Dynamic mesh adaptation for moving fronts and interfaces: application to the modeling of premixed flames and primary atomization

V. Moureau, P. Bénard, G. Lartigue
CORIA, CNRS UMR6614, Normandie Univ, UNIROUEN, INSA Rouen

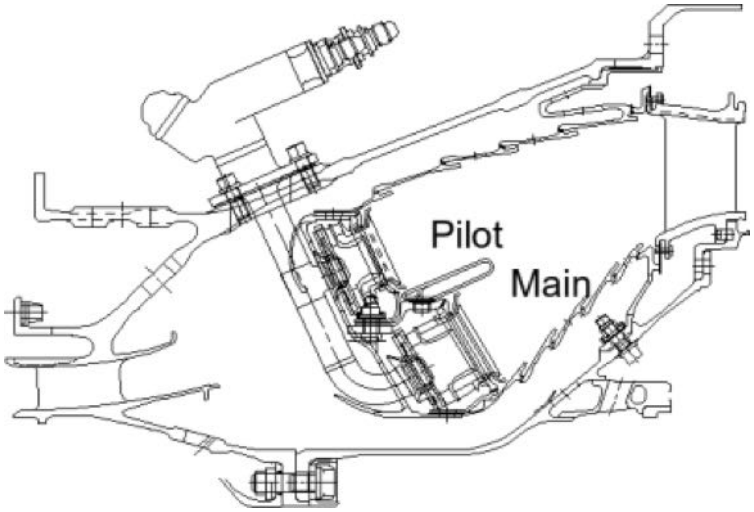
R. Mercier, M. Cailler – SAFRAN Tech, Magny-les-Hameaux

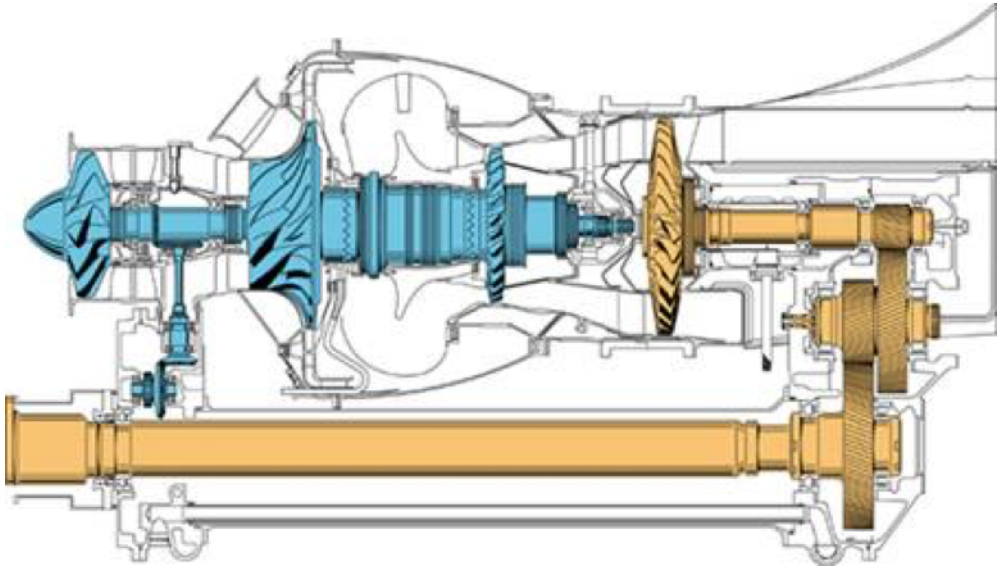
A. Froehly – LMB/INRIA Bordeaux, France

In memoriam **C. Dobrzynski** – LMB/INRIA Bordeaux, France

Tetrahedron VI workshop, 3rd of October, 2019, INRIA Saclay

Context: aeronautical engines





Aeronautical engine design is driven by two major constraints

Fuel efficiency

- Economic constraints
  - reduced fuel consumption
  - reduced CO_2 emissions
- Global efficiency of the engine

Turbofan = ***Ducted fan*** + ***Gas turbine (core engine)***

$$\eta_{glob} = \eta_{prop} \times \eta_{th}$$

Propulsive efficiency — ***Thermal efficiency***

High bypass ratio architecture

Ultra-high pressure ratio core engine

$$\eta_{th} = 1 - \left(\frac{P_1}{P_2}\right)^{\frac{\gamma - 1}{\gamma}}$$

Pollutant emissions

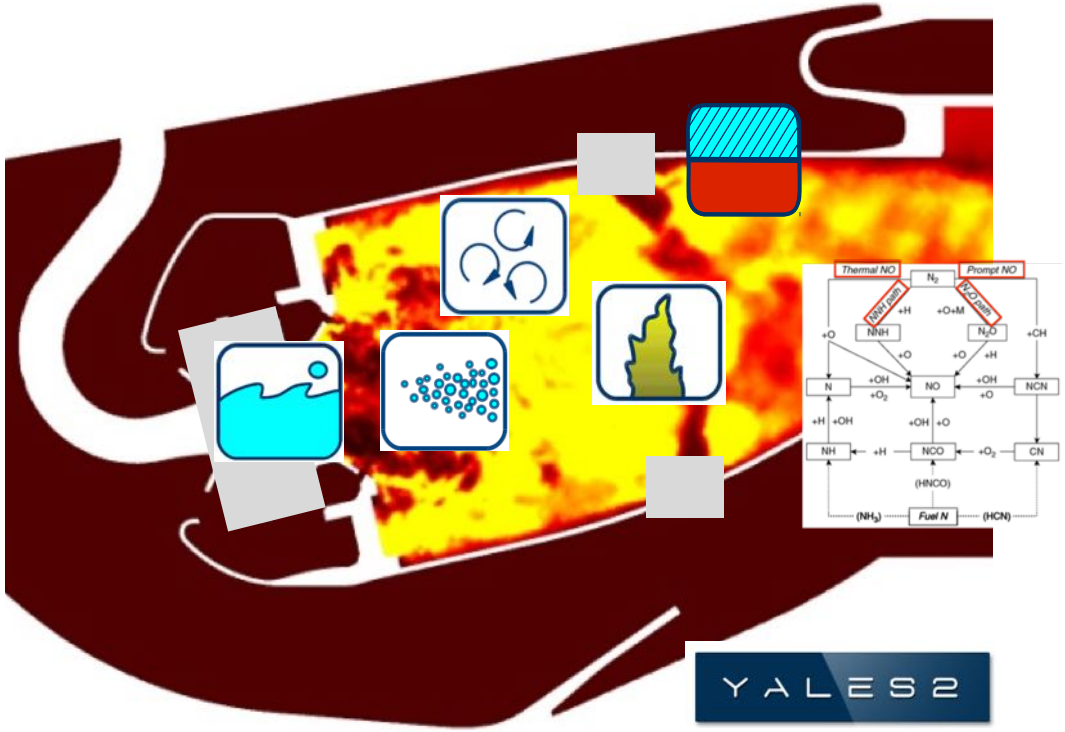
- International regulations
  - CAEP regulations
-

