MPMD parallelization of an aerodynamic code with bodies in relative motion

JM. Couteyen, J. Roman, P. Brenner

July 16, 2015

Outlines

AIRBUS DEFENCE & SPACE

Inria INVENTEURS DU MONDE NUMÉRIQUE

Outlines

Aerodynamics and CFD at Airbus DS

Test and simulations since 80'

Specific problems:

- Stage separations
- Takeoff Blast wave
- Dynamic stability at reentry

FLUSEPA

About

- Navier-Stokes code developed for more than 20 years
- Development choices driven by unsteady simulations
- No competitive COTS software available
- 1999 "Science & Défense" Award (Pollet-Brenner)

Characteristics

- Unstructured meshes
- Changing topology (bodies in relative motion)
- Geometric intersections (CHIMERA-like)
- Temporal adaptive time integration scheme

AIRBUS DEFENCE & SPACE

Inria INVENTEURS DU MONDE NUMÉRIQUE

FLUSEPA

Architecture compatibility

- 1987 : SIMD / Vector computers
- 1999 : SMP (using OpenMP)
- 2012 : Starting work for exploiting Clusters (using MPI and still OpenMP)
- 2014 : Starting work on a task-based version

AIRBUS DEFENCE & SPACE

Inria INVENTEURS DU MONDE NUMÉRIQUE

Outlines

General scheme

Algorithm 1 General iteration

```
1: Aerodynamic Solver
2: Current_kinematic+=Kinematic Computation
3: if Important motion since last intersection then
4:     Body Displacement(Current_kinematic)
5:     Intersection Computation
6:     Current_kinematic=0
7: end if
```

Current Production Version

MPMD, OpenMP/MPI

- Different specialized processes
  - Master (Kinematic Computation,I/O, Mesh Partitioning, Load Balancing)
  - Aerodynamic Solver
  - Mesh intersection (bodies in relative motion)
- # of processes defined at start (but can be changed when the computation is restarted)

Parallelization issue : Temporal Adaptive Solver

- Explicit solver competitive for unsteady phenomenon

Temporal adaptive solver principles

- Compute each cell near its maximum physical time step
- Interpolation for boundaries between temporal levels

Difficulties

- Load Balancing
- Temporal zones evolve during computation
- A lot of synchronizations may be implied by the implementation of the algorithm

AIRBUS DEFENCE & SPACE

Inria INVENTEURS DU MONDE NUMÉRIQUE

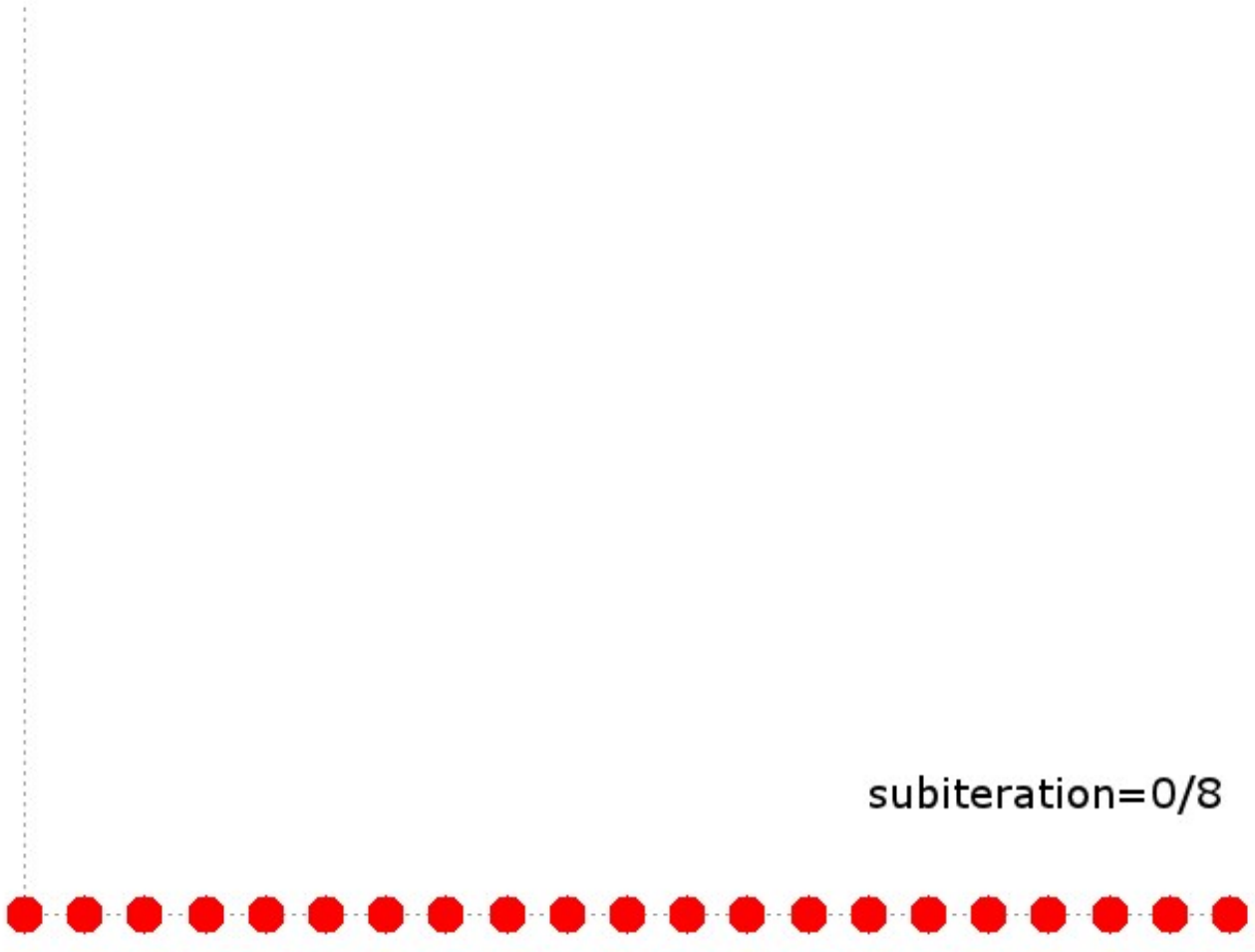
Temporal Adaptive Solver

Algorithm 2 Temporal Adaptive in Flusepa, one general iteration of the aerodynamic solver

