NORMA RESEARCH REPORT

T4-D1 (Montpellier+INRIA): (Report M24, 1er Mars 2022) Adaptation of multi-rate schemes to aeroacosutics:

A volume-agglomeration multirate approach.

Florian Miralles(*), Stephen Wornom(*), Bruno Koobus(*), Alain Dervieux(**)

(*) IMAG, Univ. Montpellier, Place E. Bataillon, 34090 Montpellier
(**) Univ. Côte d'Azur, INRIA-Ecuador, B.P.93, 06902 Sophia-Antipolis

Avertissement

La Tâche 4 "Time-advancing algorithmics and parallelism" a pour but d'augmenter l'efficacité des calculs instationnaires combinant LES et acoustique. Nous avons exploré la possibilité d'utiliser une formulation "multirate" permettant de restreindre à une petite région du domaine de calcul l'avancement en temps à petit pas de temps. Le document proposé ici décrit cette approche en CFD compressible sans aller jusqu'à l'extension à la propagation acoustique en milieu hétérogène. En effet, une approche alternative est d'adapter au mieux le pas de temps, notamment lorsque le maillage spatial est adapté. Cette deuxième approche a été préférée pour notre développement d'outils en aéroacosutique et est décrite en T4-D3.

ANR project NORMA

Task 4: Time-advancing algorithmics and parallelism

A volume-agglomeration multirate approach

Florian Miralles*, Stephen Wornom*, Bruno Koobus*, Alain Dervieux**

May 2021

*IMAG, Université de Montpellier, Montpellier, France
florian.miralles@umontpellier.fr, bruno.koobus@umontpellier.fr
** INRIA, Sophia-Antipolis, France
alain.dervieux@inria.fr

Abstract

A 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, multistage 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 this document, we present a multirate scheme based on control volume agglomeration 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 effciency of the multirate method are evaluated on two benchmarks flows: the problem of a moving contact discontinuity (inviscid flow), and the computation with a hybrid turbulence model of a flow around a circular cylinder.

Keywords: Computational fluid dynamics, multirate time advancing, explicit scheme, volume agglomeration, unstructured mesh, compressible Navier-Stokes equations.

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.

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 close to unity). 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.

The 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. 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 apply 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. Multirate time advancing has been studied for PDEs and hyperbolic conservation laws [5, 25, 21, 17, 27, 20], and for a few applications in CFD [20, 27]. Other references are proposed in the annex of this document.

In this document, we present a multirate scheme which is based on control volume agglomeration. This scheme 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 for the mixed finite volume/finite element approximation. 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 multirate algorithm is described in Section 2. Section 3 provides some motivations of

this construction. Section 4 gives some examples of applications. Some concluding remarks are drawn in Section 5, and a short review of multirate works is given in Annex.

2 Multirate time advancing by volume agglomeration

2.1 Finite-Volume Navier-Stokes

The 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. This scheme is implemented in the code Aironum shared by INRIA and University of Montpellier. 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 the quantity vol_i is the volume of cell i, Δt denotes 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.

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

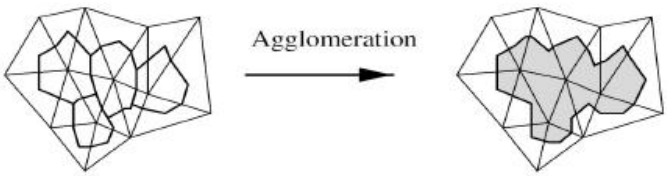
$$\Delta t_i < K\Delta t.$$

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 $\alpha = 1, nstep$

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

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

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

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 [28], 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

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

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

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 $\alpha = 1, nstep$

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

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

  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] [11] [10]. Here we solely propose a L^∞ analysis of the 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.□

The application of an adequate neighboring-cell agglomeration, like in [19] 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

The proposed two-level MR 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 $K \mapsto N^{small}(K) \leq N$ 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