Smoke in the trail of a B-52

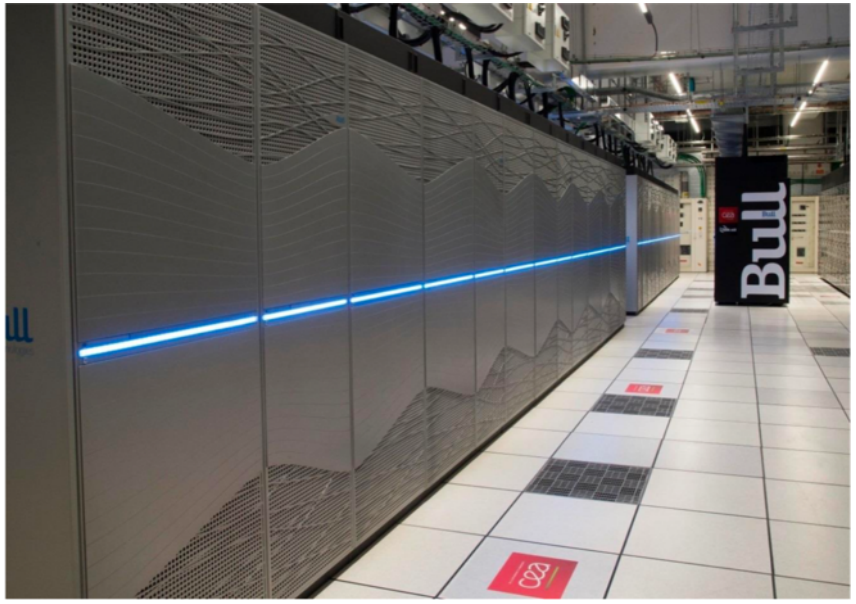
amber

Large-Eddy Simulation in the aeronautical context

- LES solves the large scales and models the small scales
- It is well suited to the high-fidelity simulation of aeronautical burners
- Challenges
  - Unsteady, multi-scale, multi-physics flows
  - Need to exploit modern super-computers



Aircraft engine chamber



JOLIOT-CURIE, PRACE/GENCI at the Très Grand Centre de Calcul, CEA
9 Petaflops, 124 000 cores

The CFD platform: YALES2

- Developed by CORIA and the French Combustion Community
  - 250+ researchers/engineers trained at CORIA since 2009
  - 150+ articles (Google Scholar)
- A unique network to ease collaboration and transfer of numerics and models to the industry

Academic partners

GIS SUCCESS [1]
CORIA, IMAG, LEGI, EM2C
IMFT, CERFACS, IFP-EN, LMA

ULB, UMONS, UCL, LOMC,
PPRIME, LMB/INRIA,
CORNELL U., SHERBROOK U.
VERMONT U.

HPC experts

ECR lab
INTEL/CEA/GENCI/UVSQ
IBM/ROMEO

HPC centers

CRIHAN, IDRIS, CINES, TGCC
GENCI, PRACE

Industrial partners

SAFRAN
ARIANE GROUP
SOLVAY
AIR LIQUIDE
SIEMENS/GAMESA
…

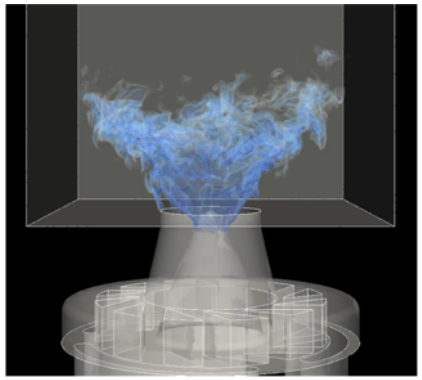
SMEs

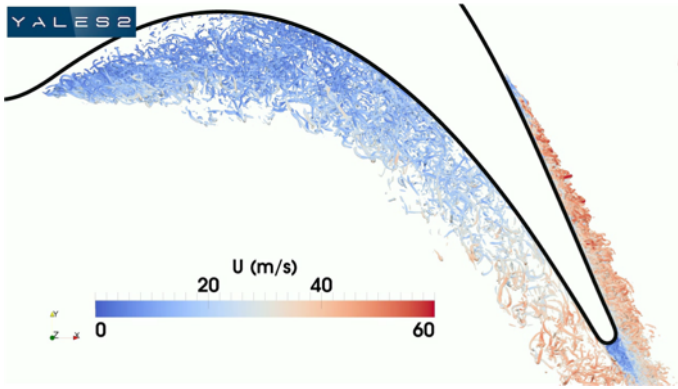
GDTech

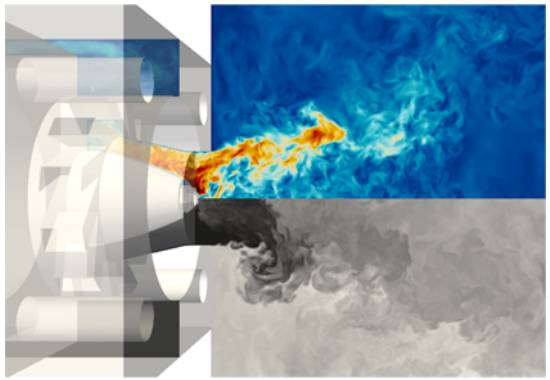
[1] http://success.coria-cfd.fr

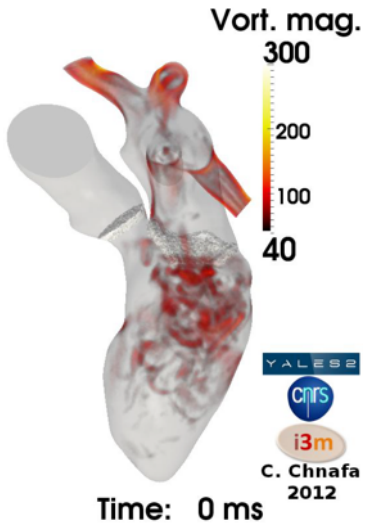
The CFD platform: YALES2

- Features
  - **Unstructured meshes** (complex geometries) and **adaptive grid refinement**
  - **Low-Mach number Navier-Stokes equations** (incompressible and variable density) solved with a projection method
  - **Double-domain decomposition** [3] and **hybrid OpenMP/MPI communications**
  - Highly efficient solvers for linear system inversion (PCG, DPCG)
  - **4th-order** central finite-volume method and **4th-order** time integration
  - Two-phase flows (Lagrangian particles)
  - Spray and atomization (Levelset)
  - Combustion modeling (Tabulated or complex chemistry, NOx prediction model…)
  - **Suited for massively parallel computing** (>32 000 procs)