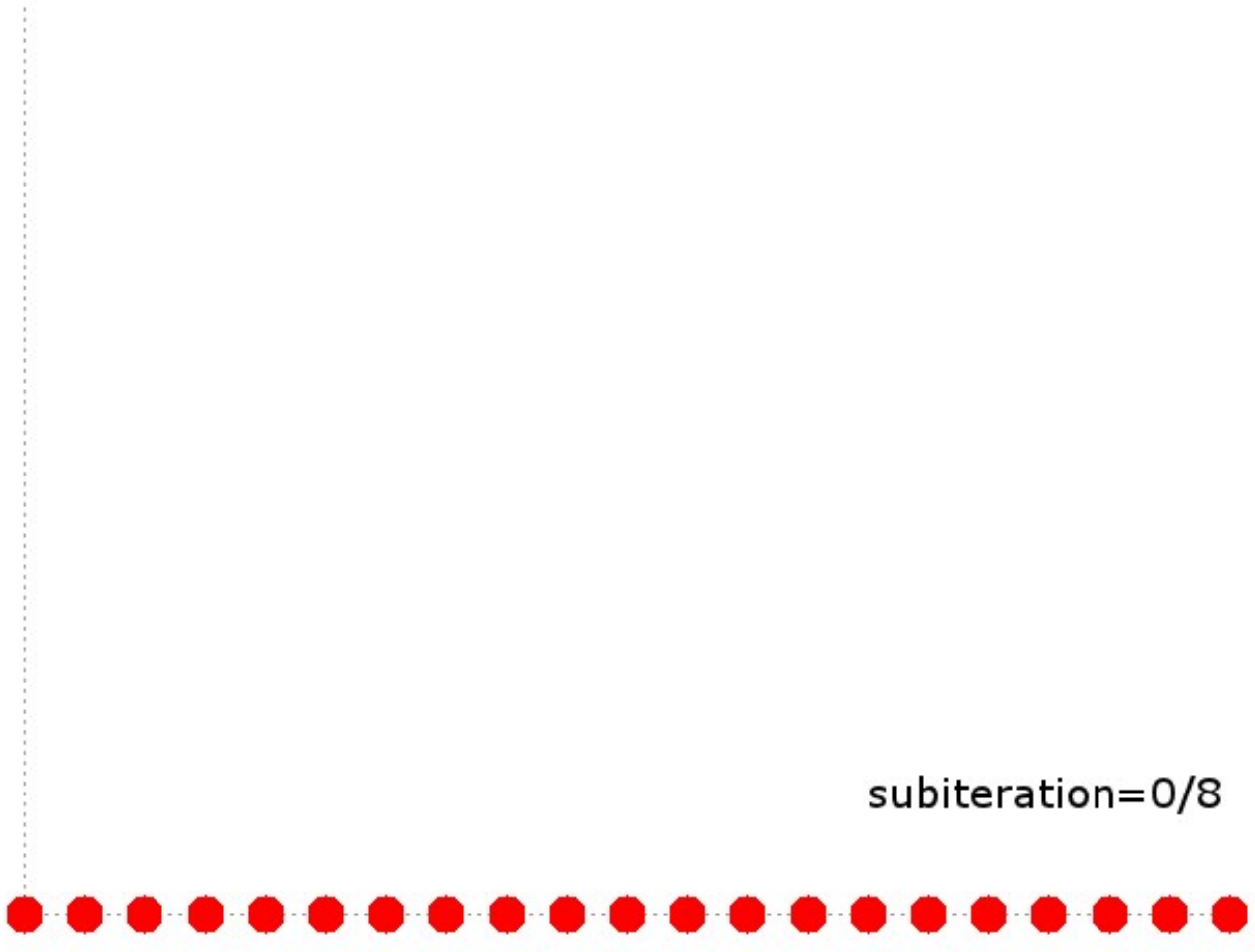
```
1:  Timestep computation for every cell
2:  Classification of every cell inside a temporal class.
3:  Temporal adapative loop:
4:  for subiteration=1 to 2^θ + 1 do
5:      τ = 0
6:      for tmp = 1 to θ + 1 do
7:          if (mod(subiteration − 1, 2^tmp) == 0) then
8:              τ = tmp
9:          end if
10:     end for
11:     Predictor (0 to τ)
12:     if τ ≠ θ then
13:         Intensive repositionning of τ + 1 temporal class
14:     end if
15:     for τ' = τ to 0 do
16:         Corrector
17:     end for
18: end for
```

τ : temporal class currently integrated
θ : higher temporal class

[2,2,2,1,1,1,1,0,1,1,1,1,2,2,2,2,2,2,2,2,2]



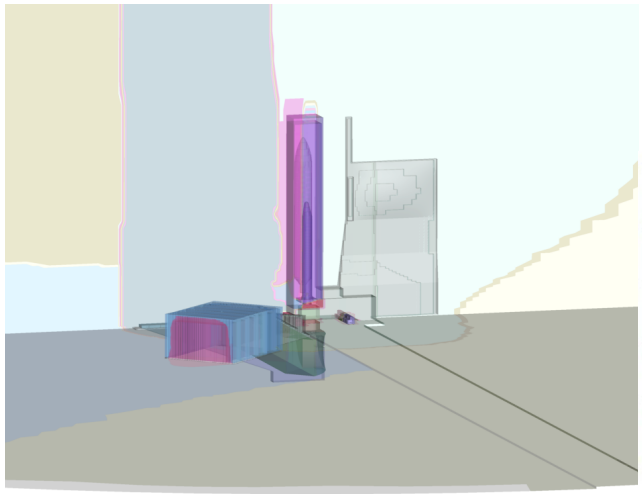
[2,2,2,1,1,1,1,0,1,1,1,1,2,2,2,2,2,2,2,2,2]



Aerodynamic solver parallelization

Domain Decomposition

Partitioning the mesh in several parts, each associated to a different process



P	A	f
Q	B	g
R	C	h
S	D	i
T	E	j

Domain 1

A	f	k
B	g	l
C	h	m
D	i	n
E	j	o

Domain 2

Ghost Cells

A cell belong to one and only one domain, but at the borders cells are replicated (Border faces are duplicated)

Aerodynamic solver parallelization

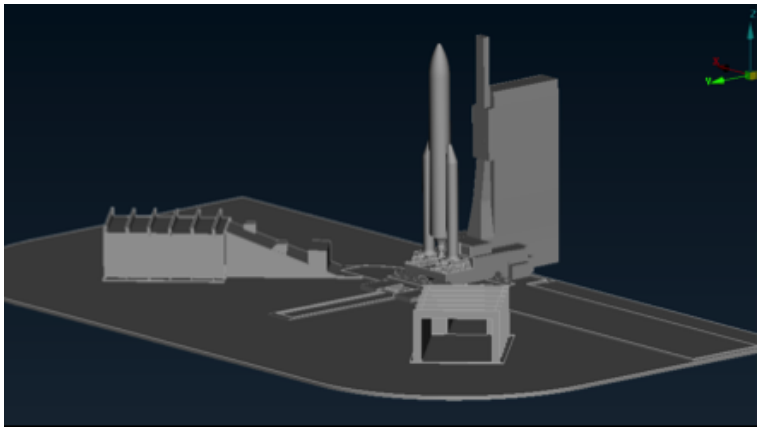
Asynchronous Point-to-Point Communications

```
computation_part1()
foreach neighbour:
    Isend(border_cells)
    Irecv(ghost_cells)
computation_part2()
foreach neighbour:
    Wait(border_cells) #Isend
    Wait(ghost_cells) #Irecv
computation_part3()
```

