A Volume-agglomeration multirate time advancing for high Reynolds number flow simulation

Emmanuelle Itam, Stephen Wornom, Bruno Koobus, Alain Dervieux

A Volume-agglomeration multirate time advancing for high Reynolds number flow simulation

Emmanuelle Itam, CNAM, Paris, France,
Stephen Wornom, Lemma, Biot, France,
Bruno Koobus, Univ Montpellier, France,
Alain Dervieux, Université Côte d'Azur, INRIA, France

2018
December

SummaryA frequent configuration in computational fluid mechanics combines an explicit time advancing scheme for accuracy purposes and a computational grid with a very small portion of much smaller elements than in the remaining mesh. Two examples of such situations are the travel of a discontinuity followed by a moving mesh, and the large eddy simulation of high Reynolds number flows around bluff bodies where together very thin boundary layers and vortices of much more important size need to be captured. For such configurations, multi-stage explicit time advancing schemes with global time stepping are very accurate, but also very CPU consuming. In order to reduce this problem, the multirate time stepping approach represents an interesting improvement. The objective of such schemes, which allow to use *different time steps* in the computational domain, is to avoid penalizing the computational cost of the time advancement of unsteady solutions which would become large due to the use of small global time steps imposed by the smallest elements such as those constituting the boundary layers. In the present work, a new multirate scheme based on control volume agglomeration is proposed for the solution of the compressible Navier-Stokes equations equipped with turbulence models. The method relies on a prediction step where large time steps are performed with an evaluation of the

fluxes on macro-cells for the smaller elements for stability purpose, and on a correction step in which small time steps are employed only for the smaller elements. The accuracy and efficiency of the proposed method are evaluated on several benchmarks flows: the problem of a moving contact discontinuity (inviscid flow), the computation with a hybrid turbulence model of flows around bluff bodies like a flow around a space probe model at Reynolds number 10^6, a circular cylinder at Reynolds number 8.4×10^6, and two tandem cylinders at Reynolds number 1.66×10^5 and 1.4×10^5.

computational fluid dynamics, multirate time advancing, explicit scheme, volume agglomeration, unstructured grid, hybrid turbulence model

1 INTRODUCTION

A frequent configuration in Computational Fluid Dynamics (CFD) calculations combines an explicit time advancing scheme for accuracy purpose and a computational grid with a very small portion of much smaller elements than in the remaining mesh. Two typical examples are the following:

A first example is the hybrid RANS/LES simulation of high Reynolds number flows around bluff bodies. In that case, very thin boundary layers need be addressed with extremely small cells. When applying explicit time advancing, the computation is penalized by the very small time-step to be applied (CFL number of order 1). But this is not the only interesting region of the computational domain. An important part of the meshing effort is devoted to large regions of medium cell size in which the motion of vortices need be accurately captured. For these vortices, the efficient and accurate time-step is of the order of the ratio of local mesh size by vortex velocity. We can apply an implicit scheme with such a time-step, which would produce a local CFL of order 1 for the vortices advection and a local CFL of order hundreds for the boundary layer. However, this may have several disadvantages. First this standpoint completely neglects a possible need of unsteady accuracy in the small-cell region, due to unsteady separation, for example. Second, considering the need of accuracy for vortices motion on medium cells, highly accurate explicit schemes are easily assembled. This includes the TVD third-order ones, and the standard fourth-order Runge-Kutta method (RK4). In contrast, high-order implicit schemes are complex and cpu consuming. More simple implicit schemes using backward differencing show much more dissipation than explicit schemes.

Our second example concerns an important complexity issue in unsteady mesh adaptation. Indeed, unsteady mesh adaptive calculations are penalized by the very small time-step imposed by accuracy requirements on regions involving small space-time scales. This small time step is for example an important computational penalty for mesh adaptive methods of AMR type [5]. This is also the case for unsteady fixed-point mesh-adaptive methods as in [4]. In that latter method, the loss of efficiency is even more crucial when the anisotropic mesh is locally strongly stretched. In [4], due to this loss, the numerical convergence order for discontinuities is limited to 8/5 instead of second-order convergence. This limitation also applies to mesh adaptation by mesh motion. In the present work, our first numerical example concerns the computation of an isolated traveling discontinuity. The discontinuity needs

to be followed by the mesh, preferably in a mesh-adaptive mode. Except if the adaptation works in a purely Lagrangian mode, an implicit scheme will smear the discontinuity of the solution. An explicit scheme will applied a costly very small time step on the whole computational domain.

In order to overcome these problems, the multirate time stepping approach represents an interesting alternative. A part of the computational domain is advanced in time with the small time-step imposed by accuracy and stability constraints. Another part is advanced with the larger time-step giving a good compromise between accuracy and efficiency.

The development of multirate schemes was first limited to ODEs with different time scales and their application was restricted to a small number of special industrial problems. The first multirate method was due to the pioneering work of Rice [25]. In this work, a system of first-order ODEs made of a latent component (slow variation) and an active component (fast variation) was considered. Runge-Kutta type integration methods were developed, in which different integration steps (large for the latent component and small for the active component) were used and appropriate extrapolations were made for the latent components when the active components are integrated. Following this early work, many other developments have been conducted in the field of multirate schemes and ODEs [25, 1, 10, 31, 26, 2, 16, 9, 8, 15, 14, 28]. These multirate strategies usually combine classic numerical integration methods (Backward Differentiation Formulas, Adams methods, Runge-Kutta schemes, Rosenbrock-Wanner methods), in which a large integration step is used for the slow subsystem of EDOs and a smaller integration step is used for the fast subsystem, with extrapolation/interpolation algorithms for coupling active to latent parts and vice versa. In order to assess their stability and efficiency, these multirate methods were applied to (more or less large) systems of stiff ODEs, often derived from electric circuits problems. As an example, in [16], a multirate 4-steps Rosenbrock-Wanner method was implemented to solve systems of $250 - 4000$ ODEs which model electric circuits, leading to a gain in efficiency up to 2.8 compared to a classic 4-steps Runge-Kutta method.
Fewer and more recent multirate works were conducted in the field of PDEs and hyperbolic conservation laws [6, 27, 24, 19, 29, 22], and rare applications were performed in CFD [22, 29]. The first multirate work dealing with hyperbolic problems and applications in CFD was made by Löhner *et al.* [22]. A domain-splitting method was developped in which the Euler equations are

discretized by a second order explicit finite element scheme on a domain splitted in several subregions of different grid resolution. For the time-integration of the equations, larger time steps are used in subregions of coarser mesh fineness leading to an increase in efficiency. One and two-dimensional test-cases were considered : a shock tube problem, and supersonic inviscid flows around a circular cylinder and past a wedge. The results showed that the method can handle shocks and that a speedup of 2 between the multirate scheme and its single-rate counterpart can be reached.
Another work of interest, more recent, is the one of Constantinescu *et al.* [6] which focuses on the development of multirate methods for the solution of one-dimensinal scalar hyperbolic equations. They propose strategies that are based on an appropriate transition between subregions of different local stability conditions where a classic Runge-Kutta scheme is used, by introducing buffer regions, where an adapted Runge-Kutta method is employed. For the case of the one-dimensional Burger equation, speedups up to 2.5 were obtained.