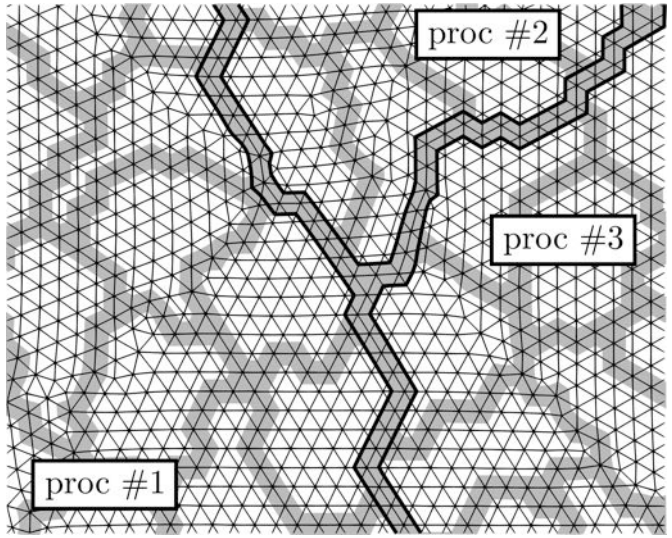




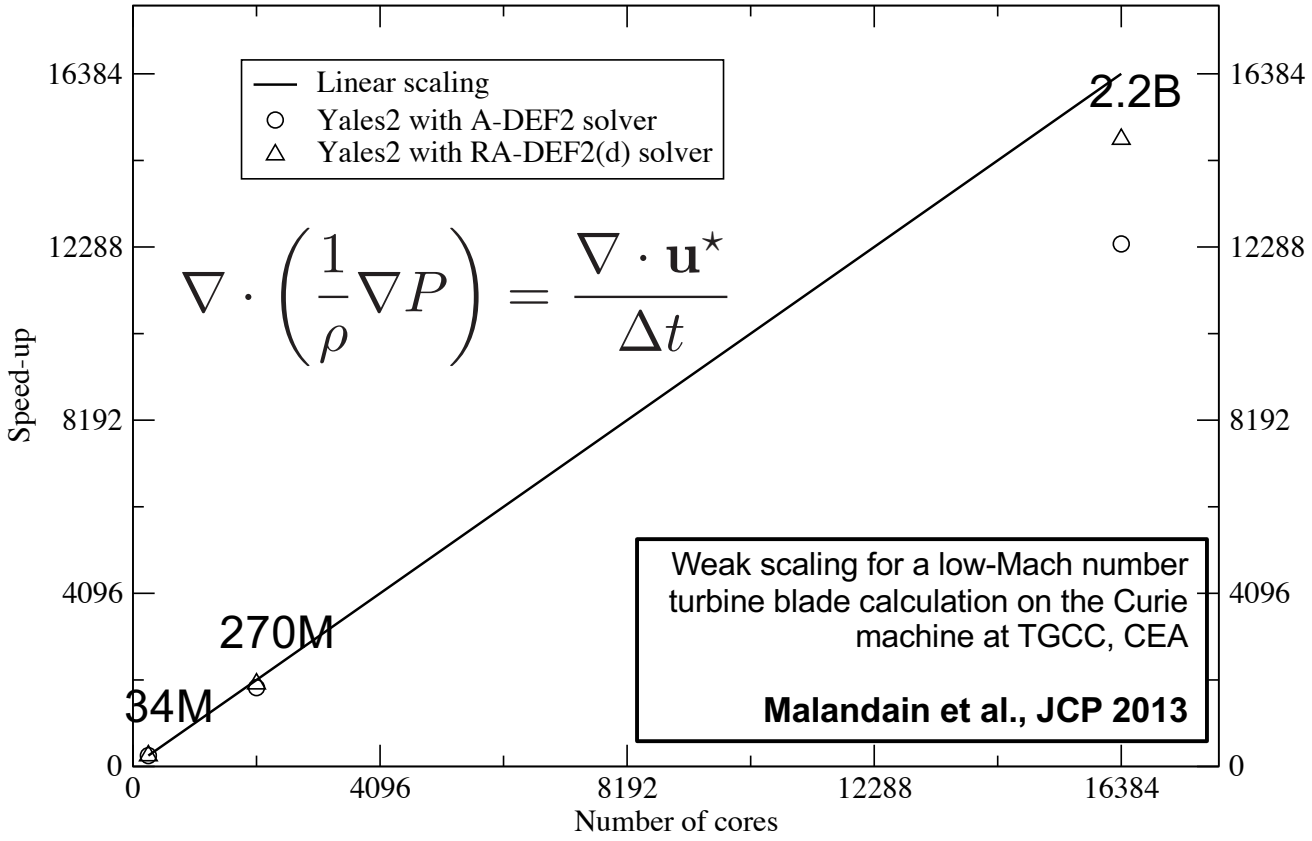
[1] YALES2 web site, http://www.coria-cfd.fr
[2] SUCCESS web site, http://success.coria-cfd.fr
[3] Moureau et. al., CR Mecanique, 2011

HPC: in-core and parallel performances

- MPI and hybrid coarse-grain OpenMP/MPI
- Two-level domain decomposition [1]
  - Enables to fit in L2 cache memory
  - Used for the preconditioning of the linear solvers



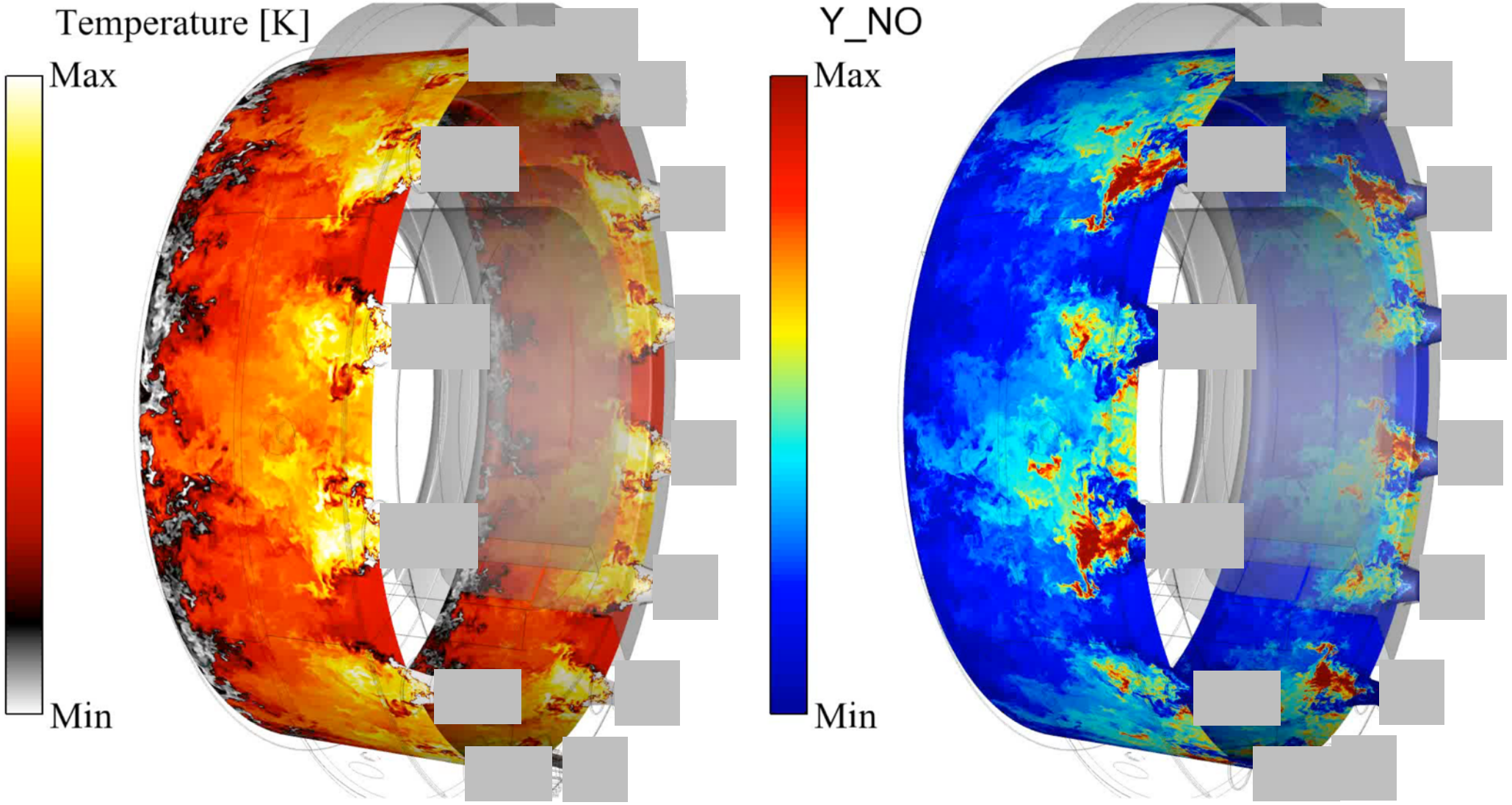
- In-house linear solvers



[1] Moureau et. al., CR Mecanique, 2011

Prediction of pollutant emissions in realistic burners: Application to a low-NOx helicopter engine