- Communication/Computation Overlap
- Communication only when sharing a border of the given temporal level with neighbours

Blast wave on the full scale launch pad

- 20.7 million cells with overlapped grids
- Objective: simulate the overpressure levels due to the boosters ignition
- Validation: Comparison with flight sensors



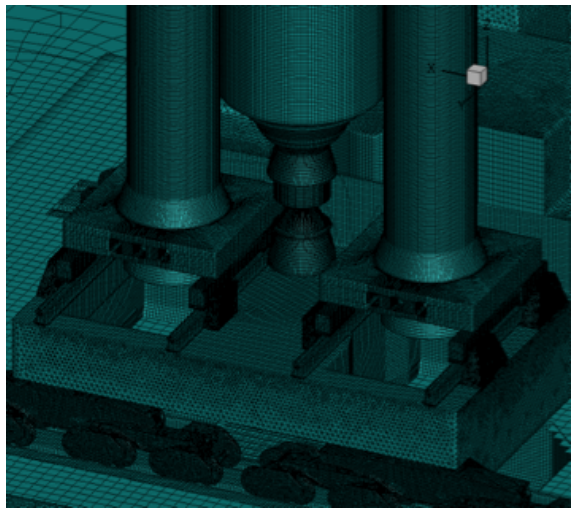


Figure: Geometry and grid of the full scale launch pad

Hypothesis about physics

- No water injection in ducts
- No after-burning phenomenon (which may occurs at the end on the nozzle)

Qualitative results

Blast wave phenomenology

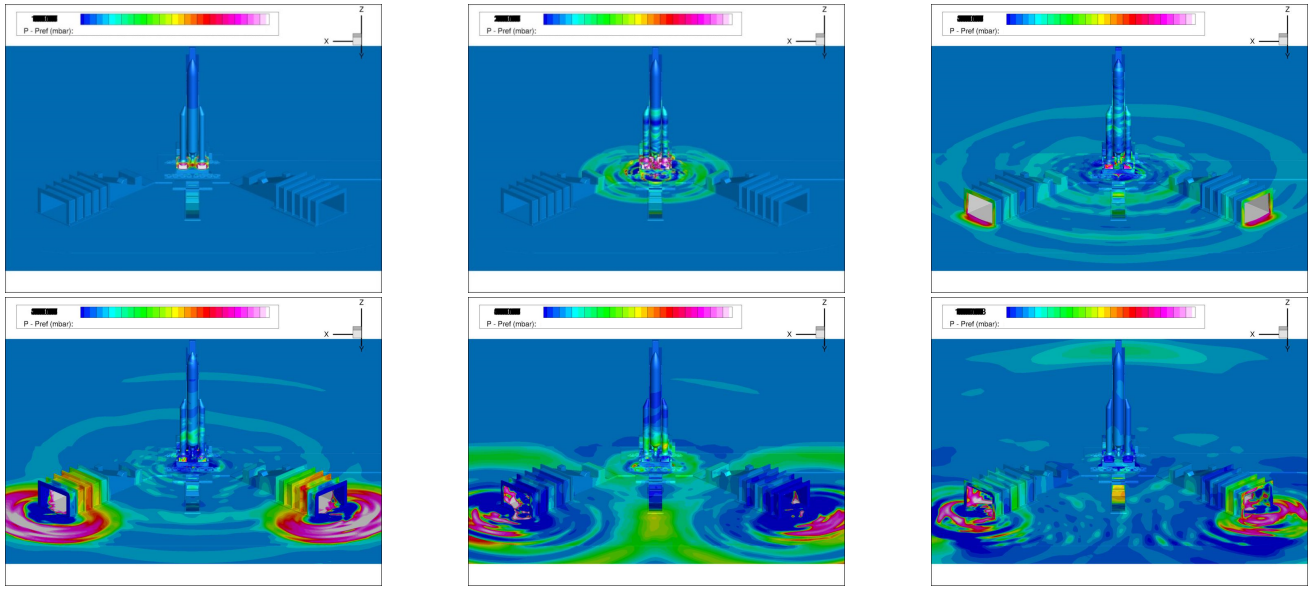


Figure: Blast wave phenomenology, pressure contours

- Strong overpressure wave from nozzle exit, called IOP (Ignition overpressure)
- Transmission inside the ducts, second overpressure wave is generated, called DOP (Duct overpressure)
- Overpressure levels $\rightarrow$ aerodynamic loads on the launcher