In the context of parallel computing, an interesting work on multirate methods is the recent one of Seny *et al.* [29]. This work focuses on the efficient parallel implementation of explicit multirate Runge-Kutta schemes in the framework of discontinuous Galerkin methods. The multirate Runge-Kutta scheme used is the approach proposed by Constantinescu [6]. In order to optimize the parallel efficiency of the multirate scheme, Seny *et al.* propose a solution based on multi-constraint mesh partitioning [18]. The objective is to ensure that the workload, for each stage of the multirate algorithm, is almost equally shared by each computer core i.e. the same number of elements are active on each core, while minimizing inter-processor communications. The Metis software is used for the mesh decomposition, and the parallel programming is performed with the Message Passing Interface (MPI). The efficiency of the parallel multirate strategy is evaluated on three test cases: the wind driven circulation in a square basin and the propagation of a tsunami wave using a shallow water model (two-dimensional), and the acoustic propagation in a turbofan engine intake using the linearized Euler equations (three-dimensional). It is shown that the multi-constraint partitioning strategy increases the efficiency of the parallel multirate scheme compared to the classic single-constraint partitioning. Speed-up as high as 4.4 were obtained.

In the present work, we propose a new multirate scheme based on control

volume agglomeration which is at the same time very simple to develop in an existing software relying on explicit time-advancing, and well suited to a large class of finite volume approximations. The agglomeration produces macro-cells by grouping together several neighboring cells of the initial mesh. The method relies on a prediction step where large time steps are used with an evaluation of the fluxes performed on the macro-cells for the region of smallest cells, and on a correction step advancing solely the region of small cells, this time with a small time step.

We demonstrate the method in a numerical framework using a vertex centered approximation, the mixed finite volume/finite element formulation.

Target applications are three-dimensional unsteady flows modeled by the compressible Navier-Stokes equations equipped with turbulence models and discretized on unstructured possibly deformable meshes. The numerical illustration involves the two above examples.

The proposed algorithm is described in Section 2. Section 3 provides some motivations of this construction. Section 4 gives several examples of applications.

2 Multirate time advancing by volume agglomeration

2.1 Finite-Volume Navier-Stokes

The proposed multirate time advancing scheme based on volume agglomeration will be denoted by MR and is developed for the solution of the three-dimensional compressible Navier-Stokes equations. The main assumption is that the computational domain is split into computational finite volume cells such that cells intersect only by their boundaries and cover the whole computational domain. The discrete Navier-Stokes system is assembled by a flux summation Ψ_i involving the convective and diffusive fluxes evaluated at all the interfaces separating cell i and its neighbors. More precisely, the finite-volume spatial discretization combined with an explicit forward-Euler time-advancing writes for the Navier-Stokes equations possibly equipped with a $k-\varepsilon$ model:

$$vol_i \; w_i^{n+1} = vol_i \; w_i^n \; + \; \Delta t \; \Psi_i, \quad \forall \; i = 1, ..., ncell,$$

where vol_i is the volume of cell i, Δt the time step, and $w_i^n = (\rho_i^n, (\rho u)_i^n, (\rho v)_i^n, (\rho w)_i^n, E_i^n, (\rho k)_i^n, (\rho \varepsilon)_i^n)$ are as usually the density, moments, total energy, turbulent energy and turbulent dissipation at cell i and time level t^n, and $ncell$ the total number of cells in the mesh.
In the examples given below, the accuracy of the initial scheme can be defined as a third-order spatial accuracy on smooth meshes, through the use of a MUSCL-type upwind-biased finite volume, combined with a third-order time accurate Shu multistage scheme, see [20] for details.

Given an explicit -conditionally stable- time advancing, we assume that we can define a maximal stable time step (*local time step*) $\Delta t_i, i = 1, ..., ncell$ on each node. For the Navier-Stokes model, the stable local time step is defined by the combination of a viscous stability limit and an advective one according to the following formula:

$$\Delta t_i \leq \frac{CFL \times \Delta l_i^2}{\Delta l_i(||\mathbf{u_i}|| + c_i) + 2\frac{\gamma}{\rho_i}\left(\frac{\mu_i}{Pr} + \frac{\mu_{t_i}}{Pr_t}\right)} \tag{1}$$

where Δl_i is a local characteristic mesh size, $\mathbf{u}_i$ the local velocity, c_i the sound celerity, γ the ratio of specific heats, ρ_i the density, $\frac{\mu_i}{Pr} + \frac{\mu_{t_i}}{Pr_t}$ the sum of local viscosity to Prandtl ratio, laminar and turbulent, and CFL a parameter depending of the time advancing scheme, of the order of unity. Using the local time step Δt_i leads to a stable but not consistent time advancing. A stable and consistent time advancing should use a *global/uniform time step* defined by:

$$\Delta t = \min_{1,ncell} \Delta t_i.$$

For many advective explicit time advancing, in regions where Δt is of the order of Δt_i, accuracy is quasi-optimal, and in other regions, the accuracy is suboptimal, due to the relatively large spatial mesh size.

2.2 Inner and outer zones

We first define the inner zone and the outer zone, the coarse grid, and the construction of the fluxes on the coarse grid, ingredients on which our MR time advancing scheme is based. For this splitting into two zones, the user is supposed to choose a (integer) *time step factor* $K > 1$. We define the **outer zone** as the set of cells i for which the explicit scheme is stable for a time step $K\Delta t$

$$\Delta t_i \geq K\Delta t,$$

the **inner zone** is the set of cells i for which

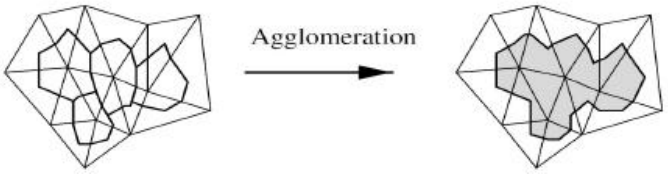
$$\Delta t_i \ < \ K\Delta t.$$

We shall build over the whole domain a coarse grid which should allow that:

- Advancement in time is performed with time step $K\Delta t$,
- Advancement in time preserves accuracy in the outer zone,
- Advancement in time is consistent in the inner zone.

A **coarse grid** is defined on the inner zone by applying cell agglomeration in such a way that on each macro-cell, the maximal local stable time step is at least $K\Delta t$. Agglomeration consists in considering each cell and aggregating to it neigboring cells which are not yet aggregated to an other one (Figure 1). Agglomeration into macro-cell is re-iterated until macro-cells with maximal time step smaller than $K\Delta t$ have disappeared. We advance in time the chosen explicit scheme on the coarse grid with $K\Delta t$ as time step. The nodal fluxes Ψ_i are assembled on the fine cells (as usual). Fluxes are then summed on the macro-cells I (inner zone) :

$$\Psi^I = \sum_{k \in I} \Psi_k. \tag{2}$$



fine cells i macro-cell I

Figure 1: Sketch (in 2D) of the agglomeration of 4 cells into a macro-cell. Cells are dual cells of triangles, bounded by sections of triangle medians.