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

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:

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

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:

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

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

Remark 2: Once we have evaluated $K \mapsto N^{small}(K)$ for a given mesh it is possible to predict a theoretical optimum K_{opt} for minimising the CPU time in *scalar* execution. However we shall see that the pertinence of the above theory will be strongly noised by *parallel* implementation conditions.□

3.4 Towards many rates

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 $Grid1$ and $Grid3$,
- prediction on $Grid3$,
- correction on inner part of $Grid1$,

in a three-rate calculation, the correction step is replaced by two corrections:
- prediction on $Grid3$,
- correction on medium part of $Grid2$,
- 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). In contrast to other MR methods, we have a (second) duplication of flux assembly on the inner zone. However, this increment remains limited, since this computation is done $1 + K^{-1} + K^{-2}$ times (111% for $K = 10$).

3.5 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 [6] 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

$$error = O(N_{st}^{-\alpha/4}), \tag{10}$$

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 (10), 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.

3.6 Parallelism

3.6.1 Implementation of MR

The proposed method needs be adapted to massive parallelism. We consider its adaptation to parallel MPI computation relying on mesh partitioning. In a preprocessing phase, the cell-agglomeration is applied at run time inside each partition, which saves communications, and over the whole partition. The motivation is to do it once for the whole computation, while fluctuations of the inner zone at each time level will be taken into account by changing the list of active macro-cells to be agglomerated in inner zone. 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.

3.6.2 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 [27]), 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 useful in some case. It would be compulsory if a strong mesh adaption is combined with the multirate time advancing. However, in the 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 [16] and available in METIS is the *multi constraint partitioning* (MCP) which minimizes the communication cost with multiple constrains. The two constraints for MR are:
- partition is balanced for the whole computational domain, which would be optimal for the prediction step,
- partition is balanced for the inner part of the computational domain, which would be optimal for the correction step.
More precisely, 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 Applications

The MR algorithm is implemented into the parallel (MPI) CFD code AIRONUM shared by INRIA and university of Montpellier. A MUSCL-type upwind-biased finite volume ensuring a third-order spatial accuracy on smooth meshes is used. 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 the first test case, 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 [18] 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 Circular cylinder at very high Reynolds number

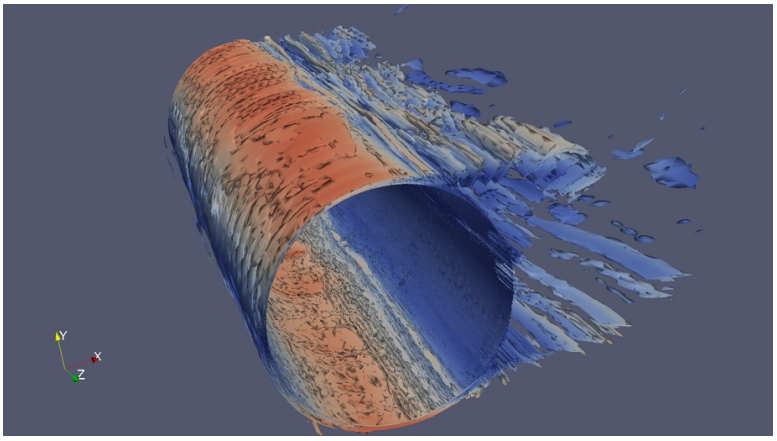


Figure 2: **Circular cylinder at Reynolds number** 8.4×10^6. Instantaneous Q-criterion isosurfaces (coloured with velocity modulus).

The discussion concerning the parallel processing of the multirate scheme will rely on a non-adaptative application. It is the simulation of the flow around a circular cylinder at Reynolds number 8.4×10^6. 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, a hybrid RANS/VMS-LES model is used to compute this flow, which implies that both the fluid and turbulent variables need to be advanced by the time integration scheme, and therefore also the MR method. Figure 2 depicts the Q-criterion isosurfaces and shows the very 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 1). For each simulation, 768 cores were used on a Bullx B720

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 1: **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.