Quantitative results

- Good prediction of overpressure levels and loads applied on the launcher

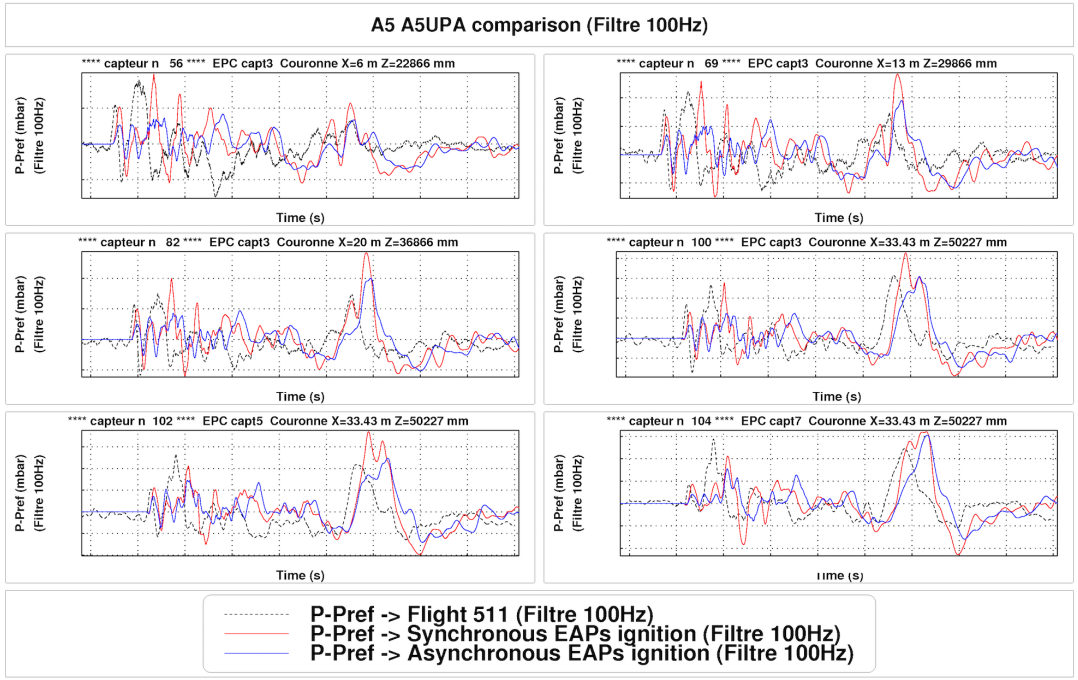
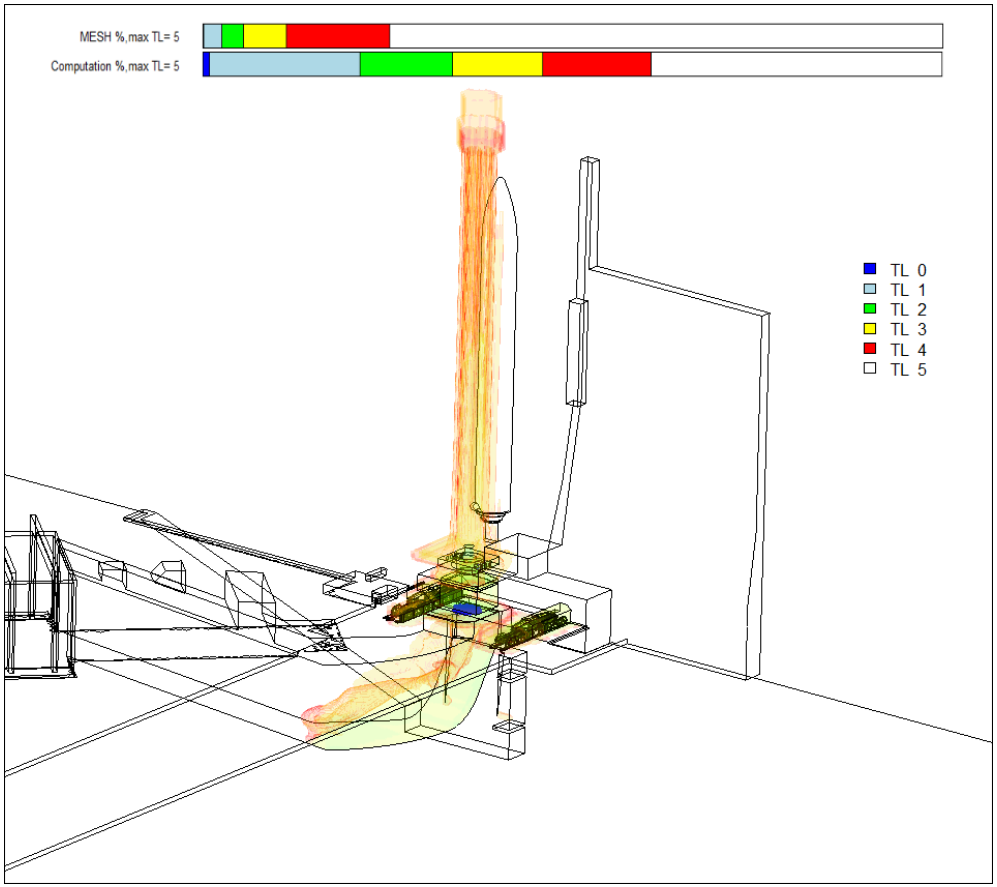


Figure: Overpressure levels on sensors

To go further and reduce the modeling and numerical errors

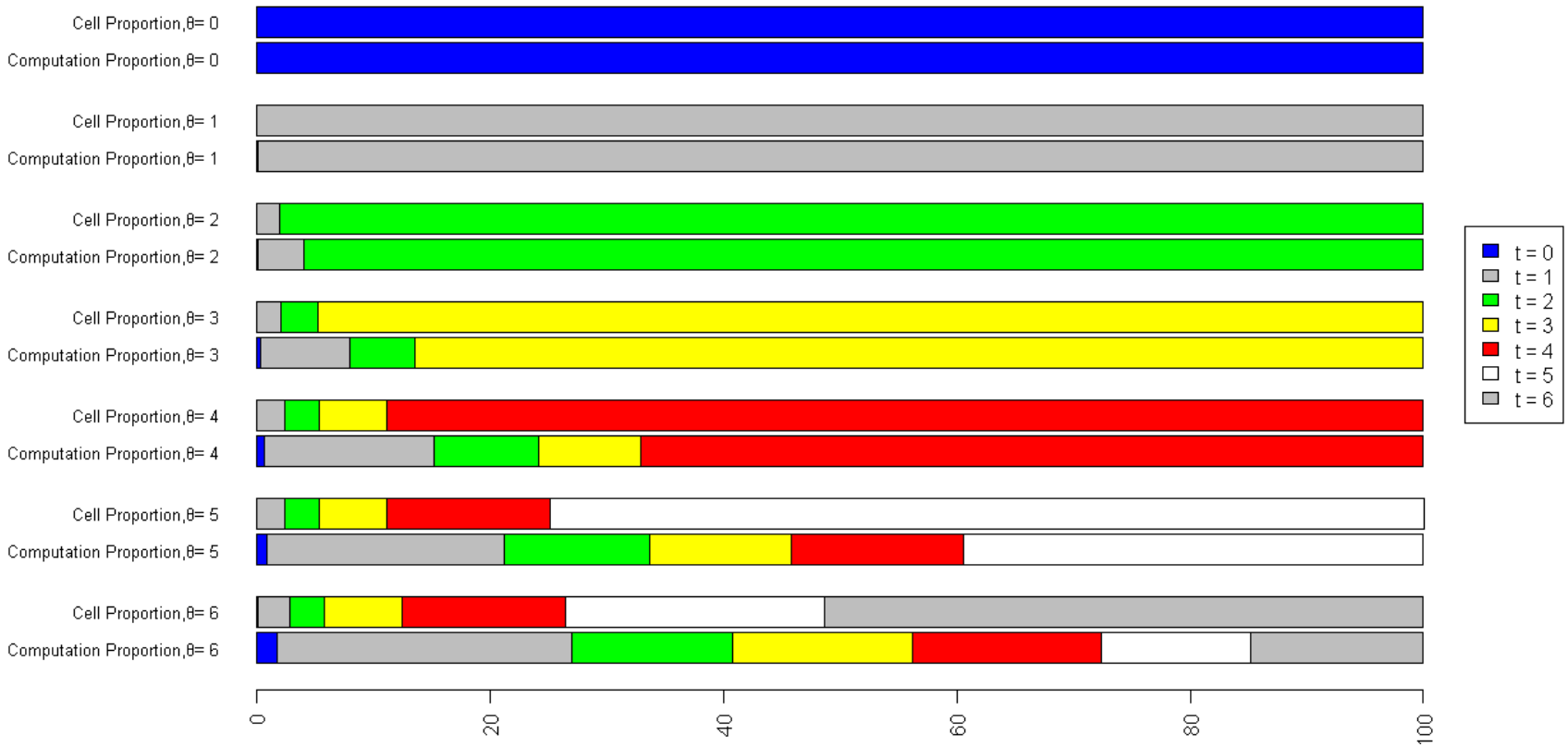
- Modeling errors: more details about the inputs are needed: a realistic model for the ignition and water injection (the size of the drops for example)
- Numerical errors: using the third order numerical scheme may reduce the phase error

