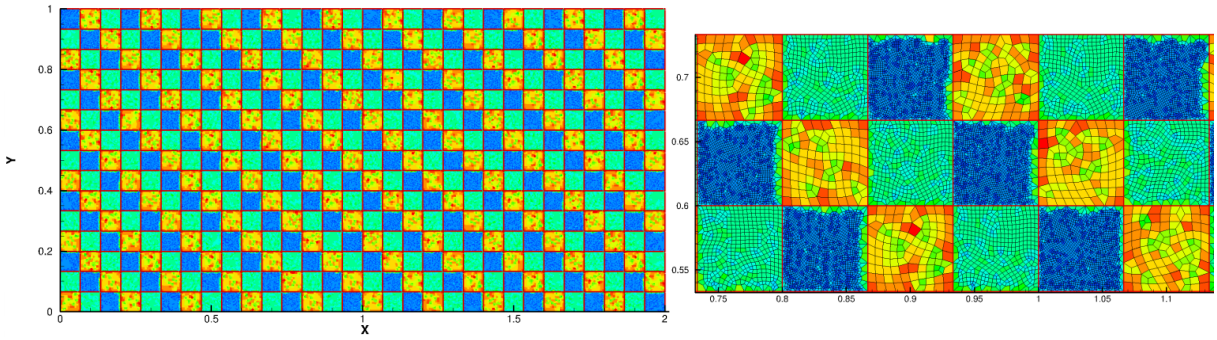
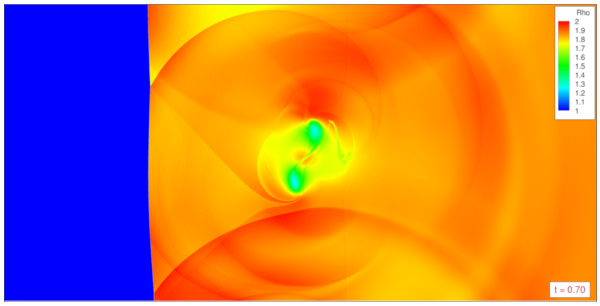
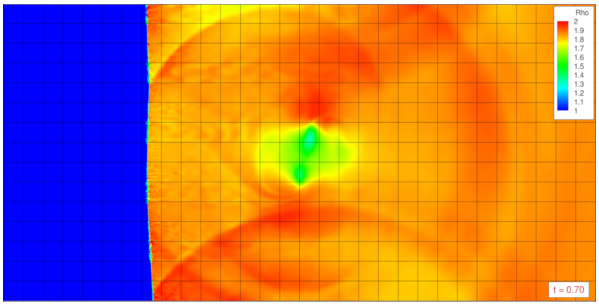


Figure 9: Vortex-shock interaction case: computational domain split into 450 zones with heterogeneous mesh refinement (zoom on the right figure).

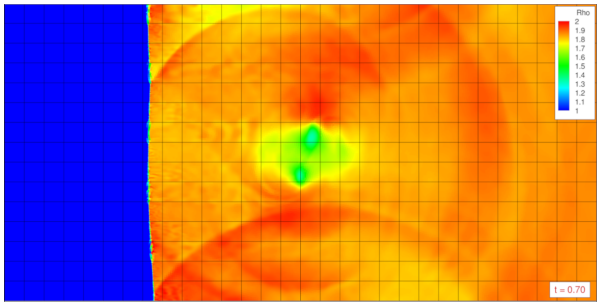
mentioning that this robustness may be challenged when the conservation correction is applied. Actually this is related to the issue of the time accuracy at boundaries, which is a crucial aspect that will be briefly discussed in the next section.



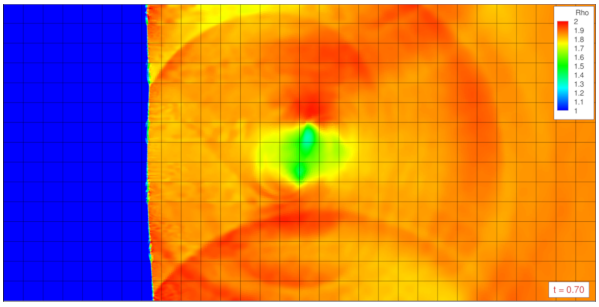
(a) 16M-cell cartesian grid (reference)



(b) Global time step



(c) Zonal time step **with correction**



(d) Zonal time step **without correction**

Figure 10: Vortex-shock interaction case: density fields at the final time $t = 0.7$.