2.3 MR time advancing

The MR algorithm is based on a **prediction step** and a **correction step** as defined hereafter :

Step 1 (prediction step) :
The solution is advanced in time with time step $K\Delta t$, on the fine cells in the outer zone and on the macro-cells in the inner zone (which means using the macro-cell volumes and the coarse fluxes as defined in expression (2)) :

For $\alpha = 1, nstep$

$$\text{outer zone} : vol_i w_i^{(\alpha)} = a_\alpha vol_i w_i^{(0)} + b_\alpha \big(vol_i w_i^{(\alpha-1)} + K\Delta t \ \Psi_i^{(\alpha-1)} \big) \quad (3)$$

$$\text{inner zone} : vol^I w^{I,(\alpha)} = a_\alpha vol^I w^{I,(0)} + b_\alpha \big(vol^I w^{I,(\alpha-1)} + K\Delta t \ \Psi^{I,(\alpha-1)} \big) \quad (4)$$

$$w_i^{(\alpha)} = w^{I,(\alpha)} \quad \text{for } i \in I \quad (5)$$

EndFor α.

where $w^{I,(\alpha)}$ denotes the fluid/turbulent variables at macro-cell I and stage α, (a_α, b_α) are the multi-stage parameters, $a_1 = 0, a_2 = 3/4, a_3 = 1/3$ and $b_1 = 1, b_2 = 1/4, b_3 = 2/3$ for the three-stage Shu time advancing [30], and vol^I is the volume of macro-cell I.

From a practical point of view, we do not introduce in our software a new variable w^I to that already existing w_i. In other words, the previous sequence (4) is actually

$$\text{inner zone} : vol^I w_i^{(\alpha)} = a_\alpha vol^I w_i^{(0)} + b_\alpha \big(vol^I w_i^{(\alpha-1)} + K\Delta t \ \Psi^{I,(\alpha-1)} \big) \quad (6)$$

$$\text{where } I \text{ is the macro-cell containing cell } i. \quad (7)$$

On the other hand, the coarse fluxes Ψ^I are not stored in a specific variable, which means that no extra storage is necessary than the one required

to store the fluxes Ψ_i. Indeed, after the computation of the fluxes $\Psi_i^{(\alpha-1)}$ (using the values of $w_i^{(\alpha-1)}$) at each stage α of the multi-stage time advancing scheme, the coarse flux $\Psi^{I,(\alpha-1)}$ is evaluated according to expression (2) for each macro-cell I and then stored in the memory space allocated to $\Psi_i^{(\alpha-1)}$ for each cell i in the inner zone belonging to the macro-cell I.

Step 2 (correction step) :

- The unknowns in the outer zone are frozen at level $t^n + K\Delta t$.
- The unknowns in the outer zone close to the inner zone, which are necessary for advancing in time the inner zone (which means those which are useful for the computation of the fluxes Ψ_i in the inner zone), are interpolated in time.
- Using these interpolated values for the computation of the fluxes Ψ_i in the inner zone (at each stage of the time-advancing sheme), the solution in the inner zone is advanced in time with the chosen explicit scheme and time step Δt.

This time advancing writes:

For $kt = 1, K$
  For $\alpha = 1, nstep$

$$\text{inner zone} : vol_i w_i^{(\alpha)} = a_\alpha vol_i w_i^{(0)} + b_\alpha \big(vol_i w_i^{(\alpha-1)} + K\Delta t \ \Psi_i^{(\alpha-1)} \big) \quad (8)$$

$$\text{(outer zone : nothing is done)} \quad (9)$$

  EndFor α.
EndFor kt.

The arithmetic complexity, proportional to the number of points in the inner zone, is therefore mastered.

3 Elements of analysis

3.1 Stability

The central question concerning the coarse grid is the stability resulting from its use in the computation. Considering (1), we expect that the viscous stability limit will improve by a factor four (1D) for a twice larger cell. The viscous stability limit can therefore be considered as more easily addressed by our coarsening. For the advective stability limit, we can be a little more precise. The coarse mesh is an unstructured partition of the domain in which cells are polyhedra. Analyses of time advancing schemes on unstructured meshes are available in L^2 norm for unstructured meshes, see [3] [12] [11]. Here we solely propose a L^∞ analysis of the first order advection scheme. The gain in L^∞ stability can be analysed for a first-order upwind advection scheme. We get the following (obvious) lemma:

Lemma : The upwind advection scheme is positive on the mesh made of macro-cells as soon as for all macro-cell I :

$$\Delta t \; ||V_I|| \quad < \quad \Big[\sum_{J \in \mathcal{N}(I)} \int_{\partial cell(I) \cap \partial cell(J)} \mathrm{d}\Sigma \Big]^{-1} \quad \int_{cell(I)} \mathrm{d}\mathbf{x}$$

where $\mathcal{N}(I)$ holds for the neighbouring macro-cells of I.$\square$

The application of an adequate neighboring-cell agglomeration, producing large macro-cells of good aspect ratio will produce a K-times larger stability limit.

3.2 Accuracy

In contrast to more sophisticated MR algorithms, the proposed method has not a rigorous control of the accuracy. Let us however remark that the generic situation involves variable-size meshes, which limits the unsteady accuracy on small scales propagation, already before applying the MR method.

However the two following remarks tend to show that the scheme accuracy is conserved:

- the predictor step involves simply a sum of the fluxes and the formal accuracy order is kept, with a coarser mesh size.

- still during the predictor step, if we assume that the mesh is reasonably smooth, then the CFL applied in the inner part near the matching zone will

be close to the explicit CFL (applied on the outer part near the matching zone) and therefore accuracy is high.

Under these conditions, the effect of the corrector step will be just improving the result.

In practice, most of our experiments will involve a comparison between the three-stage Shu explicit time advancing and the MR algorithm using the same explicit scheme.

3.3 Efficiency

Two-level MR: The proposed method depends on only one parameter, the ratio K between the large and small time step. Considering a mesh with N vertices, a short loop on the mesh will produce the function $K \mapsto N^{inner}(K) \leq N$ which gives the number of cells in the inner region for K.

If $CPU_{ExpNode}(\Delta t)$ denotes the CPU per node and per time step Δt of the underlying explicit scheme, a model for the MR cpu per Δt would be

$$CPU_{MR(K)}(\Delta t) = \left(\frac{N}{K} + N^{inner}(K) \right) \times CPU_{ExpNode}(\Delta t) \qquad (10)$$

to be compared with the explicit case:

$$CPU_{Expli}(\Delta t) = N \times CPU_{ExpNode}(\Delta t).$$

We shall call the *expected gain* the ratio:

$$\text{Gain} = \frac{CPU_{Expli}(\Delta t)}{CPU_{MR(K)}(\Delta t)} = \frac{1}{\frac{1}{K} \ + \ \frac{N^{inner}(K)}{N}}.$$

The above formula emphasizes the crucial influence of a very small proportion of inner cells.