Performance Analysis



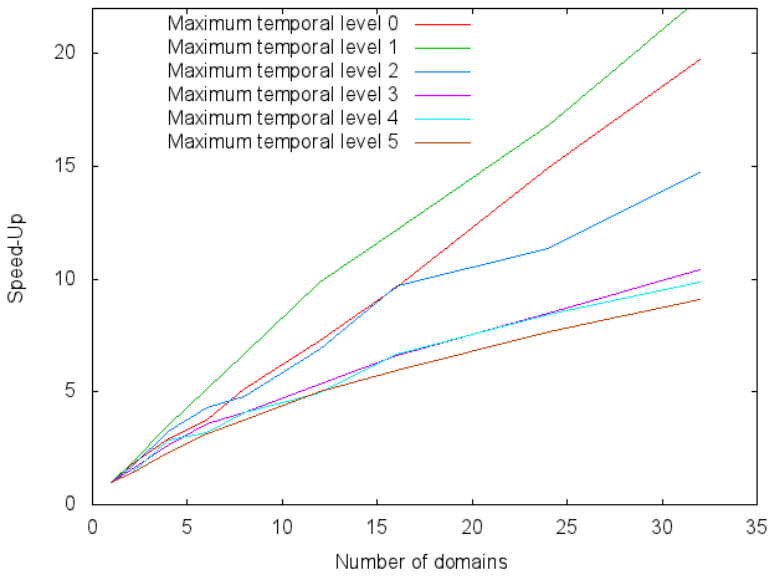
Location of cells according to their temporal level (5 temporal levels)

Performance Analysis

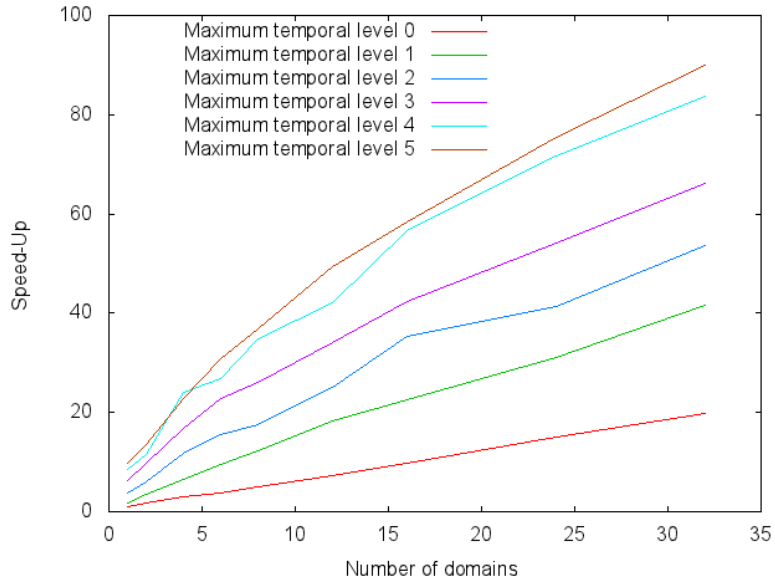


Proportion of cells of each temporal level and the associated computation proportion with different maximum temporal level

Even with low scalability, Temporal Adaptive is competitive



Scalability with 1 domain as reference for each curve



Speed-up with global time step for one domain as reference

Asynchronous intersections

Motions and intersections

When bodies are in relative motion...

- Actual volume is estimated via a volume flow computation, when relative variation is lower than a threshold
- When this variation is more important, an intersection is computed

Independent with aerodynamic computations

Topology evolves during computation

Asynchronous computation of intersections

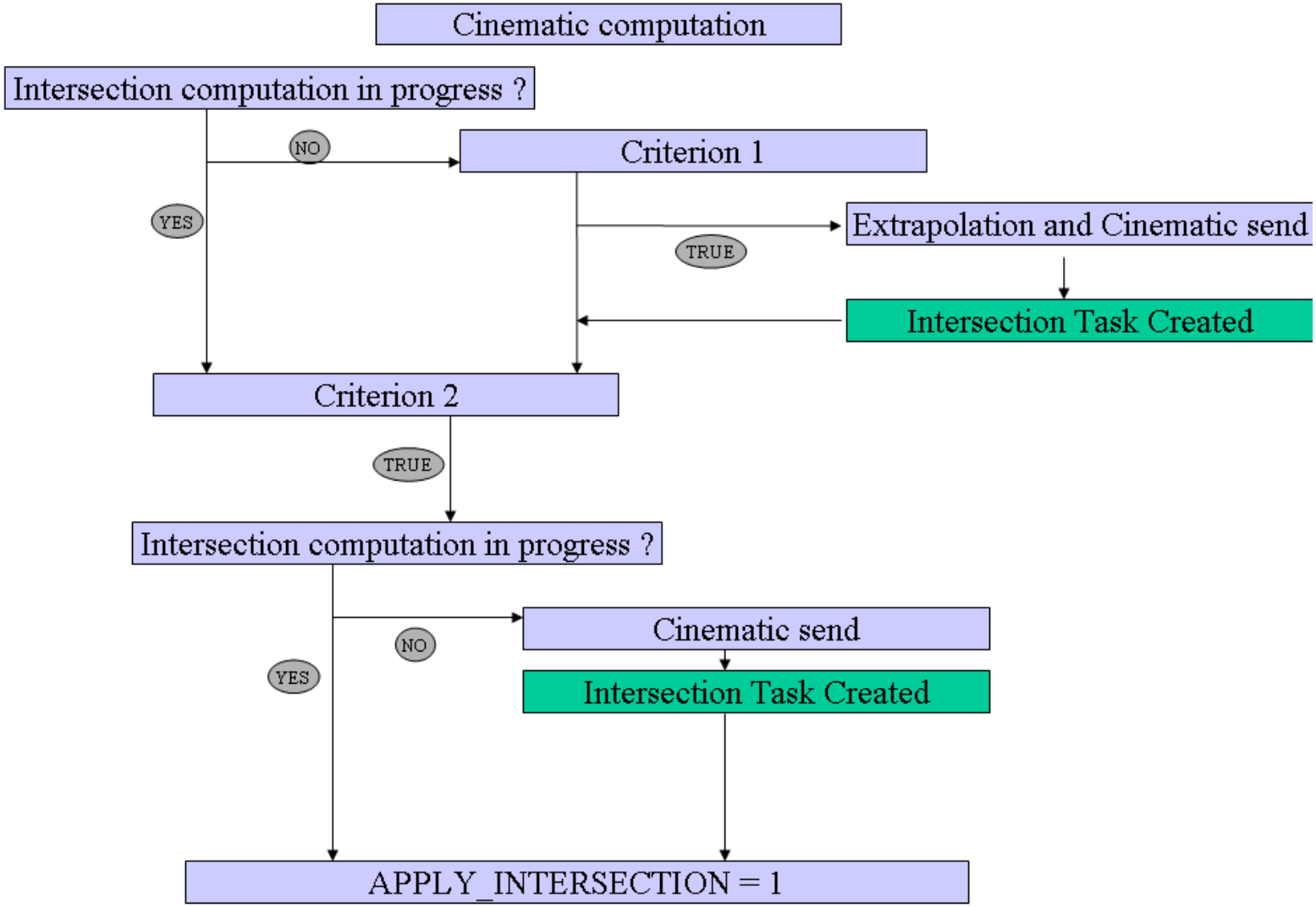
Start intersection computation as soon as possible