4 PERSPECTIVES

The major perspective for the evolutive zonal time step method is related to the issue of the time accuracy at boundaries. The question is how to describe the evolution in neighboring zones between 2 synchronization times. Indeed, to preserve the time accuracy we need to account for

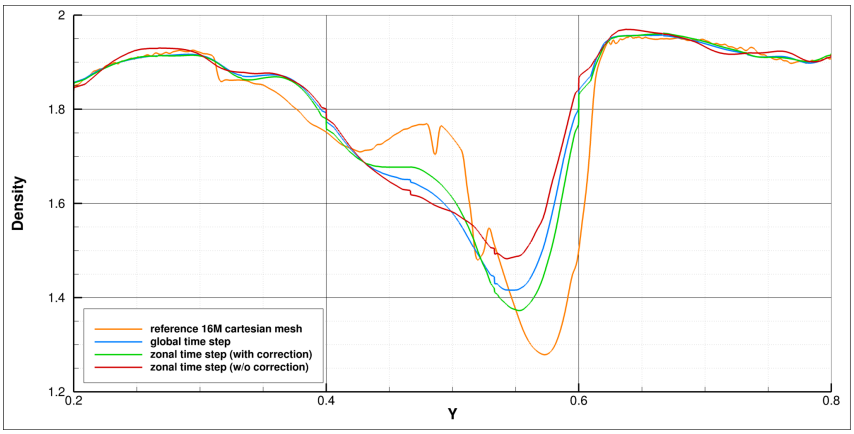
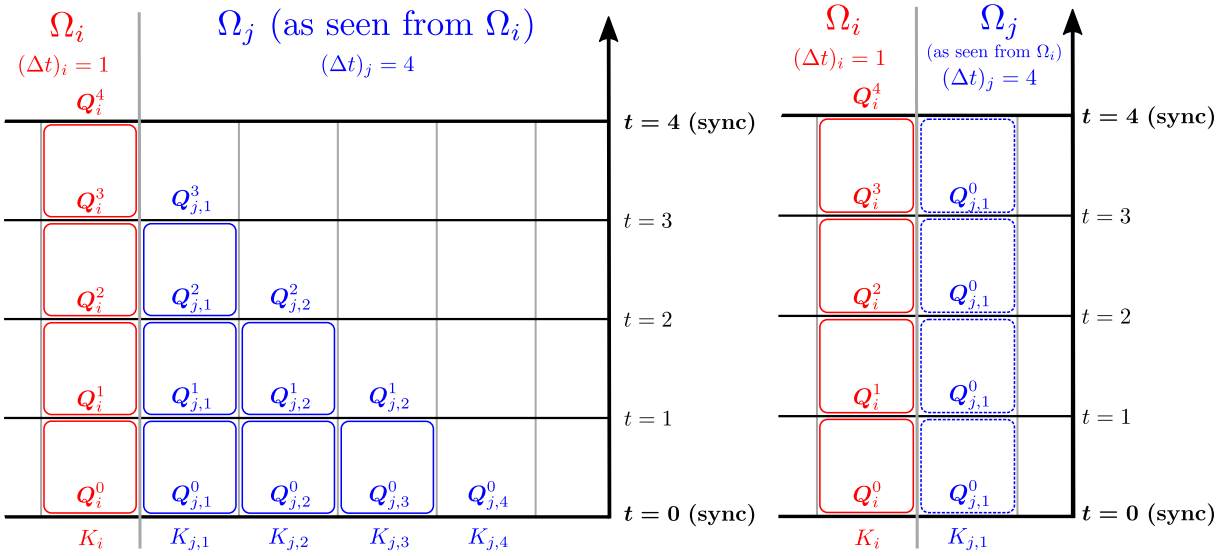


Figure 11: Vortex-shock interaction case: density profiles along the vertical line $x = 1.05$ at the final time $t = 0.7$.

a sufficient cell depth in the neighboring zone, which is often referred to as a "buffer zone", and whose depth depends on the time step ratio between the two zones, on the number of steps s of the RK method, and also on the stencil of the space discretization method. In the example shown in figure 12, we consider a simple case in which the time step ratio is 4 and with a first-order scheme in both space and time. In that case we need a 4-cell buffer zone, but the number of cell layers can rapidly increase with high-order methods and/or with higher time step ratios.



(a) A 4-cell buffer zone is required for time accuracy

(b) Current approach in CEDRE: frozen states in 1st layer of ghost cells

Figure 12: Example with $\frac{(\Delta t)_j}{(\Delta t)_i} = 4$ and a first-order method in space and time.