Remark 1: In most other multirate methods, the phase with a larger time-step does not concern the inner region and then their gain would be modelled by:

$$\text{Gain} = \frac{1}{\frac{1}{KN}(N - N^{inner}(K)) \ + \ \frac{N^{inner}(K)}{N}}.$$

Both gains are bounded by $N/N^{small}(K)$ and show that this ratio has to be sufficiently large.□

Three-level MR: Multirate strategies are supposed to extend to more than two different time step lengths while keeping a reasonable algebraic complexity. Let us examine the case of three lengths, namely Δt, $K\Delta t$, $K^2\Delta t$. It is then necessary to generate two *nested* levels of agglomeration in such a way that $Grid1$ is stable for $CFL = 1$, $Grid2$ is stable for $CFL = K$, $Grid3$ is stable for $CFL = K^2$. While a two-rate calculation would involve a prediction-correction based on $Grid3$ and $Grid2 \cup Grid1$ and a factor K:
- prediction on $Grid3 \cup Grid2 \cup Grid1$,
- correction on $Grid2 \cup Grid1$,
in a three-rate calculation, the correction step is replaced by two corrections and the whole process writes:
- prediction on $Grid3 \cup Grid2 \cup Grid1$,
- correction on medium part of $Grid2 \cup Grid1$,
- correction on inner part of $Grid1$,
but this replacement is just the substitution (on a part of the mesh) of a single-rate advancing by a two-rate one and therefore can carry a higher efficiency (the smallest time step is restricted to a smaller inner zone $Grid1$). In contrast to other MR methods, we have a (second) duplication of flux assembly on the inner zone:

$$CPU_{MR(K)}(\Delta t) = \left(\frac{N^1 + N^2 + N^3}{K^2} + \frac{N^1 + N^2}{K} + N^1(K) \right) \times CPU_{ExpNode}(\Delta t) \tag{11}$$

to be compared with (10), noting that $N = N^1 + N^2 + N^3$. However, this increment by a factor $1 + K^{-1} + K^{-2}$ remains limited (111% for $K = 10$).

3.4 Impact of our MR complexity on mesh adaption

An important impact of MR methods is the increment of accuracy order in unsteady mesh adaption methods. We now check that the proposed MR indeed improves mesh adaption accuracy order. Let us consider the space-time mesh used by a time advancing method. A usual time advancing uses the Cartesian product $\{t_0, t_1, ...t_N\} \times$ spatial mesh as space-time mesh. The space-time mesh is a measure of the computational cost since the discrete derivatives are evaluated on each node (t_k, x_k) of the $N_{time} \times N_{space}$ nodes of

the space-time mesh.

In [7] is proposed an analysis which determines the maximal convergence order (in terms of number of space-time nodes) which can be attained on a given family of mesh. This analysis is useful for evaluating mesh-adaptive methods. For example, in [4], a 3D mesh adaptive method for computing a traveling discontinuity has a convergence order α not better than $\alpha_{max} = 8/5$, according to

$$error = O(N_{st}^{-\alpha/4}), \tag{12}$$

4 being the space-time dimension. The purpose of this section is to show that replacing the usual time-advancing by the MR algorithm will indeed improve the maximal convergence order of a mesh adaption method defined as in [4].

Let us concentrate on the calculation of a planar horizontal discontinuity vertically travelling in a 3D spatial cubic domain and its approximation by a second-order accurate P_1 finite-element method. The approximation error is of first order (space and time) near the discontinuity. Then starting from an uniform space-time mesh, dividing mesh size and time step by a factor 4 will divide by 4 the error, at the cost of $N_2/N_1 = 256$ times more space-time nodes. We verify that, according to (12), this is an order of 1.

In order not to go into many details given in [4] , we ask the reader to believe us if we say that an anisotropic mesh adaptor is able to improve the spatial mesh of each time level in such a way that starting from a constant time step and adapted space meshes at each time level, and dividing the time step by 4 and using adapted meshes with only 8 times more nodes than before will produce a 4 times smaller error. In short, this is due to the fact that mesh size normal to the discontinuity can be divided by 4, ensuring a 4 times smaller error with only a small fraction of N_{space}, typically of the order $N_{space}^{2/3}$, the number of points lying on the planar discontinuity. The resulting performance an order of $\frac{8}{5}$ is poor with respect to second order due to the uniform division by 4 of the time step. With a MR time advancing, the four-times smaller time step can be restricted to the $N_{space}^{2/3}$ smaller spatial cells and a two-times smaller time step is applied to the others. The amplification of CPU time is then $16 + 32 * N_{space}^{-1/3}$ for the proposed algorithm and $(16 * (1 - N_{space}^{-1/3}) + 32 * N_{space}^{-1/3}$ for a usual MR algorithm). Both formulas, for N_{space} large, give the second order ($\alpha = 2$) convergence.

3.5 Parallelism

3.5.1 Implementation of MR

The proposed method is experimented with a parallel MPI software relying on mesh partitioning. In a preprocessing phase, the cell-agglomeration is applied at run time inside each partition, which saves communications. Further, it is done once for the whole computation, while fluctuations of the inner zone will be taken into account by using solely the agglomeration of the cells in inner zone at each time level. Since our purpose is to remain with a rather simplified modification of the initial software, we did not modify the communication library in order to restrict the communications to the inner zone when it is possible (*i.e.* in the correction step). Due to this, the complexity of the correction step is not strictly of order of the number of nodes in the inner zone, while its arithmetic complexity satisfies this condition.

To synthetize, the MR algorithm involves at each time step:
- an updating of the inner zone (with a volume agglomeration done once for the computation),
- a prediction step which is similar to an explicit step (with a larger time step length), but with also a local sum of the fluxes in each macro-cell,
- a correction step which is similar to explicit arithmetics restricted to the inner region, and for simplicity of coding, communications which are left identical to the explicit advancing, that is communications applied to both inner and outer zones.

An intrinsic extra cost of our algorithm with respect to previous multirate algorithms is the computations on the inner zone during the predictor step. An intrinsic cost is also the local sums on macro-cell which account for a very small part of the computational cost.
The correction step complexity is close to an explicit advancing one on the inner zone except the two phases of time interpolation and communications. Time interpolation can be implemented with a better efficiency by applying it only on a layer around the inner zone. However, the cost of the time interpolation is a very small part of the total cost. Global communications are less costly than 10% of the explicit time step cost. If the inner zone is 30% of the domain, developing communication restricted to inner zone will reduce the communication from 10% to 3% of the explicit time step on the whole domain, which shows that the correction step would be decreased from

40% to 33% of an explicit time stepping CPU.

3.5.2 Load balancing

The usual Metis software can be applied on the basis of a balanced repartion of the mesh. However, as remarked in previous works (see for example [29]), if the mesh partition does not take into account the inner zone, then the work effort will not be balanced during the correction step. The bad work balance for correction step can be of low impact if this step concerns a sufficiently small part of the mesh, resulting in a small part of the global work. However, a more reasonable assumption is that the correction phase represents a non-negligible part of the effort. In this section we discuss the question of a partitioning taking into account the correction phase. We observe that in the proposed method the inner zone depends on the flow through the CFL condition. This means that dynamic load balancing may be necessary. However, in the important class of flow which we consider, the change in inner zone can be neglected and we consider only static balancing. An option resulting from the work of Karypis and co-workers [18] and available in Metis is the application of a multi-constrained communication cost minimisation, with the two constraints that:
- partition is balanced for the whole computational domain, which will be favourable to the prediction step,
- partition is balanced for the inner part of the computational domain, which will be favourable to the correction step.
This partitioning algorithm produces a compromise between:
- the number of nodes in each subdomain of the global mesh,
- the number of nodes in each subdomain of the inner part of the mesh,
- the communications between the subdomains which are minimized in the multi-constrained algorithm.

In some particular cases, the user can specify an evident partition which perfectly balances the number of nodes in each subdomain of the global mesh and in each subdomain of the inner part of the mesh. In our experiments, we explicitly specify when it is the case and how it is performed.

4 Numerical tests

The MR algorithm is implemented into the MPI-parallel code AIRONUM shared by INRIA, Lemma company and university of Montpellier. A description of this tool, which solves with a mixed element/volume method on unstructured meshes the compressible Euler and Navier-Stokes equations possibly equipped with a turbulence model, can be found in [23] and [17]. The explicit time-advancing is a three-stage Shu method. The mean CPU for an explicit time step per mesh node varies between 10^{-7} and 4×10^{-7} seconds according to the partition quality and the number of nodes per subdomain.