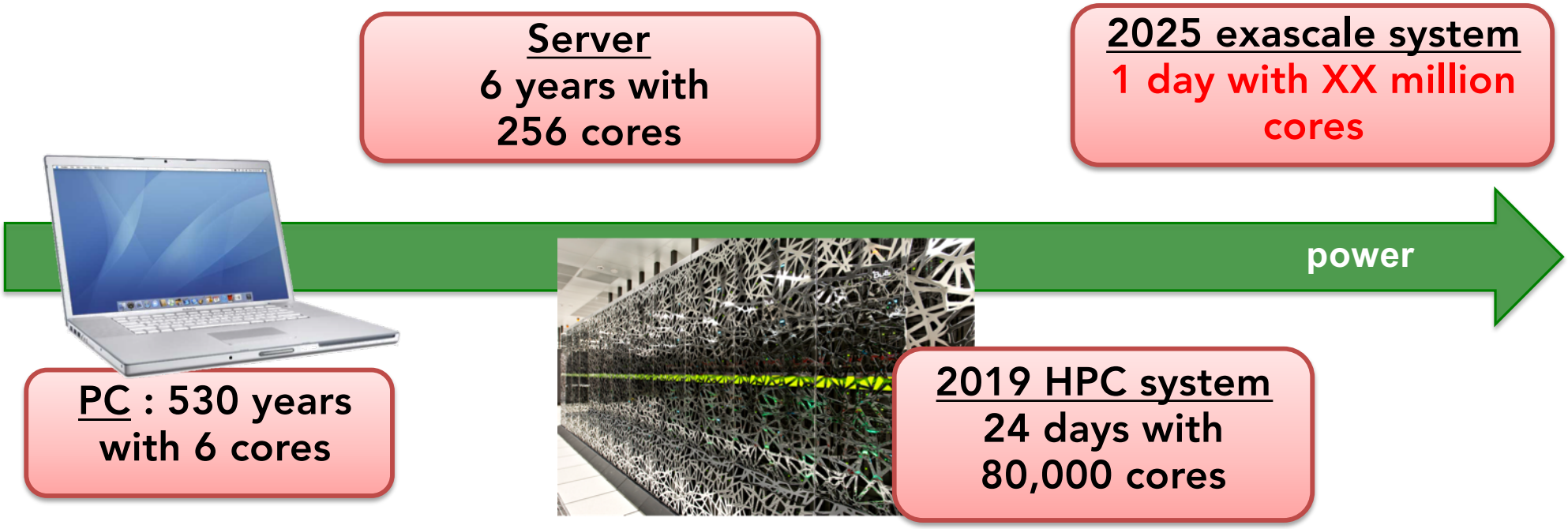
- LES from J. Lamouroux, SAFRAN Helicopter Engines, in 2015
- 376M elements for 2 injectors, tabulated chemistry, dedicated NOx model [1]



[1] Pecquery et al. (2014) Comb. Flame 161(2)

Future large-scale computations in the industry

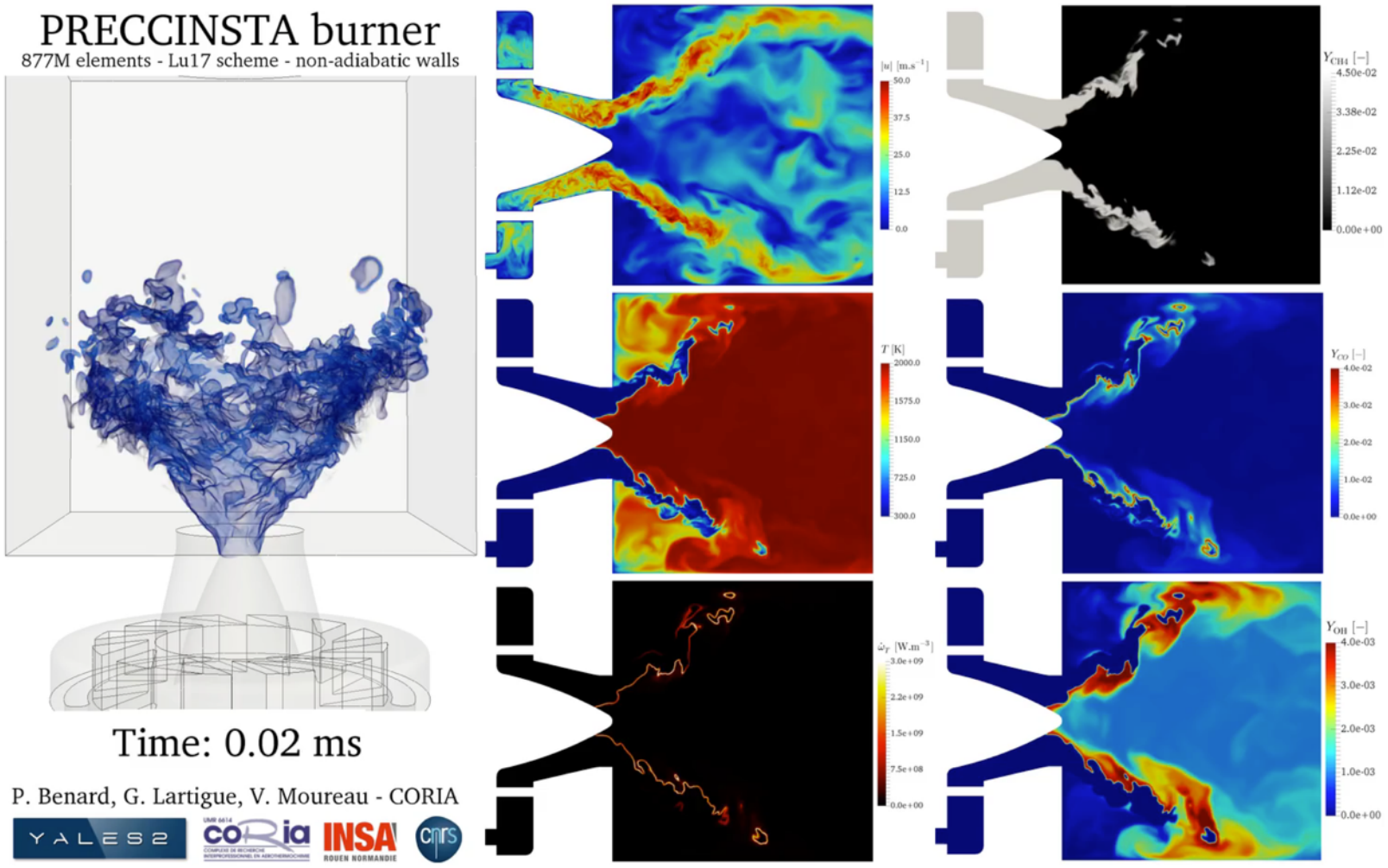
- Simulation time for an aeronautical combustor with a mesh of 10 billion tets (2025+ target)