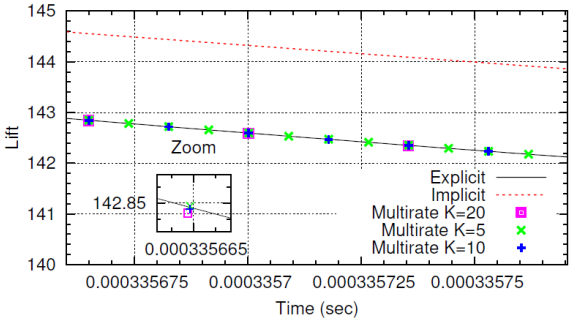


Figure 3: **Circular cylinder at Reynolds number** 8.4×10^6. Zoom of the lift curves obtained with explicit, implicit and MR schemes.

cluster, and the CFL number was set to 0.5. CPU times for the explicit and MR schemes with different values of K are given in Table 1. One can observe that the efficiency of the MR approach is rather moderate, with however a noticeable improvement of the gain in the case of a radial optimal partition. 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 for the usual partition. 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 1 and Figure 3) . This is probably related to the very small scale fluctuations which arise at this Reynolds number (Figure 2). 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 (less than the MR case with the radial optimal partition).

4.2 Mesh adaption for a contact discontinuity

This example is a simplified case of mesh adaptation. 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 4) while velocity and pressure are uniform.

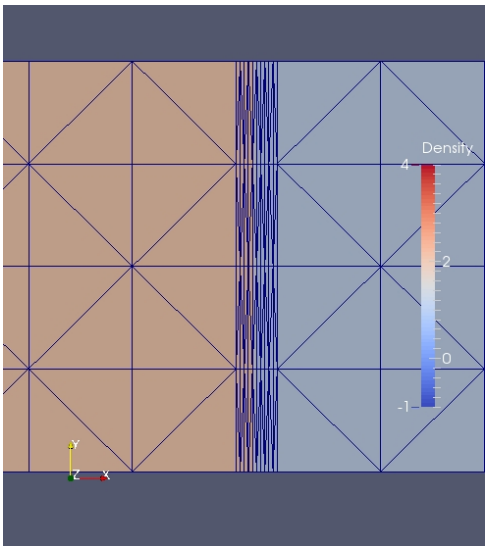


Figure 4: **Mesh adaptative calculation of a traveling contact discontinuity**. Instantaneous mesh with mesh concentration in the middle of zoom and corresponding advected discontinuous fluid density.

The uniform velocity is a purely horizontal one. As can be seen in Figure 4, small cells are present on either side of the discontinuity. The mesh adapts analytically during the computation in such a way that the nodes located at the discontinuity are still the same, and that the number of small cells are equally balanced on either side of the discontinuity. An

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 2: **Mesh adaptative 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.

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 2, CPU times (prediction step / correction step) are given for the MR approach and different time step factors K. The 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.

5 Conclusion

A simplified multirate time advancing strategy for unstructured finite volume CFD is presented in this document. The motivation for using such a multirate approach is two folds. *First*, 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. *Second*, 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.

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.

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 a hybrid RANS/LES turbulence model. A simplified mesh adaptive calculation of a moving shock is also performed, as a preliminary test for mesh adaptation.

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.

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. 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.

6 Annex: a short review of multirate works

A brief review of multirate methods is proposed. It starts from Skelboes's pioneering work using Backward Differentiation Formulas (BDF) [29] to recent works dealing with hyperbolic conservation laws [5, 25, 21, 27].

