

MPMD parallelization of an aerodynamic code with bodies in relative motion

Jean-Marie Couteyen^{‡,†}, Grégoire Pont[‡], Jean Roman[†], and Pierre Brenner[‡]

[‡]Airbus Defence and Space, 51-61 Route de Verneuil, 78130 Les Mureaux, France

[†]Inria Bordeaux - Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence, France

FLUSEPA is an advanced simulation tool which performs a large panel of aerodynamic studies. It is the unstructured finite-volume solver developed by Airbus Defence & Space company to calculate compressible, multidimensional, unsteady, viscous and reactive flows around bodies in relative motion [1]. The numerical strategy in FLUSEPA is designed for highly compressible flows and keeps its accuracy in non-Cartesian grids. According the desired accuracy, a second-order accurate shock-capturing scheme is generally used for RANS and URANS simulations and a fourth order accurate vortex-centered scheme is used for hybrid RANS/LES simulations [3]. The meshing strategy is based on multi-overlapping grid intersection which is conservative and allows to quickly and properly mesh 3D complex geometries. It can be seen as a CHIMERA strategy without interpolation. Each body is meshed independently and immersed in background grids.

Computation time is mainly shared between the aerodynamic solver and intersection computations but the ratio between the two of them varies. To take into account motion during aerodynamic computations, loads are gathered around the bodies to compute a cinematic. In order to limit the number of intersection computations, we also integrate the volume displacement between cells. Up to a certain threshold, the evolution of the cell volume can be evaluate practically with a good level of accuracy. After that, a new intersection must be computed.

Because aerodynamics and intersection are loosely coupled, we have chosen a MPMD approach with specialized processes : one master process which is in charge of coordinating the simulation and several aerodynamic and intersection processes. Communications between processes are done using MPI and each process is multi-threaded by using OpenMP. In order to be able to perform asynchronously intersection and aerodynamic computations, the cinematic is extrapolated. Intersection processes compute a future position and this position is given to aerodynamic processes at the right time. If the cinematic extrapolation is not satisfactory, a new intersection is computed as a fail-safe measure.

The most challenging part of the parallelization is the aerodynamic solver because of the way the time is integrated. The time integration in FLUSEPA is done using an explicit temporal adaptive method [2]. Instead of using a time step imposed by the slowest cell, cells are grouped inside temporal classes according to their maximum allowed time step. This method allows to perform a fewer number of operations at the expense of some complexity. An iteration of the aerodynamic solver is separated into several sub-iterations. Every cell is not integrated during each sub-iteration but cells which share a face with a slower one are interpolated. At the end of the iteration, each cell has reached the same time. Between iterations, temporal classes can evolve. To parallelize the solver, a classical domain decomposition is used which takes into account the temporal level of each cell.

The new version of this code is used to compute in a reasonable time RANS and URANS simulations on the whole launcher, blast wave simulation during the lift-off on the full size launch pad and stage separation (see Figure 1). Hybrid RANS/LES computations are also performed to get more details about the unsteady behaviour of the flow, more precisely the buffeting on the shrinkage of the launcher (see Figure 2).

Performance analysis were performed in order to highlight the bottlenecks, notably on an Ariane 5 lift-off and on Ariane 5 EAP separation simulations. Despite asynchronous communications with computation/communication overlapping, it appears that reducing synchronizations in the aerodynamic solver is

a critical point. In the other hand, the intersection computation brings two major problems : the intersection/aerodynamic computation ratio evolves during time and the application of a newly computed intersection is costly. Expressing more precisely the parallelism by a fine grain task graph approach and exploiting it by a runtime system is currently under evaluation to overcome these issues.