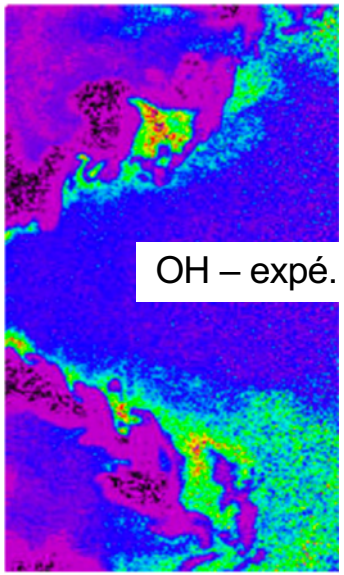


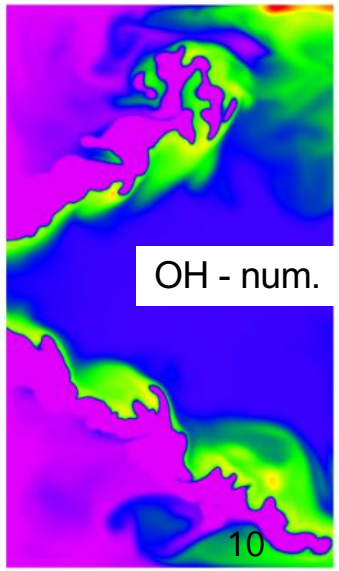
- Many issues have to be addressed
  - Mesh management, solving of large linear systems, load balancing of chemistry integration, data-mining, …

High-fidelity simulation of semi-industrial burners

- LES of the lean-premixed PRECCINSTA burner with finite-rate chemistry (Sankaran et al. scheme, 17 species, 73 reactions) and heat loss [1]
- 2018 FIRELES PRACE project, 16384 cores on Curie

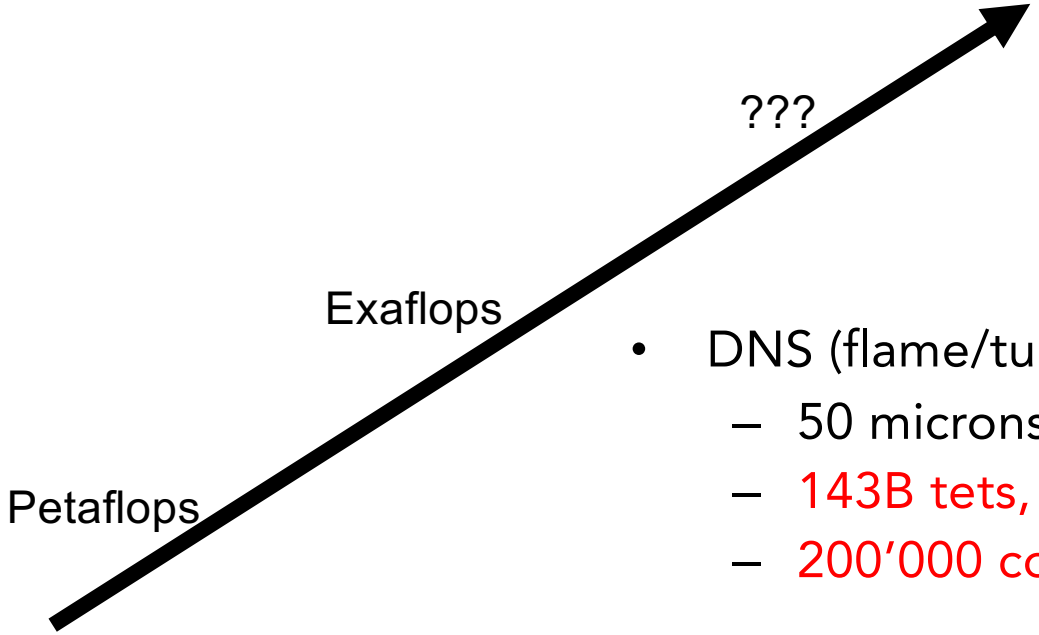






[1] P. Bénard et al. (2018) Int. Comb. Symp., Dublin

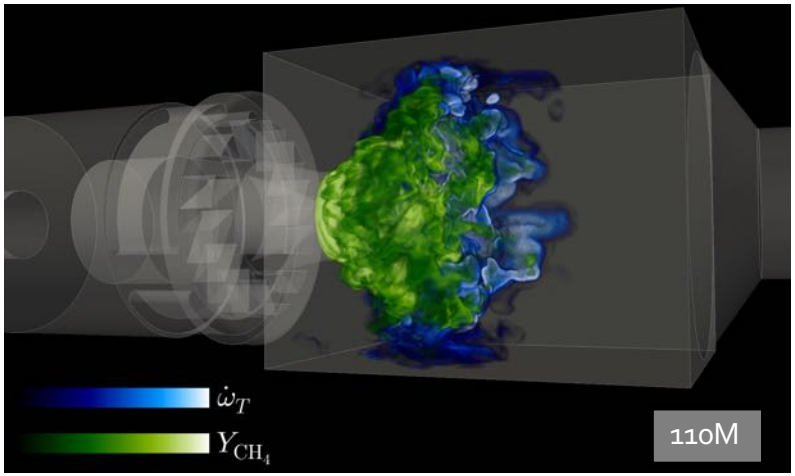
When can we expect a DNS of semi-industrial burners?



- DNS (species profile resolving)
  - 10 microns
  - 89’000B tets
  - ???

- DNS (flame/turbulence interactions resolving)
  - 50 microns
  - 143B tets, 442M CPUh
  - 200’000 cores, 13 weeks

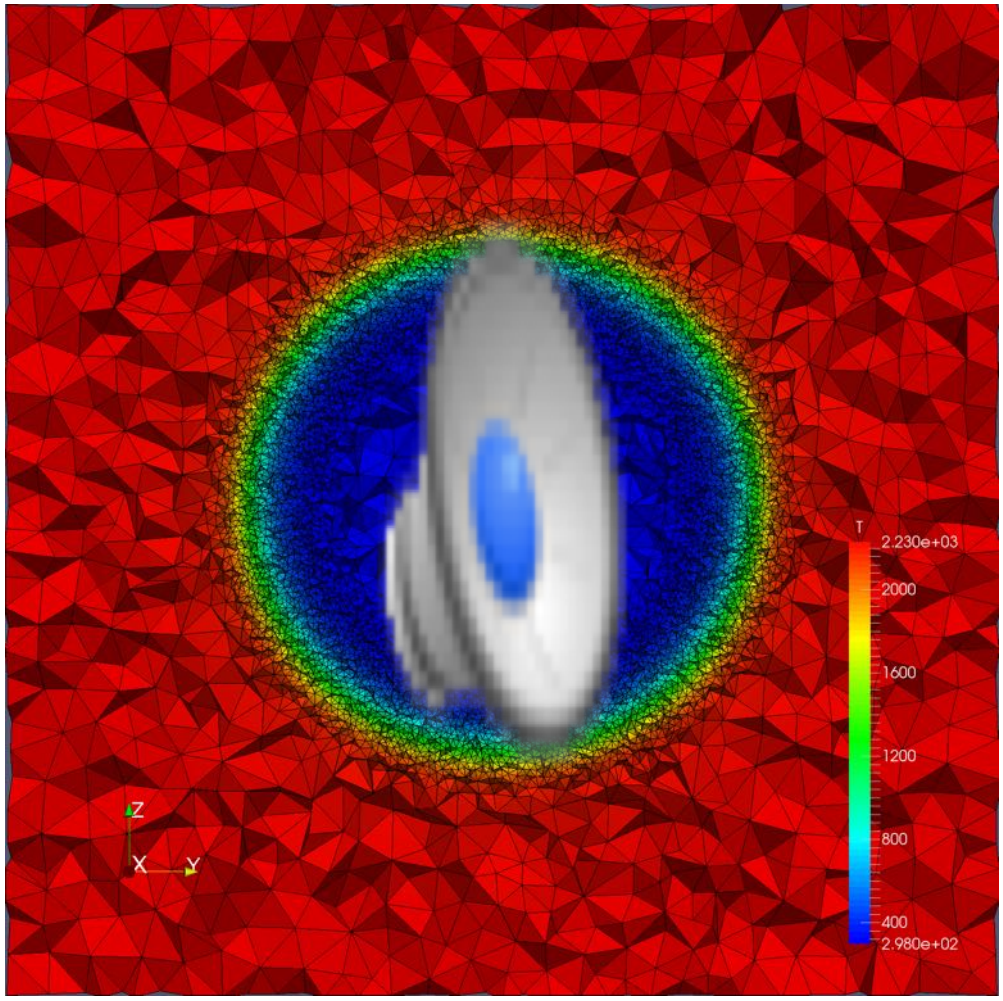
- LES of PRECCINSTA burner
  - 17 species, 72 reactions
  - 150 microns
  - 877M tets, 4M CPUh
  - 16,384 cores, 1 week