6.1 Introduction

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

- Multi-method shemes: for systems of EDOs containing both non-stiff and stiff parts, an explicit scheme is used for the non-stiff subsystem and an implicit method for the stiff one [15, 22, 30].
- Multi-order schemes: for non-stiff system of ODEs, the same explicit method and step size are used, but the order of the method is selected according to the activity level (fast/slow) of the considered subsystem of EDOs [7].
- Multirate schemes: for stiff and non-stiff problems, the same explicit or implicit method with the same order is applied to all subsystems, but the step size is chosen according to the activity level. The first multirate time integration algorithm goes back to the work of Rice [23].

In what follows, we focus on the multirate approach. The application of such shemes was first limited to ODEs [23, 1, 9, 29, 24, 2, 14, 8, 7, 13, 12, 26] and restricted to a low number of industrial problems. In the last fifteen years, the developpment and application of such methods to the time integration of PDEs was also performed. In particular, a

few works were conducted on the system of ODEs that arise after semidiscretization of hyperbolic conservation laws [5, 25, 21, 17, 27, 20], and rare applications were performed in Computational Fluid Dynamics (CFD) [27, 20] for which we are interested.

In the following review, a short survey of some important works performed in the domain of multirate approaches is given in Subsection 1.2. It starts from Skelboe's work on multirate BDF methods [29] to recent works dealing with hyperlic conservation laws [5, 25, 21, 27].

6.2 Several works on multirate schemes

6.2.1 The work of Skelboe on multirate BDF methods, 1989 [29].

In this work concerning multirate BDF schemes, first order EDOs, made of a fast subsystem and a slow subsystem, are considered. In the proposed multirate strategy, the fast subsystem is integrated by a k-step BDF formula (BDF-k) with step length h, and the slow subsystem is integrated by the same BDF-k formula but with time step $H = qh$ where q is an integer multiplying factor. Interpolation and extrapolation values (following a Newton type formula) of the solution are used in the proposed algorithms.
As for the application part, a 2×2 test problem is considered for investigating the stability properties of the multirate algorithms. From this application, it appears that the proposed multirate algorithms are not necessarily A-stable, limiting the use of such methods.

6.2.2 The work of Günther and Rentrop on multirate Rosenbrock-Wanner (ROW) methods, 1993 [14].

In this work regarding multirate ROW algorithms, autonomous first order EDOs, which can be split into active and latent components, are considered. The multirate strategy is based on a ROW method in which a large time step H is used for the latent subsystem and a small time step $h = H/m$ is employed for advancing the active components of the solution. In the proposed multirate method, the latent and active components of the solution are extrapolated using a Padé approximation.
The application part concerns the simulation of electric circuits (inverter chain) leading to the solution of stiff EDOs (system of 250-4000 differential equations). A multirate 4-steps ROW method was implemented, leading to a A-stable algorithm, and a speedup up to 2.8 compared to the classical explicit 4-stages Runge Kutta (RK) method.

6.2.3 The work of Löhner-Morgan-Zienkiewicz on explicit multirate schemes for hyperbolic problems with CFD applications, 1984 [20].

To our knowledge, this work is the first one on multirate methods which deals with applications in CFD. In the model problem, two subregions with different grid resolution are considered and the solution is advanced in time using a given explicit scheme. A large time step Δt_1 is used in the coarse subregion Ω_1, and a small time step $\Delta t_2 = \Delta t_1/n$ is employed in the fine subregion Ω_2. In the proposed multirate strategy, some grid points belonging to Ω_1 are added to the fine subregion Ω_2 (which becomes Ω_2'), and appropriate boundary conditions are used for both subregions Ω_1 and Ω_2', together with mean values at some points belonging to Ω_1 and Ω_2'. The proposed multirate scheme was implemented with a second order explicit finite element scheme (Taylor-Galerkin method of Donea).
Several CFD test cases were considered, which illustrate that shocks can be handled by the multirate method and that a speedup of 2 between the multirate scheme and its single-rate counterpart can be obtained.