For some of the test cases, it will be interesting to compare the efficiency and accuracy of the proposed MR time advancing with an implicit calculation of the same flow over the same time interval. The implicit algorithm (BDF2) which we use combines a second-order backward differencing formula for the time quadrature and a GMRES linear solver using a Restrictive-Additive Schwarz preconditioner and ILU(0) in each partition, see [20] for further details. In the cases computed with the implicit scheme, the CFL is fixed to 30 and the total number of GMRES iterations for one time step is around 20. For this CFL, the gain of an implicit computation with respect to an explicit one at CFL 0.5 is measured between 12. and 22. depending on the number of nodes per processor. The implicit scheme scalability decreases with partitions less than $10,000$ vertices while the explicit (and MR) scheme remains scalable for partitions of $5,000$ vertices. The BDF2 algorithm is second-order accurate in time and we shall use this property when estimating which time step reduction is necessary for reducing by a given factor the deviation with respect to explicit time-advancing.

4.1 Contact discontinuity

In this first example, we consider the case of a moving contact discontinuity. For this purpose, the compressible Euler equations are solved in a rectangular parallelepiped as computational domain where the density is initially discontinuous at its middle (see Figure 2) while velocity and pressure are uniform.

The uniform velocity is a purely horizontal one. As can be seen in Figure 2, small cells are present on either side of the discontinuity. The mesh moves during the computation in such a way that the nodes located at the discon-

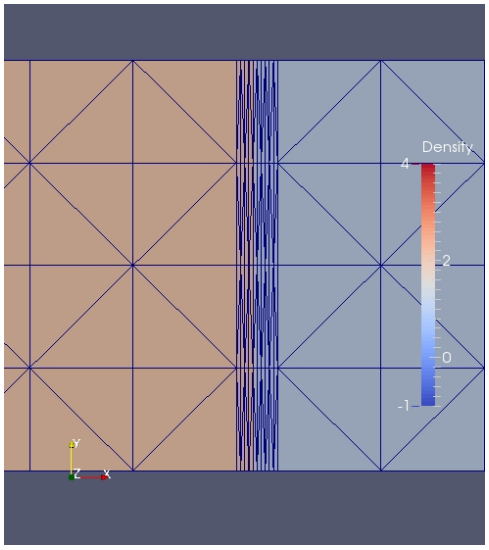


Figure 2: **ALE calculation of a traveling contact discontinuity**. Instantaneous mesh with mesh concentration in the middle of zoom and corresponding advected discontinuous fluid density.

tinuity are still the same, and that the number of small cells are equally balanced on either side of the discontinuity. An Arbitrary Lagrangian-Eulerian formulation is then used to solve the Euler equations on the resulting deforming mesh. Our long term objective is to combine the MR time advancing with a mesh adaptation algorithm in such a way that the small time steps imposed by the necessary good resolution of the discontinuity remain of weak impact on the global computational time.

The 3D mesh used in this simulation contains 25000 nodes and 96000 tetrahedra. The computational domain is decomposed into 2 subdomains, the partition interface being defined in such a way that it follows the center plan of the discontinuity. When integer K, used for the definition of the inner and outer zones, is set to 5, 10 and 15, the percentage of nodes located in the inner zone is always 1.3%, which corresponds to the vertices of the small cells located on either side of the discontinuity. The CFL with respect to propagation is 0.5. The MR scheme with the aforementioned values of K is used for the computation. Each simulation was run on 2 cores of a Bullx B720 cluster. In Table 1, CPU times (prediction step / correction step) are given for the MR approach and different time step factors K. The

K	$N^{small}(K)/N$ (%)	Expected gain (theoretical)	CPU pred. step (s/KΔt)	CPU correc. step (s/KΔt)	Measured gain (parallel)
5	1.3	4.7	0.124	0.244	1.7
10	1.3	8.8	0.124	0.482	2.0
15	1.3	12.5	0.124	0.729	2.2

Table 1: **ALE propagation of a contact discontinuity**: Time step factor K, CPU of the explicit scheme per explicit time-step Δt and per node, percentage of nodes in the inner region, theoretical gain in scalar mode, CPU of the prediction step per time-step $K\Delta t$, CPU of the correction step per time-step $K\Delta t$, and measured parallel gain.

correction step, consists of explicit time advancing on inner zone, 1.3% of the mesh (solely 78 vertices on each partition), but, due to parallel and vector inefficiency, one Shu step of it is 39% of one Shu explicit step on the whole mesh. As a result, an improvement in the efficiency of about 1.7, 2.0 and 2.2 is observed when K is set to 5, 10 and 15, respectively, unstead of the 4.7, 8.8 and 12.5 predicted by the theory.

4.2 Spatial probe

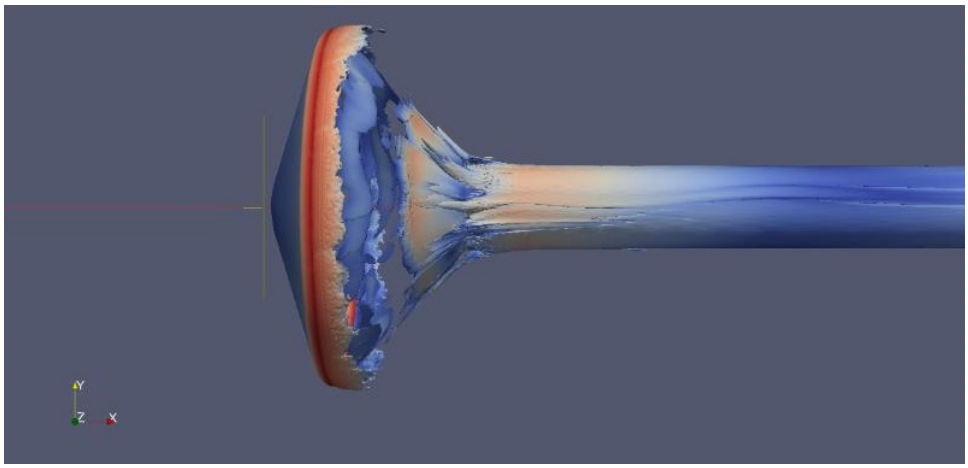


Figure 3: **Spatial probe at Reynolds number** 1 **million**. Q criterion.

K	$\frac{N^{small}}{N}$ (%)	Expected gain (theoretical)	CPU pred./corr. (s/KΔt)	Measured gain (parallel)	Error (%)
10	0.015	8.69	1.81/4.36	2.93	1 10−5
40	0.04	15.38	1.83/17.3	3.82	1.6 10−4
BDF2					
CFL=30				36.	$2\ .10^{-2}$
CFL=2.7(est.)				3.29	$1.6\ .10^{-4}$

Table 2: **Spatial probe**: Time step factor K, CPU of the explicit scheme per explicit time-step Δt and per node, percentage of nodes in the inner region, theoretical gain in scalar mode, CPU of the prediction step per time-step $K\Delta t$, CPU of the correction step per time-step $K\Delta t$, measured parallel gain, and relative lift deviation with explicit scheme after a shedding period.