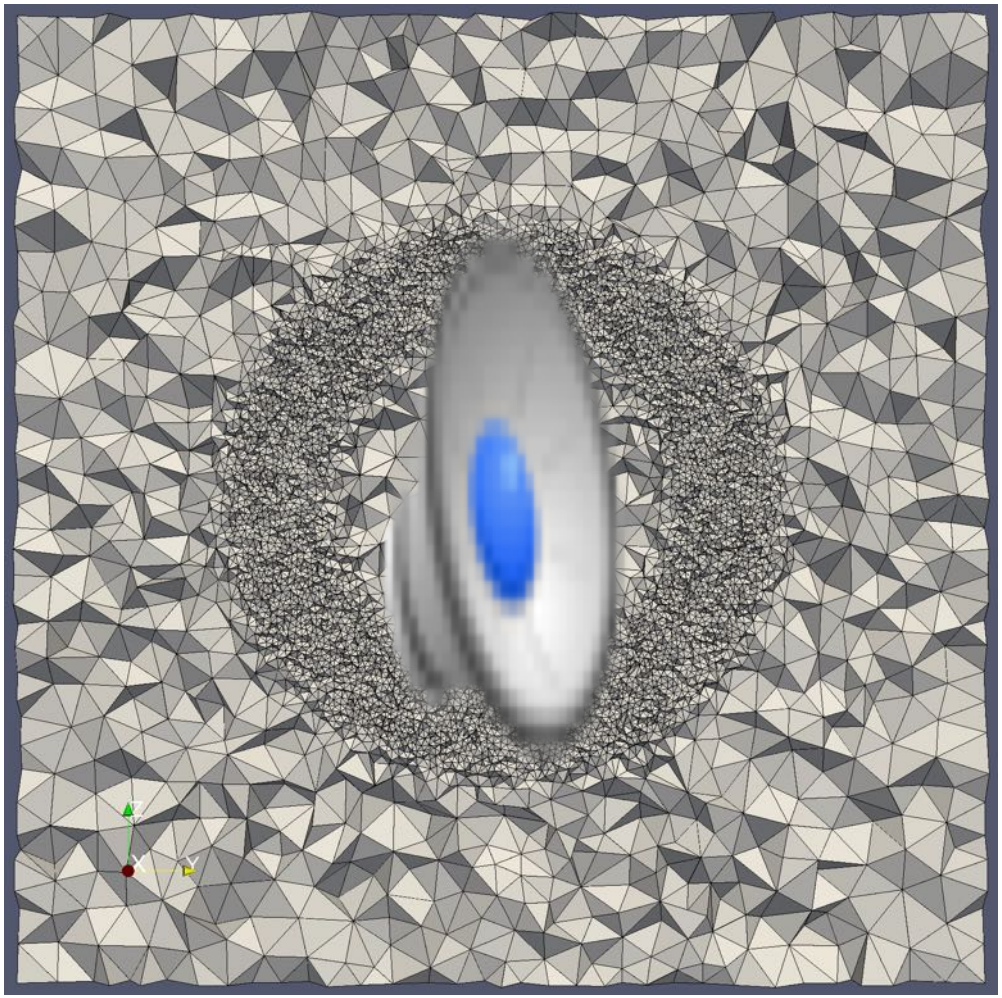


- Alternative to reach DNS faster ?

A possible way to reach our target simulations faster

- Simulation of a premixed flame with dynamic h-adaptation
- F-TACLES combustion model [1], refinement ratio = 6





- A prerequisite is to know how to perform parallel h-adaptation

[1] Fiorina et al., C&F, 2009

Outline

- Context & Motivation

- **Parallel mesh adaptation**
  - **Principle**
  - **Load balancing and data transfer**
  - **Example**

- Dynamic mesh adaptation
  - Metric definition and adaptation triggering
  - Examples
  - Performances

- Improvements to the dynamic mesh adaptation process

- Conclusions & perspectives

Objectives

- To design a dynamic mesh adaptation method with the constraints:
  - Tailored for **distributed-memory** system (10,000+ cores)
  - **Efficient** enough to be called every 10 fluid iterations
  - Compatible with the modeling approach (**finite-volume + LES**)
  - No remeshing at material interfaces to avoid interpolation errors

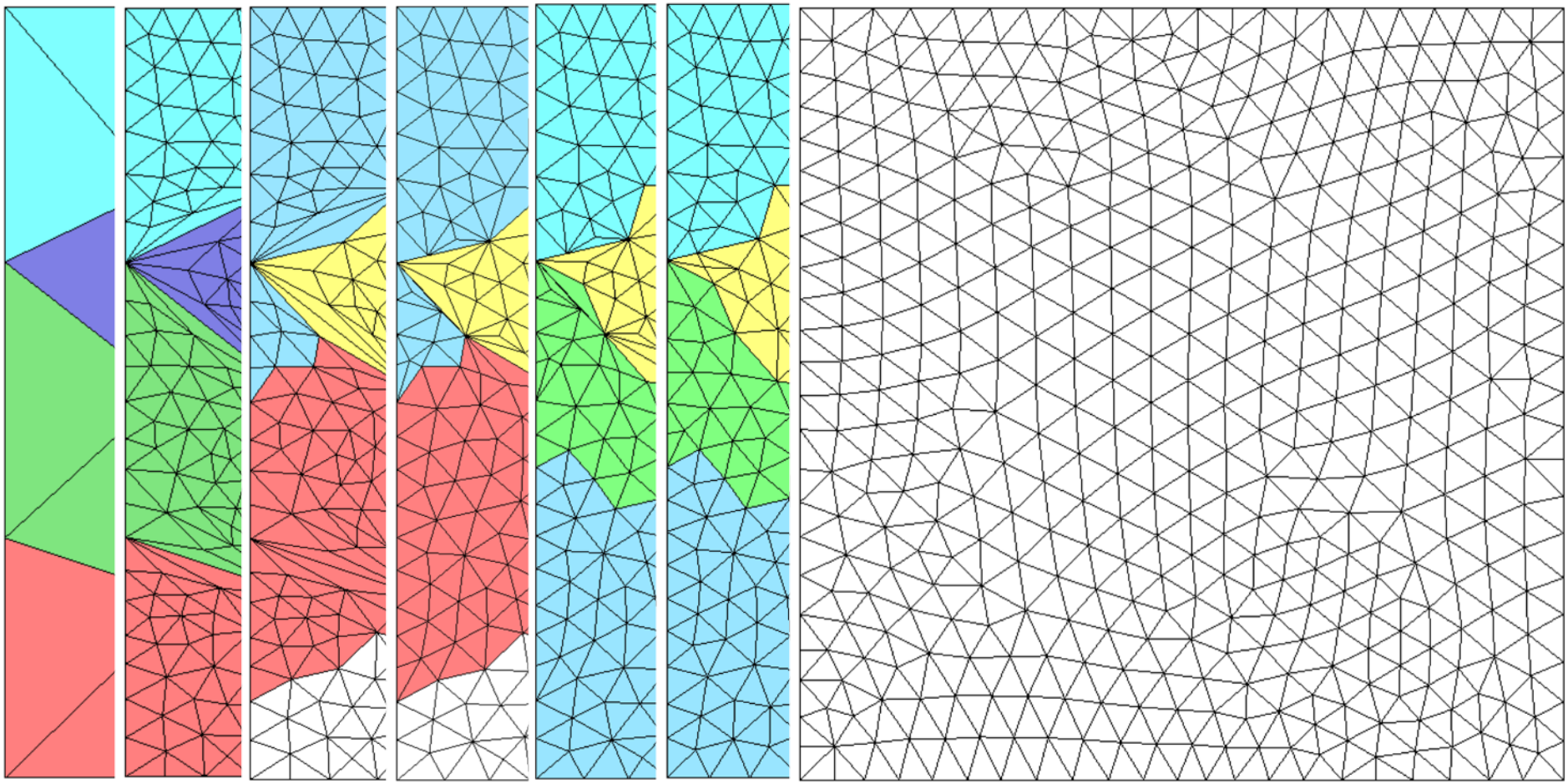
- Some choices
  - **Isotropic** mesh adaptation
  - Parallelism handled by the flow solver (interpolation, data transfer)