- Cinematic extrapolation
- Intersection computation is performed concurrently with aerodynamic computations

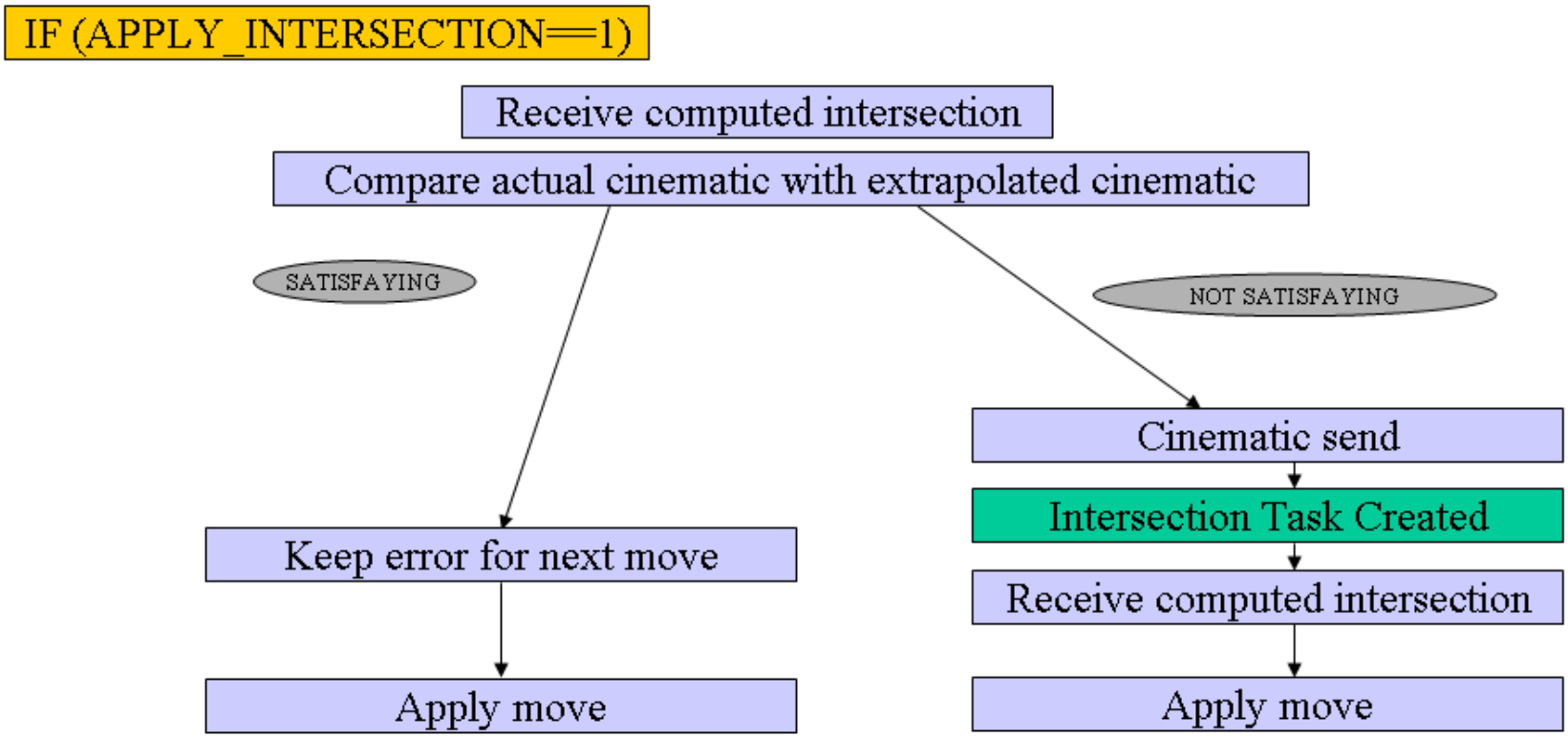
Intersection applied when needed

- The new topology is passed to the Master...
- ...and given to the Aero processes