We now consider a less academic example, the supersonic flow around a probe model for Exomars (see for example [13]). The Reynolds number is 1 million with respect to probe diameter. Delicate features in this simulation are a separation arising on a highly curved wall and a relatively large recirculation zone at afterbody. A hybrid RANS-LES calculation [17], which brings more information than pure RANS does, is used for this simulation. The mesh involves $4,380,000$ cells and the smallest element thickness is 2.10^{-5}. The smallest cells are only a few and are concentrated near the largest radius of the probe. The mesh is partitioned with the usual option of Metis, into 192 subdomains. A sketch of the flow computed is presented in Figure 3. Two typical values of K are shown (see Table 2). For $K = 10$, only 657 cells are in the inner zone (located in the high curvature region of the boundary layer where few very small cells are present. The gain in efficiency is already 2.93. This illustrates the interest of the proposed MR when the mesh involves only a very small number of small cells. For $K = 40$, yet only 1752 cells are in the inner zone. The gain is 3.82, much smaller than the scalar theoretical evaluation. Further, for higher K the gain does not improve. In this particular case, the limitation is not related to an increased number of inner cells, but to the inefficient parallel treatment of the inner zone, due to a poor and heterogeneous partition of it, since the 1752 cells are distributed

on 16 subdomains, the other 176 containing none of these cells. In order to

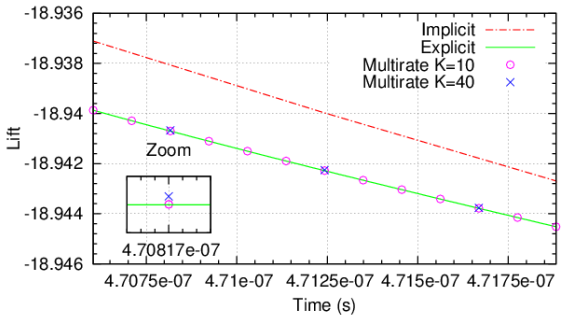


Figure 4: **Spatial probe at Reynolds number 1 million**. Zoom of the lift curves obtained with explicit, implicit and MR schemes.

evaluate the impact on accuracy, we consider as initial condition the result of a preliminary calculation giving an already installed vortex shedding. Then we continue the computation for one shedding cycle with either the explicit time advancing, or the MR one, or the BDF2 implicit time advancing (Figure 4). For 20 shedding cycles, the BDF2 run would produce a 0.4% deviation, which is probably reasonable for measuring bulk coefficients and probably not enough accurate for high frequency fluctuations. The deviation of the MR run is 125 times smaller for $K = 40$. According to the second-order accuracy of the BDF2, the CFL should be reduced to 2.7 if the user needs the same level of accuracy as MR. Then the measured gain is no longer in favor of the implicit algorithm (see Table 2).

4.3 Circular cylinder at very high Reynolds number

The third application concerns the simulation of the flow around a circular cylinder at Reynolds number 8.4×10^6. As for the previous benchmark, the computational domain is made of small cells around the body in order to allow a proper representation of the very thin boundary layer that occurs at such a high Reynolds number. On the other hand, the same hybrid RANS/VMS-LES model as that of the previous benchmark is used to compute this flow, which implies again that both the fluid and turbulent variables need to be advanced by the time integration scheme, and therefore also the MR method. Figure 5 depicts the Q-criterion isosurfaces and shows the very

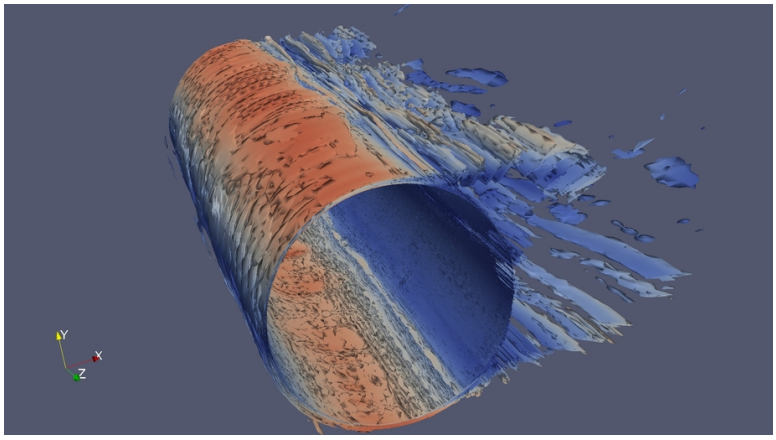


Figure 5: **Circular cylinder at Reynolds number** 8.4×10^6. Instantaneous Q-criterion isosurfaces (coloured with velocity modulus).

small and complex structures that need to be captured by the numerical and the turbulence models, which renders this simulation very challenging.

The mesh used in this simulation contains 4.3 million nodes and 25 million tetrahedra. The smallest cell thickness is $2.5.10^{-6}$. The computational domain is decomposed into 768 subdomains. When integer K, used for the definition of the inner and outer zones, is set to 5, 10 and 20, the percentage of nodes located in the inner zone is 15%, 19% and 24%, respectively (see Table 3). For each simulation, 768 cores were used on a Bullx B720 cluster,

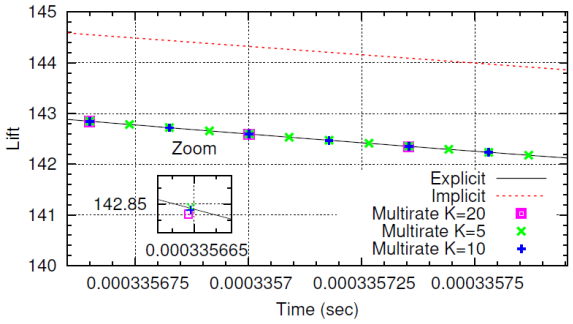


Figure 6: **Circular cylinder at Reynolds number** 8.4×10^6. Zoom of the lift curves obtained with explicit, implicit and MR schemes.

and the CFL number was set to 0.5. CPU times for the explicit and MR

K	$nproc$	$\frac{N^{small}}{N}$ (%)	Expected gain (theoretical)	Measured gain (UP/MCP/R)	Error (%)
20	192	24	3.45	1.18/1.43/2.27	$2.6\ 10^{-3}$
60	192	27	3.48	1.21/1.52/2.32	$5.\ 10^{-3}$
BDF2					
CFL=30	192			$20./-/-$	1.0
CFL=2(est.)	192			$1.5/-/-$	$5.\ 10^{-3}$

Table 3: **Circular cylinder at Reynolds number** 8.4×10^6 Time step factor K, CPU of the explicit scheme per explicit time-step Δt and per node, percentage of nodes in the inner region, theoretical gain in scalar mode, CPU of the prediction step per time-step $K\Delta t$, CPU of the correction step per time-step $K\Delta t$, measured parallel gain, and relative error for the explicit, MR and implicit BDF2 time advancing. UP holds for usual partition, MCP for Metis multi-constrained partition, R for analytic radial optimal partition.

schemes with different values of K are given in Table 3. One can observe that the efficiency of the MR approach is rather moderate. The cost of the correction step is indeed relatively high compared to the prediction step. This is certainly due to an important number of inner nodes (which implies also a moderate theoritical scalar gain) and a non uniform distribution of these nodes among the computational cores. An implicit simulation, with a CFL number set to 30, was also performed. An important gain is observed compared to the MR case, but at the cost of a degradation of the accuracy (see Table 3 and Figure 6) . This is probably related to the very small scale fluctuations which arise at this Reynolds number (Figure 5). They need be captured with a rather accurate time advancing. A 1% deviation after a shedding cycle may become 20% after 20 cycles and deteriorate the prediction of bulk fluctuations. In order to obtain the same level of error, the implicit time advancing, which is second-order accurate in time, should be run with a CFL of 2, with a gain of only 1.5.