- A first check if these objectives are reachable for tets
  - Reduced computation cost of fluid solver: 30 to 500 µs/iter/node
  - Adaptation of a distributed mesh with **MMG**: order of 100 µs/node

A moving interface method for parallel h-adaptation on distributed memory systems

- Principle with a single-level domain decomposition [1]

$\mathcal{M}_{\text{init}}$ $\mathcal{M}_{\text{target}}$



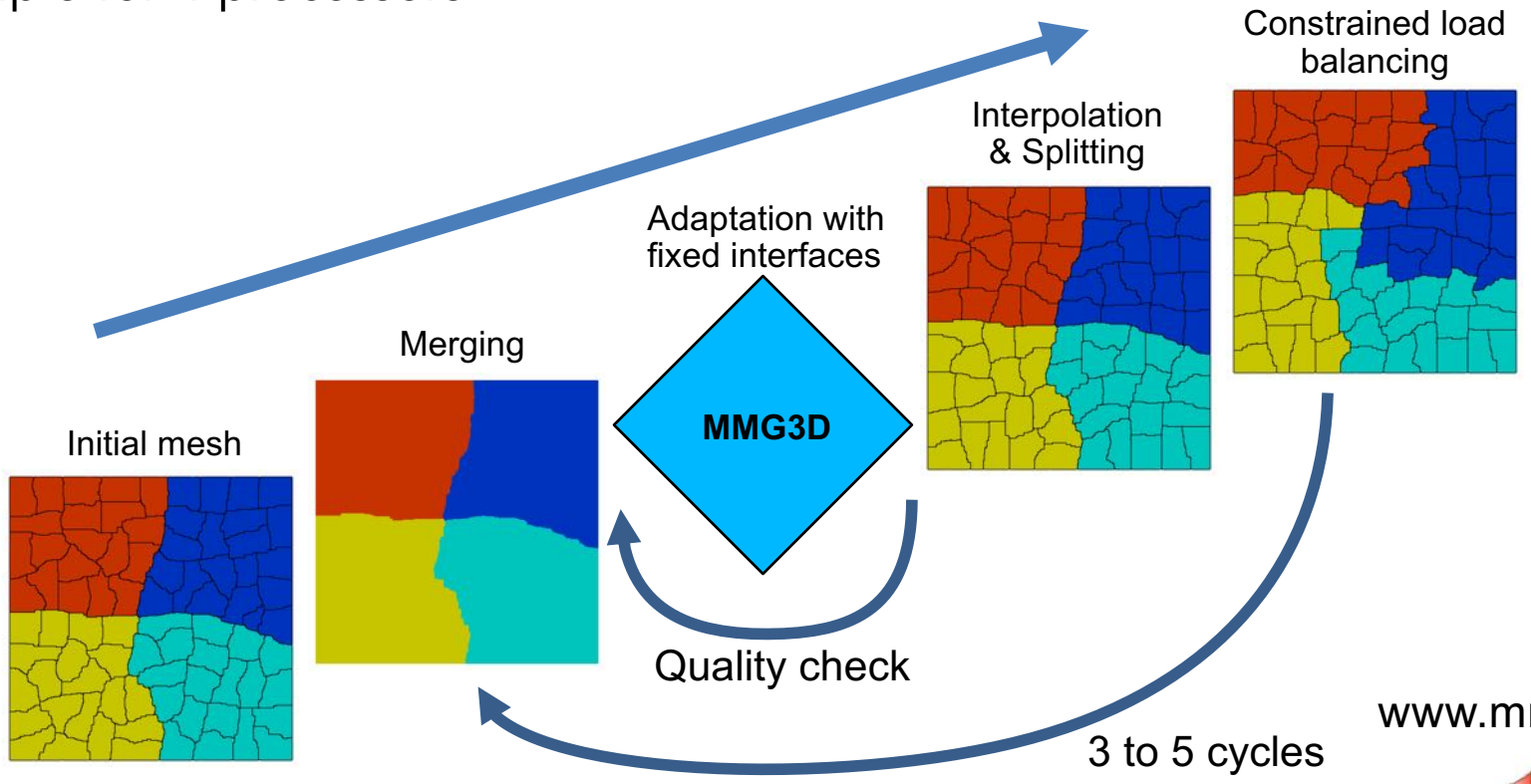
Final mesh

(courtesy A. Froehly - ParMMG)

[1] Digonnet et al., IJHPCA, 2017

A moving interface method for parallel h-adaptation on distributed memory systems

- Generalization to a 2-level domain decomposition [1]
- The MMG library [2] from INRIA/IMB adapts the mesh (vol/surf) on each rank
- Skewness is the main measure of the mesh quality
- Example for 4 processors