Asynchronous intersection : launching computation



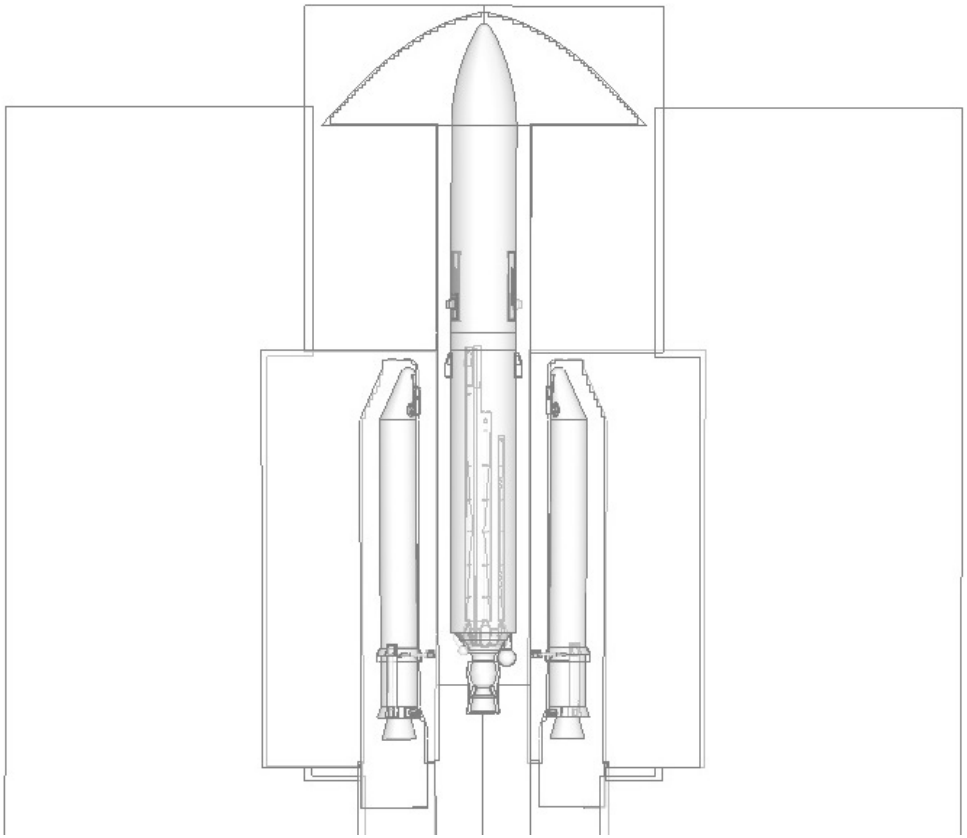
Asynchronous intersection : applying intersection

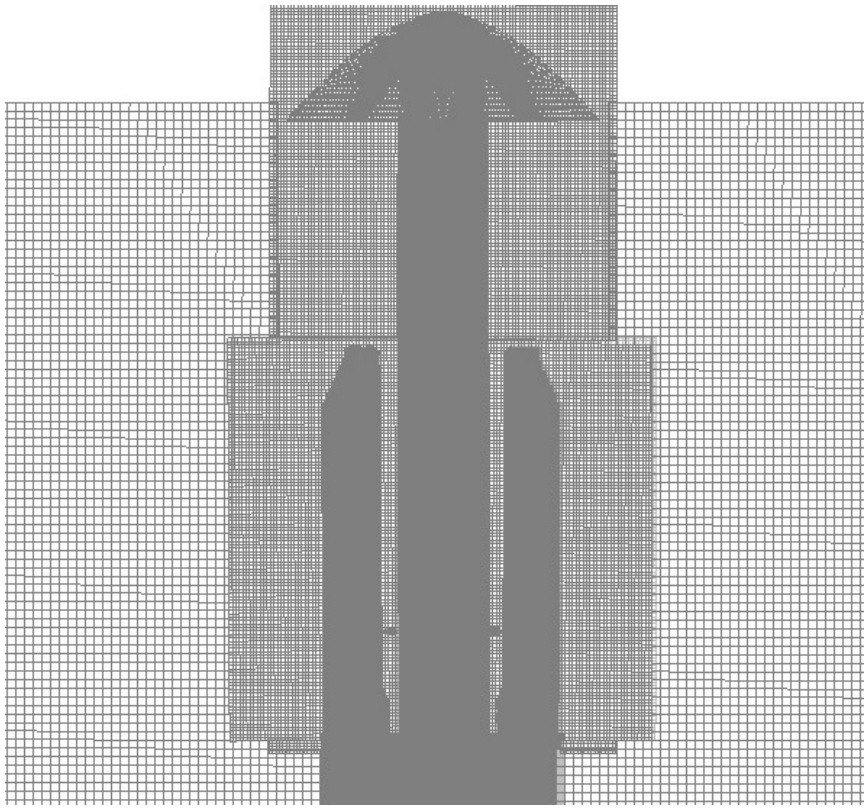


2 step asynchronous process

- Extrapolation of the cinematic to pre-compute an intersection
- Application of the intersection with a correction if "good enough"

A5 booster separation





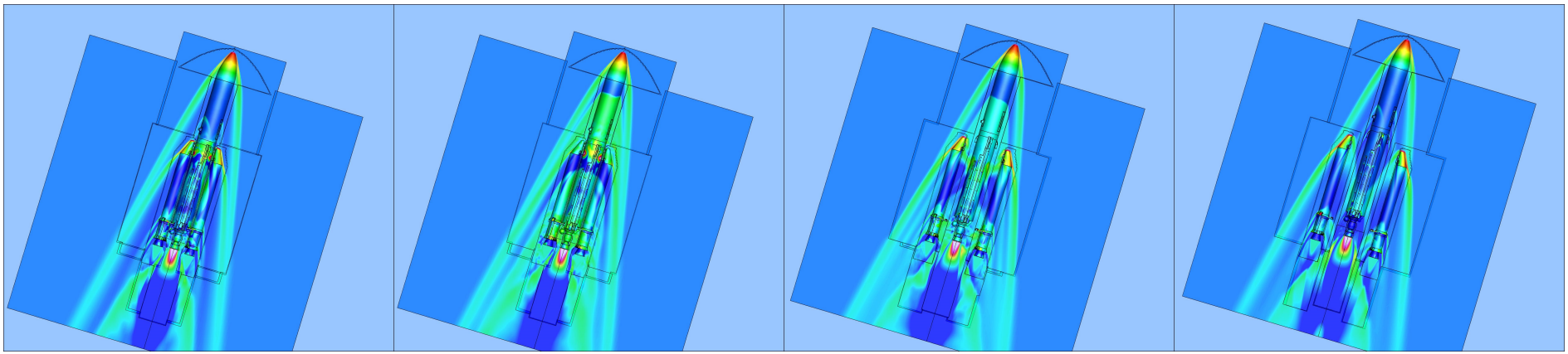
Multiples meshes, intersection, motion

- Multiple meshes are used. Each booster and the main stage are meshed independently
- Motion implied by aerodynamics and input data (distancing rocket ignition)