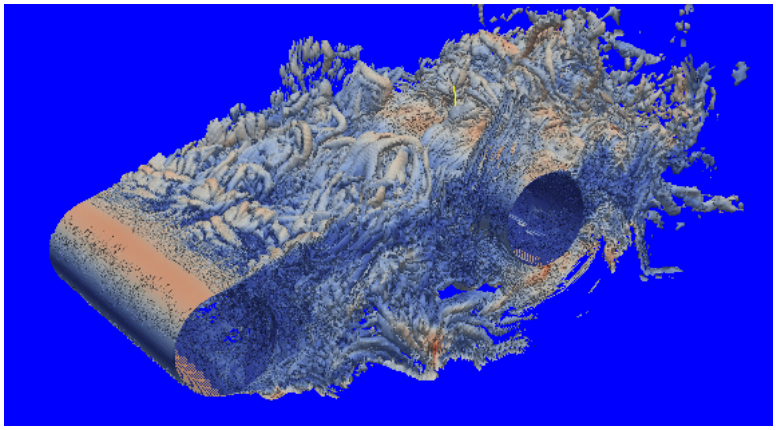


Figure 7: Tandem cylinders with distance 3.7D: instantaneous Q-criterion isosurfaces (coloured with velocity modulus).

4.4 Tandem cylinders with a separation distance of 3.7D

We turn now to a rather well-known benchmark test case, the calculation of a flow around a tandem cylinders at Reynolds number 1.66×10^5. This was a test case of an AIAA workshop, see [21]. It is a challenging computation since several complex flow features need to be captured around and between the two bodies (stagnation zones, boundary layers, shear layers, separations, laminar-turbulent transition, recirculations, vortex sheddings, wakes). In order to illustrate the quality of resolution with 16 milion vertices, the Q-criterion isosurfaces are shown in Figure 7. It shows the complex flow features and the very small structures that need to be captured by the numerical model and the turbulence model, which renders this simulation particularly challenging. Further information concerning the comparison between computation and experiments are available in [17].

When integer K, used for the definition of the inner and outer zones, is set to 20, the percentage of nodes located in the inner zone is 35% (see Table 5), which is a little too large for MR efficiency. The theoretical scalar gain is already rather small in this case, and the MR option turns out to show only a gain of 2 when 768 processors are used.

K	*nproc.*	$\frac{N^{small}}{N}$ (%)	Expected gain (theoretical)	Measured gain (UP/MCP)
20	192	35	2.50	1.12/1.77
20	768	35	2.50	1.02/2.0

Table 4: **Tandem cylinder with distance 3.7 D**. Time step factor K, number of processors, percentage of nodes in the inner region, theoretical gain in scalar mode, and measured parallel gain. UP holds for usual partition and MCP for multi-constrained partition.

4.5 Tandem cylinders with a large separation distance

We turn now to the flow around a tandem cylinder with a larger separation distance. As mentioned in the previous case, several complex flow features need to be captured around and between the two bodies. Small cells are necessary for a proper prediction of the very thin boundary layers, which implies very small global time steps so that classical explicit calculations become very costly. The case chosen here presents a supplementary difficulty since the distance between the two cylinders is 12 times a cylinder diameter. This kind of flow is of interest typically to measure the interaction between two risers in the sea. It is however a rather unusual test case and we have only measures of the horizontal mean velocity along the axis between the cylinders. The Reynolds number is 1.4 10^5. We use a hybrid turbulence model based on RANS and VMS-LES approaches [17].

As for numerical simulations, we did not find any similar computation in the litterature. In order to capture accurately the interaction between the cylinders, we use a superconvergent approximation of advective terms, able to provide up to fifth order accuracy on a Cartesian mesh. The local mesh between the cylinders is built as Cartesian in most of the interval. The mesh extends in the streamwise direction from $x = -15.$ to $x = 27.$, that is an extension of 42 diameters. The total number of vertices is 12.4 millions.

In order to illustrate the quality of resolution, vorticity contours are depicted in Figure 8. It shows the complex flow features and the very small structures that need to be captured by the numerical model and the turbulence model. The computational domain is decomposed into 192 and 768 subdomains, and as many cores on a Bullx cluster were used to perform these

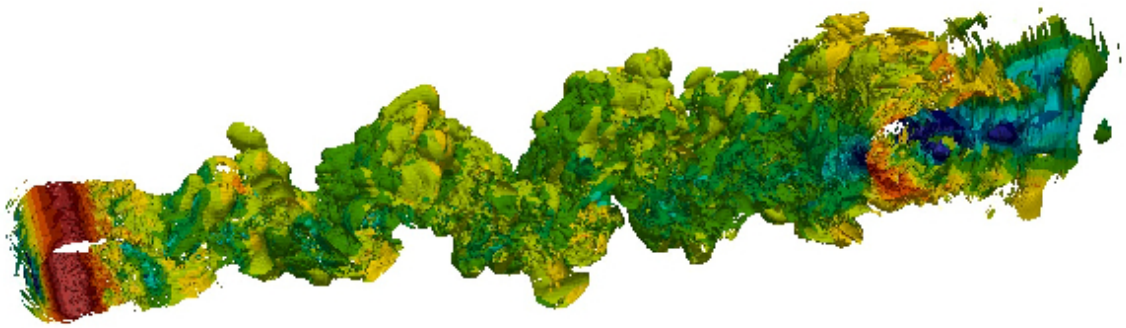


Figure 8: DDES calculation around a tandem cylinders with distance 12D. Velocity contours.

computations. When integer K, used for the definition of the inner and outer zones, is set 20, the percentage of nodes located in the inner zone is 35% (see Table 5). The CPU times for the explicit and MR schemes are shown in Table 5.

5 Conclusion

A new simplified multirate strategy for unstructured finite volume CFD is proposed in this work. The motivation of this research is two folds. *First*, we are interested by increasing the efficiency of accurate unsteady simulations. Indeed, the very high Reynolds number hybrid simulations can be computed with implicit time advancing for maintaining a reasonable cpu. But in many cases this is done with an important degradation of the accuracy with respect to smaller time steps on the same mesh. *Second*, with the arising of novel anisotropic mesh adaptation methods, the complexity of unsteady accurate computations with large and small mesh sizes needs to be mastered with new methods.

The proposed method is based on control volume agglomeration, and relies on:
- a prediction step where large time steps are used and where the fluxes for the smaller elements are evaluated on macro-cells for stability purpose,
- a correction step in which only the smaller elements of the so-called inner zone are advanced in time with a small time step.

K	*nproc.*	Theoretical gain	Measured gain/UP	Measured gain/R
40	288	3.7	1.09	2.11
60	288	3.3	1.11	2.07
40	384	3.7	1.10	2.12
60	384	3.3	1.12	2.07
40	768	3.7	1.35	2.42
60	768	3.3	1.35	2.36

Table 5: **Tandem cylinder with distance 12D**. Time step factor K, number of processors, CPU of the explicit scheme per explicit time-step Δt and per node, percentage of nodes in the inner region, theoretical gain in scalar mode, CPU of the prediction step per time-step $K\Delta t$, CPU of the correction step per time-step $K\Delta t$, and measured parallel gain. UP holds for usual partition, R for optimal radial partition.