6.2.4 The work of Kirby on a multirate forward Euler scheme for hyperbolic conservation laws, 2002 [17]

This theoritical work presents and analyzes a multirate method for the solution of one dimensional hyberbolic conservation laws. After semi-discretization by a finite volume MUSCL approach, a system of EDOs is obtained which is partitioned in fast and slow subsystems. A rather simple multirate scheme based on forward Euler steps is proposed for the solution of these subsystems. The fast subsystem is advanced with a small time step $\Delta t/m$, while a large time step Δt is used for the slow subsystem. No extrapolation/interpolation are performed in the proposed multirate strategy. It is shown that the multirate scheme satisfies the TVD property and a maximum principle under local CFL conditions, but it is only first order time accurate.

6.2.5 The work of Constantinescu and Sandu on multirate RK methods for hyperbolic conservation laws, 2007 [5]

One-dimensional scalar hyperbolic equations are considered in this study. For the solution of these equations, a second-order accurate multirate scheme, that inherits stability properties of the single rate integrator (maximum principle, TVD, TVB, monotocity-preservation, positivity), is developped. After a semi-discrete finite volume approximation (which satisfies some of the above stability properties), a system of EDOs is obtained and partionned into slow and fast subsystems. The computational domain is split into a subdomain Ω_F corresponding to a fast characteristic time where a small time step $\Delta t/m$ is used in the multirate scheme and a subdomain Ω_S with a slow characteristic time where a large time step Δt is employed. Furthermore, a buffer zone between Ω_F and Ω_S is introduced in order to bridge the transition between these two subdomains for the purpose that the multirate scheme satisfies the stability properties of the single rate scheme. In the proposed multirate method, appropriate explicit RK schemes are also used in each subdomain and the buffer zone.
The resulting multirate algorithm is assessed on 1D problems (advection equation, and Burger's equation) with fixed and moving grids. It was checked that the numerical solutions are second order accurate, positive, obey the maximum principle, TVD, wiggle free, and the integration is conservative. Speedups up to 2.4 were obtained.

6.2.6 The work of Seny *et al.* on the parallel implementation of multirate RK methods, 2014 [27]

The work of Seny *et al.* focuses on the efficient parallel implementation of explicit multirate RK schemes in the framework of discontinuous Galerkin methods. The multirate RK scheme used is the approach proposed by Constantinescu [5] and introduced in the previous subsection.
In order to optimize the parallel efficiency of the multirate scheme, they propose a solution based on multi-constraint mesh partitioning. The objective is to ensure that the workload, for each stage of the multirate algorithm, is almost equally shared by each computer core i.e. the same number of elements are active on each core, while minimizing inter-processor communications. The METIS software is used for the mesh decomposition, and the parallel programming is performed with the Message Passing Interface.
The efficiency of the parallel multirate strategy is evaluated on two- and three-dimensional CFD problems. It is shown that the multi-constraint partitioning strategy increases the efficiency of the parallel multirate scheme compared to the classical single-constraint partitioning. However, they observe that strong scalability is achieved with more difficulty with

the multirate algorithm than with its single rate counterpart, especially when the number of processors becomes important compared to the number of mesh elements. The possible low number of elements per multirate group and per processor is a limiting factor for the proposed approach.

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

[6] Y. Coudière, B. Palmerio, A. Dervieux, and D. Leservoisier. Accuracy barriers in mesh adaptation. RR-4528, INRIA, 2002.

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

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

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

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

[11] 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.

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

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

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

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

[16] 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.

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

[18] 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.

[19] M.H. Lallemand, H. Steve, and A. Dervieux. Unstructured multigridding by volume agglomeration: current status. *Comput. Fluids*, 21:397–433, 1992.

[20] 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.

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

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

[23] 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.

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

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

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

[27] 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.

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

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

[30] R. Weiner, M. Arnold, P. Rentrop, and K. Strehmel. Partioning strategies in Runge-Kutta type methods. *IMA J. of Numer. Analysis*, 13:303–319, 1993.