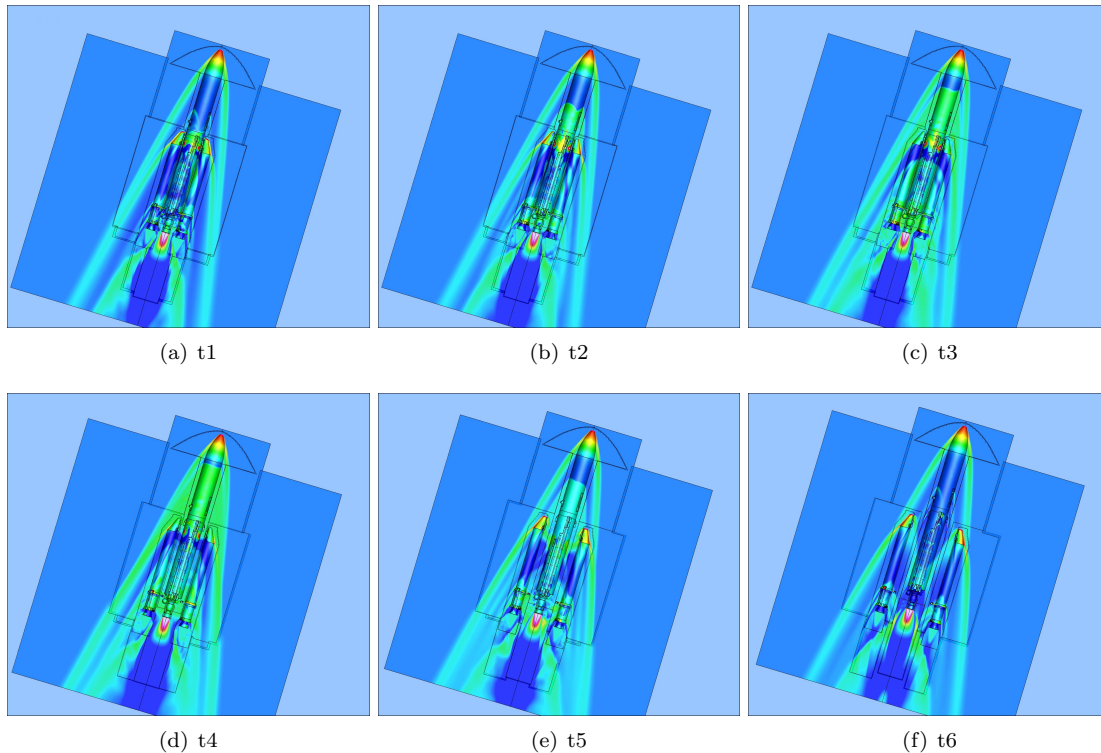


Figure 1. Ariane 5 booster separation after distancing rocket ignition, colored by pressure.

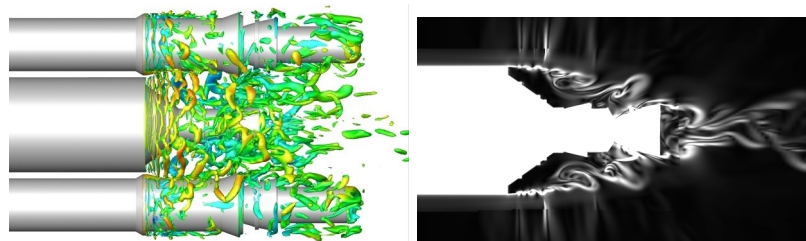


Figure 2. Instantaneous flow around Ariane 5 launcher: Q criterion colored by instantaneous velocity, density gradient.

References

- [1] P. Brenner, J-M. Carrat, and M. Pollet. “Simulation d’interactions aérodynamiques instationnaires autour de plusieurs corps en mouvement relatif”. In: *AAAF : Les interactions en aérodynamique*. Marseille, 1998.
- [2] W. L. Kleb, J. T. Batina, and M. H. Williams. “Temporal adaptive Euler/Navier-Stokes algorithm involving unstructured dynamic meshes”. In: *AIAA Journal* Vol. 30 (1992), pp. 1980–1985.
- [3] G. Pont, P. Cinnella, J-C. Robinet, and P. Brenner. “Development of Numerical Schemes for Hybrid Turbulence Modelling in an Industrial CFD Solver”. In: *21st AIAA Computational Fluid Dynamics Conference*. San Diego, 2013.