An important interest of the method is that the modification effort in an existing explicit unstructured code is very low.

Preliminary interesting results are given. They show that the proposed MR strategy can be applied to complex unsteady CFD problems such as the prediction of three-dimensional flows around bluff bodies with an hybrid RANS/LES turbulence model.

Simulations with Reynolds numbers ranging from 1.66×10^5 to 8.4 millions, representative of problems that can be encountered in industrial applications, are performed. For the considered flow calculations, the fully explicit option is still usable but of high computational cost.

All the numerical experiments are parallel computed with MPI. This allows to identify the main difficulty in obtaining high computational gain, which is related with the parallel efficiency of the computations restricted to the inner zone.

The case of the simplified mesh adaptive calculation of a moving shock is first studied, as a preliminary test for mesh adaptation. The particular case of a mesh of large cells involving, due to delicate mesh generation, a very small number of very small cells is then considered. We observed that the proposed MR strategy offers a superior efficiency in that case.

Thanks to the use of an explicit Runge-Kutta time advancing, the time

accuracy of the MR scheme remains high and the dissipation remains low, as compared with an implicit computation. Only very small time-scales are lost with respect to a pure explicit computation.

In some numerical experiments, CPU gains are obtained with a usual mesh partition. However, the workload should be equally shared for both Step1 (prediction step) and Step2 (correction step) of the MR algorithm. This has been done either with a multi-constrained version of Metis, or, in some case, by hand in an exactly optimal way. In both case, however, communication is increased.

In all the cases, the proposed MR is very stable in practice and the loss of accuracy with respect to an explicit scheme is very low, in contrast to implicit BDF2-based calculations, although we applied the implicit scheme with a CFL of 30, not much larger than with the MR calculation (CFL= 10 for K = 20 in our simulations). Implicit accuracy is limited not only by the intrinsic scheme accuracy but also by the conditions required to achieve greater efficiency which involve a sufficiently large time-step and a short, parameter dependant, convergence of the linear solver performed in the time advancing step. In contrast, explicit and MR computations are parameter safe, and the accuracy of the MR method is optimal in regions complementary to the inner zone, that is, in the case of vortex shedding flow simulations, in regions where it can be necessary to propagate accurately vortices, for example from the first cylinder to the second one in the case of the tandem cylinders.

However, the main output of the experiments is already announced by the complexity analysis: a compulsory condition for a good speedup is that for a $K \geq 10$, the inner zone is less than 10% of the complete domain. In the selected case, the speed up is about 2.

Due to its simplicity, the proposed method can be easily extended to several MR layers corresponding to different time step stability regions in order to separate, for example in the case of three layers, very small scales from intermediate ones, and intermediate scales from larger ones. In a sequel of this work, we plan to combine a higher number of layers with an unsteady mesh-adaptive algorithm.

Acknowledgments

This work has been supported by French National Research Agency (ANR) through "Modèle numérique" program (projet MAIDESC n^0 ANR-13-MONU-0010). This work was granted access to the HPC resources of CINES under the allocations 2017-A0022A05067 and 2017-A0022A06386 made by GENCI (Grand Equipement National de Calcul Intensif).

References

[1] J.F. Andrus. Numerical solution of systems of ordinary differential equations separated into subsystems. *SIAM J. Numerical Analysis*, 16(4), 1979.

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

[3] F. Angrand and A. Dervieux. Some explicit triangular finite element schemes for the Euler equations. *Int. J. for Numer. Methods in Fluids*, 4:749–764, 1984.

[4] A. Belme, A. Dervieux, and F. Alauzet. Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. *Journal of Computational Physics*, 239:6323–6348, 2012.

[5] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64 84, 1989.

[6] E. Constantinescu and A. Sandu. Multirate timestepping methods for hyperbolic conservation laws. *J. sci. Comp*, 33(3):239–278, 2007.

[7] Y. Coudière, B. Palmerio, A. Dervieux, and D. Leservoisier. Accuracy barriers in mesh adaptation. RR-4528, INRIA, 2002. https://hal.inria.fr/inria-00072060/document.

[8] C. Engstler and C. Lubich. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58:173–185, 1997.

[9] C. Engstler and C. Lubich. Mur8: a multirate extension of the eight-order Dormer-Prince method. *Applied Numerical Mathematics*, 25:185–192, 1997.

[10] C.W. Gear and D.R. Wells. Multirate linear multistep methods. *BIT*, 24(4):484–502, 1984.

[11] M.B. Giles. Energy stability analysis of multi-step methods on unstructured meshes. *CFDL Report 87-1, MIT Dept. of Aero. and Astro.*, 1987.

[12] M.B. Giles. Stability analysis of a Galerkin/Runge-Kutta Navier-Stokes discretisation on unstructured tetrahedral grids. *Journal of Computational Physics*, 132(2):201–214, 1997.

[13] A. Guelhan, J. Klevanski, and S. Willems. Experimental study of the dynamic stability of the exomars capsule. *In: Proceedings of 7th European Symposium on Aerothermodynamics. ESA Communications. 7th European Symposium on Aerothermodynamics, 9.-12. Mai 2011, Brugge, Belgium*, 2011.

[14] M. Günther, A. Kvaerno, and P. Rentrop. Multirate partitioned Runge-Kutta methods. *BIT*, 41(3):504–514, 2001.

[15] M. Günther, A. Kvaerno, P. Rentrop, A. Guelhan, J. Klevanski, and S. Willems. Multirate partitioned Runge-Kutta methods. *BIT*, 38(2):101–104, 1998.

[16] M. Günther and P. Rentrop. Multirate row methods and latency of electric circuits. *Applied Numerical Mathematics*, 13:83–102, 1993.

[17] E. Itam, S. Wornom, B. Koobus, B. Sainte-Rose, and A. Dervieux. Simulation of multiple blunt-body flows with a hybrid variational multiscale model. *Conference on Modelling Fluid Flow (CMFF15) The 16th International Conference on Fluid Flow Technologies Budapest, Hungary, September 1-4*, 2015.

[18] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 2006.

[19] R. Kirby. On the convergence of high resolution methods with multiple time scales for hyperbolic laws. *Mathematics of Computation*, 72(243):129–1250, 2002.

[20] B. Koobus, F. Alauzet, and A. Dervieux. Some compressible numerical models for unstructured meshes. In F. Magoulès, editor, *CFD Hanbook*. CRC Press, Boca Raton, London, New York, Washington, 2011.

[21] D. Lockard. Summary of the tandem cylinder solutions from the benchmark problems for airframe noise computations-I. *Proceedings of Workshop AIAA-2011-35*, 2011.

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

[23] C. Moussaed, M.V. Salvetti, S. Wornom, B. Koobus, and A. Dervieux. Simulation of the flow past a circular cylinder in the supercritical regime by blending rans and variational-multiscale les models. *Journal of Fluids and Structures*, 47:114–123, 2014.

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

[25] J. R. Rice. Split Runge-Kutta method for simultaneous equations. *Journal of Research of the National Bureau of Standards - B. Mathematics and Mathematical Physics*, 64B(3), 1960.

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

[27] A. Sandu and E. Constantinescu. Multirate explicit Adams methods for time integration of conservation laws. *J. Sci. Comp*, 38:229–249, 2009.

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

[29] 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.

[30] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

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