Most of the time-accurate methods that we find in the literature use the framework of partitioned RK methods [4, 6], which consists in using a specific RK method in each zone with ad

hoc coefficients to meet the accuracy conditions. The problem is that these methods are quite difficult to implement in an existing code because we need to deal with the buffer zone. Moreover they also suffer from a lack of flexibility, for example because the time step ratio is imposed between neighboring zones, or because it is impossible to use evolutive time steps driven by a CFL condition. This is why our current approach in CEDRE is simpler. We just use frozen states in the first layer of ghost cells at the boundaries between two synchronization times, as shown in figure 12. This has some practical advantages because this is easy to implement in the code and also more flexible because we can use any SSPRK method in each zone and any time step ratio. However it has the obvious drawback that the time accuracy is compromised at the boundaries, which amplifies the magnitude of the conservation correction and therefore may potentially end up in stability issues. And therefore this is currently our major work in progress.

But there are also some other perspectives on a longer term basis. First of all, we would like to propose an automatic and optimal building of the zones, so that the user may benefit from the optimal speedup without tedious manual work. We would also like to extend the criteria driving the evolutive zonal time step to include diffusion or source terms, and maybe extend the approach to implicit methods. Finally, the question of the compatibility with dynamic meshing methods will have to be addressed.

References

[1] A. Refloch, B. Courbet, A. Murrone, P. Villedieu, C. Laurent, P. Gilbank, J. Troyes, L. Tessé, G. Chaineray, J.B. Dargaud, E. Quémerais, and F. Vuillot. CEDRE Software. *AerospaceLab Journal*, 2:131–140, 2011.

[2] Sin-Chung Chang, Yuhui Wu, Vigor Yang, and Xiao-Yen Wang. Local time-stepping procedures for the space-time conservation element and solution element method. *International Journal of Computational Fluid Dynamics*, 19(5):359–380, July 2005. ISSN 1061-8562, 1029-0257. doi: 10.1080/10618560500092610. URL http://www.tandfonline.com/doi/abs/10.1080/10618560500092610.

[3] Chau-Lyan Chang, Balaji Shankar Venkatachari, and Gary Cheng. Time-Accurate Local Time Stepping and High-Order Space Time CESE Methods for Multi-Dimensional Flows Using Unstructured Meshes. American Institute of Aeronautics and Astronautics, June 2013. doi: 10.2514/6.2013-3069. URL http://arc.aiaa.org/doi/10.2514/6.2013-3069.

[4] Emil M. Constantinescu and Adrian Sandu. Multirate Timestepping Methods for Hyperbolic Conservation Laws. *Journal of Scientific Computing*, 33(3):239–278, October 2007. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-007-9151-y. URL http://link.springer.com/10.1007/s10915-007-9151-y.

[5] Margarete O. Domingues, Sonia M. Gomes, Olivier Roussel, and Kai Schneider. An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. *Journal of Computational Physics*, 227(8):3758–3780, April 2008. ISSN 00219991. doi: 10.1016/j.jcp.2007.11.046. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999107005219.

[6] G. Jeanmasson, I. Mary, and L. Mieussens. On some explicit local time stepping finite volume schemes for CFD. *Journal of Computational Physics*, 397:108818, November 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.07.017. URL http://www.sciencedirect.com/science/article/pii/S0021999119305029.

[7] C. Le Touze, A. Murrone, and H. Guillard. Multislope MUSCL method for general unstructured

meshes. *Journal of Computational Physics*, 284:389–418, March 2015. ISSN 00219991. doi: 10.1016/j.jcp.2014.12.032. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999114008493.

[8] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, August 1988. ISSN 0021-9991. doi: 10.1016/0021-9991(88)90177-5. URL http://www.sciencedirect.com/science/article/pii/0021999188901775.

[9] Sigal Gottlieb, David Ketcheson, and Chi-Wang Shu. *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*. WORLD SCIENTIFIC, January 2011. ISBN 978-981-4289-26-9 978-981-4289-27-6. doi: 10.1142/7498. URL http://www.worldscientific.com/worldscibooks/10.1142/7498.

[10] Thomas Unfer. Third order asynchronous time integration for gas dynamics. *Journal of Computational Physics*, 440:110434, September 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110434. URL https://www.sciencedirect.com/science/article/pii/S0021999121003296.

[11] CI2 – Inviscid Strong Vortex-Shock Wave Interaction | HiOCFD5. URL https://how5.cenaero.be/content/ci2-%E2%80%93-inviscid-strong-vortex-shock-wave-interaction.