AIRBUS DEFENCE & SPACE

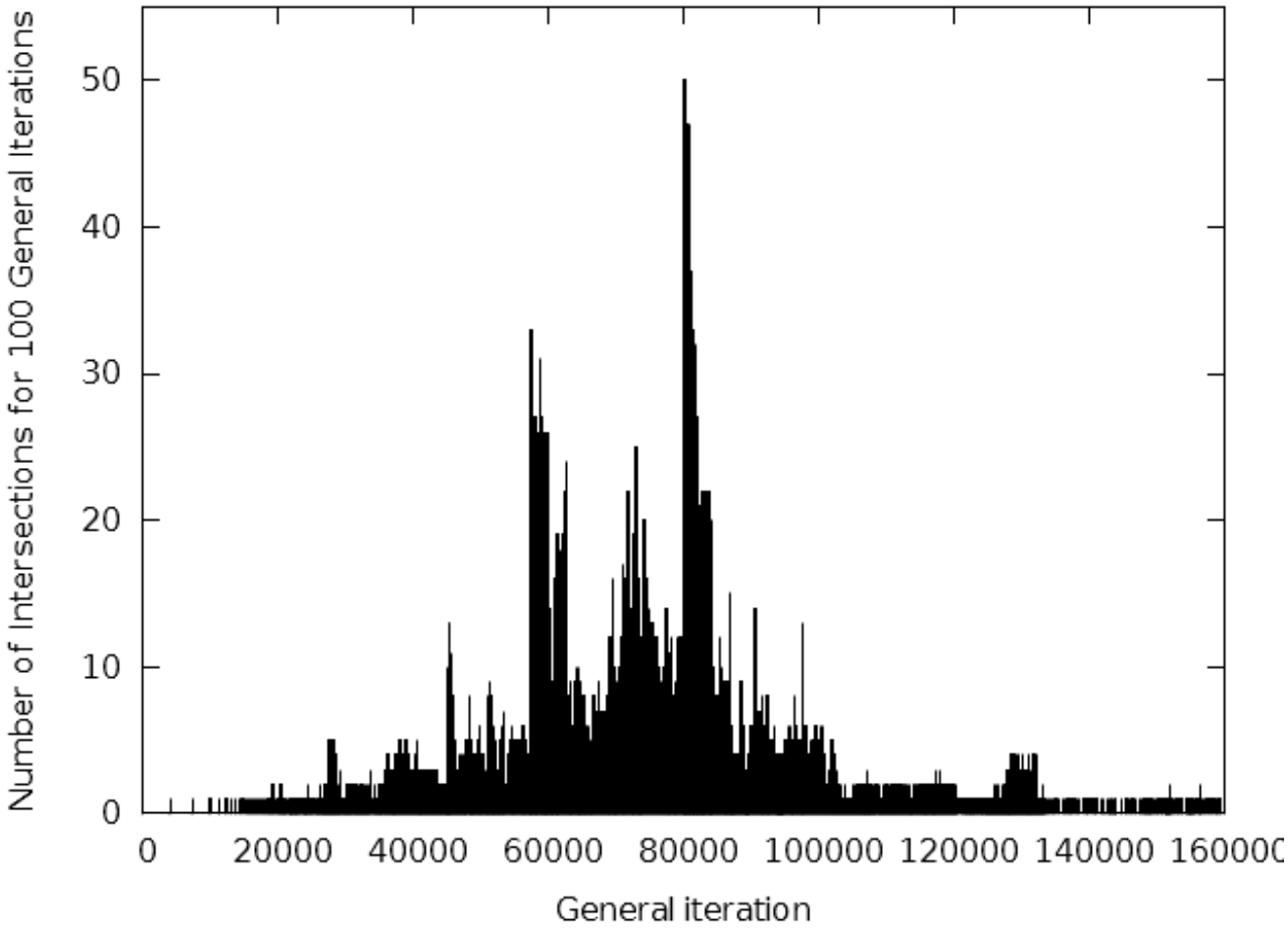
Inria INVENTEURS DU MONDE NUMÉRIQUE

A5 booster separation



The simulation of stage separation allows to calculate the relative motion between the different stages (check that there is no collision)

A5 booster separation



The computation ratio between intersection and aerodynamics evolves during the computation

Conclusion and current works

Limits

Master

- Some data are centralized : memory problem
- I/O bottleneck

Aerodynamic solver

- Synchronizations

Intersection

- The load varies during computation in an unpredictable way
- New topology delivery time to aerodynamic processes is not negligible
- Static allocation of process implies a potential under use of computational resources

Response to the limits

Aerodynamic solver

- Synchronization problem : the current algorithm implies a lot of synchronizations in distributed memory while actual dependencies are local
- Working on subdomains in shared memory could lead to a more flexible way of overlapping computation and communication

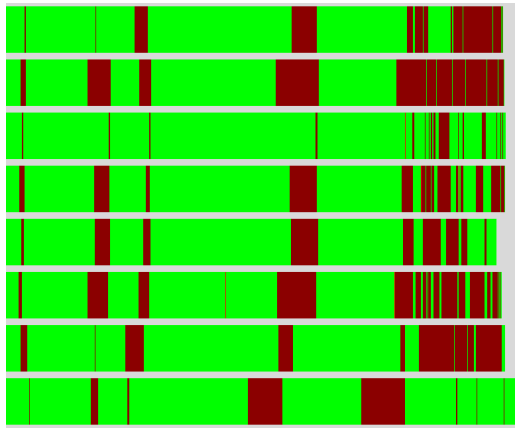
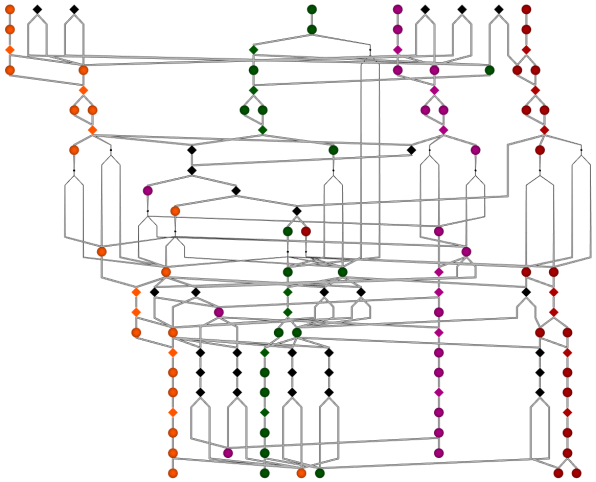


Figure: Gantt diagram showing activity of processes

Current work : Task version



Aerodynamic solver with tasks

- Task are created according to the smaller domains
- Multiple travels of the DAG are possible, which may lead to a better computation/communication overlap

Full task implementation benefits

- With a complete description of the problem in tasks, intersection and aerodynamic computations could happen in the same process, cancelling the need to deliver the topology
- Other optimization will be possible, like pipelining solver iterations

AIRBUS DEFENCE & SPACE

Inria INVENTEURS DU MONDE NUMÉRIQUE

Thank you for your attention.
Any questions?