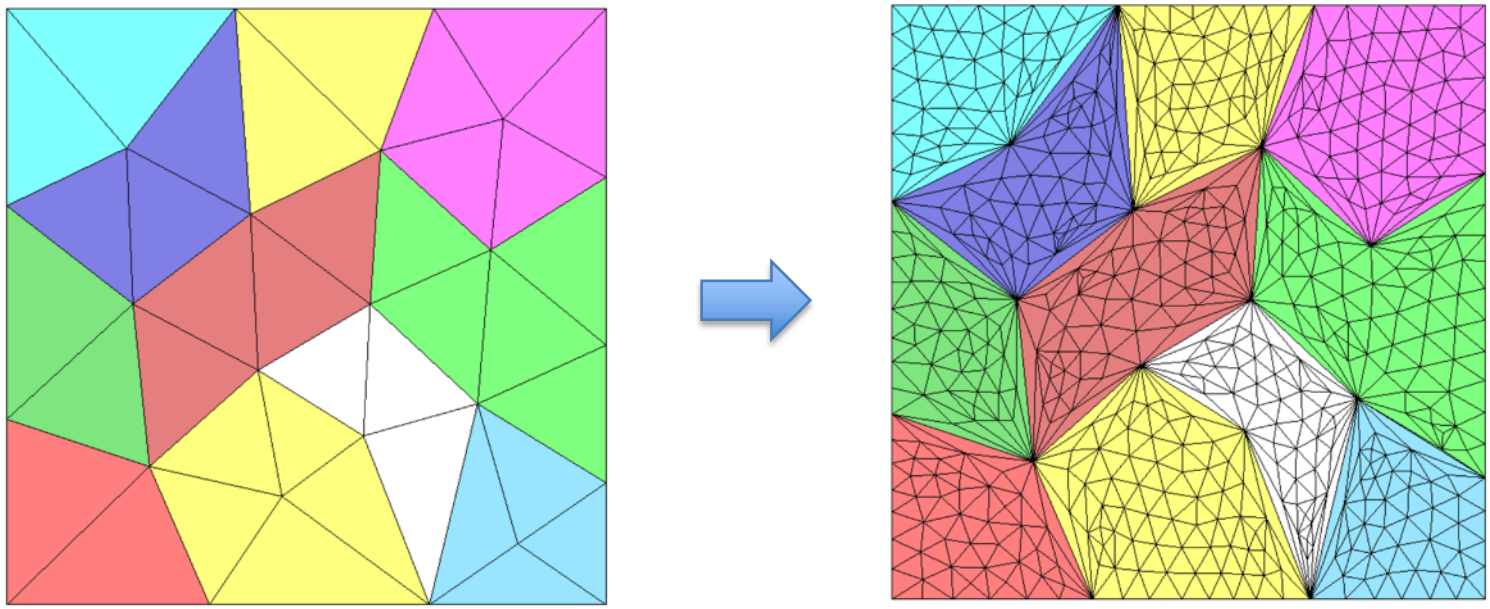
www.mmgtools.org

upgrade your meshes

- MMG input: **isotropic metric**

Input metric to MMG

- At each adaptation step, a metric is given to MMG: $\mathcal{M}_{\text{step}}$
- $\mathcal{M}_{\text{step}}$ has to be different from $\mathcal{M}_{\text{target}}$ because it is not compatible at the interfaces and walls

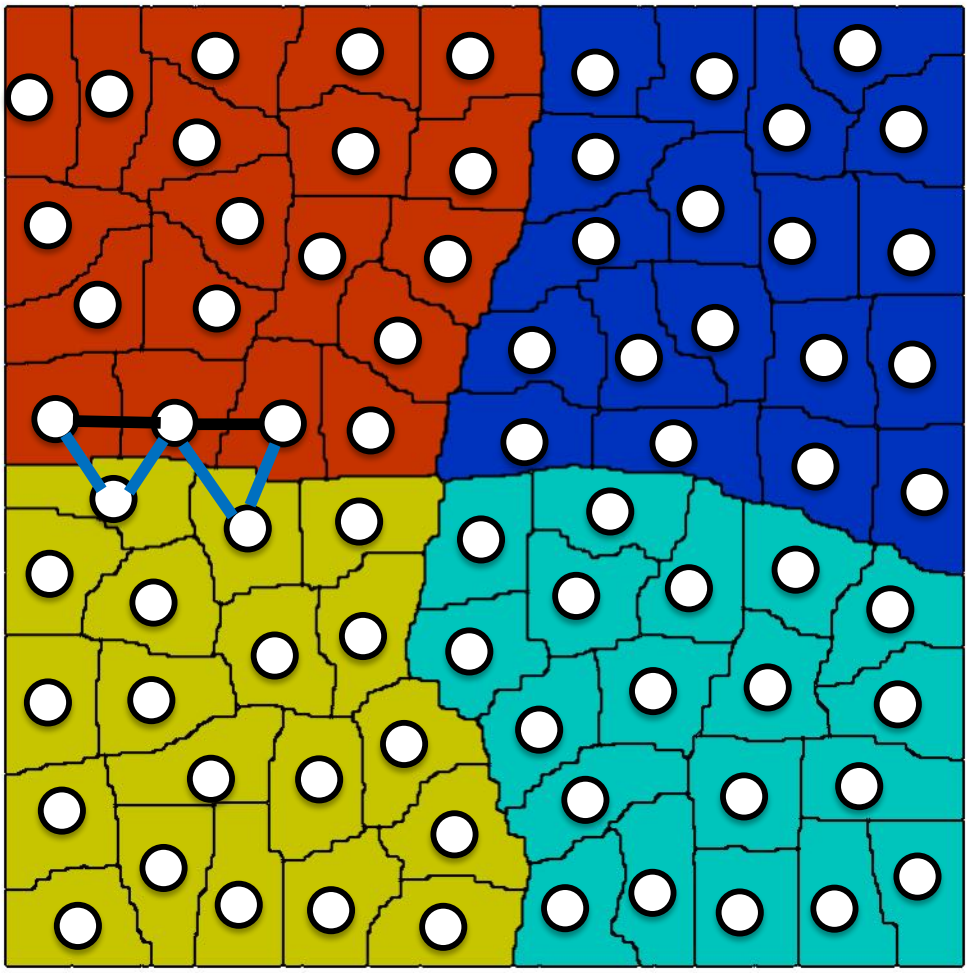


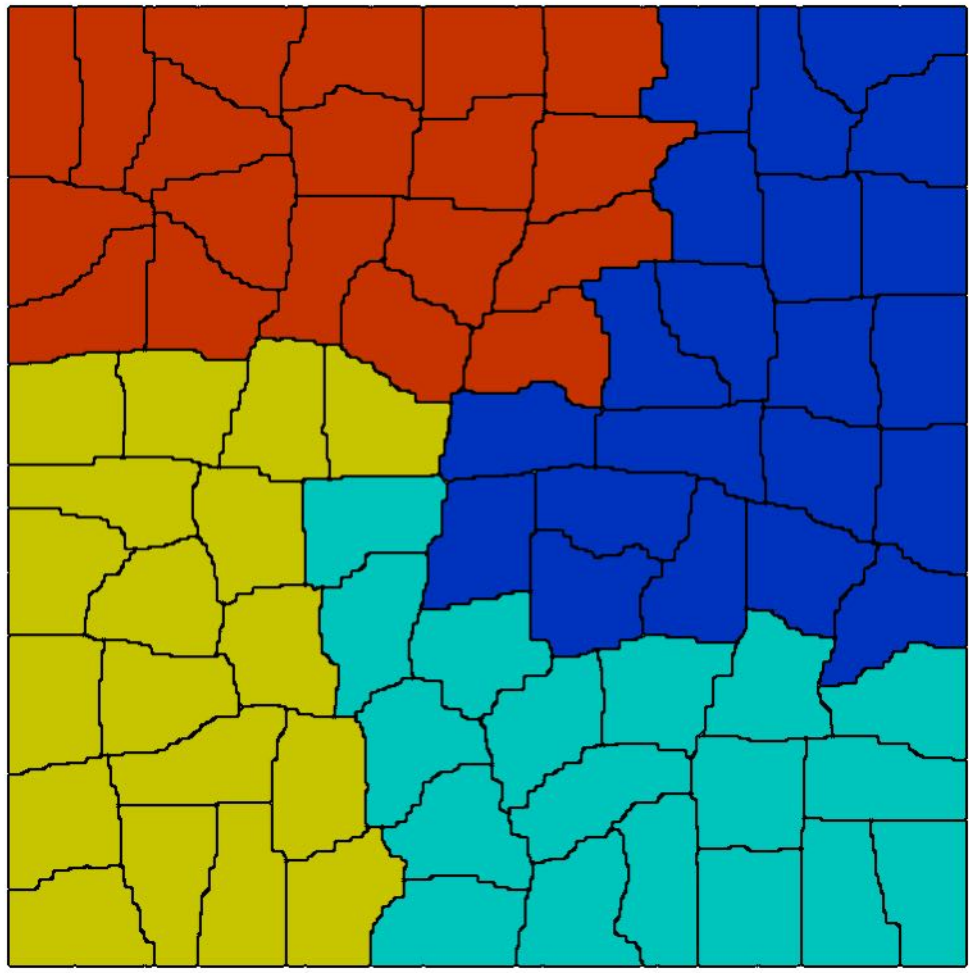
$$\begin{cases} \mathcal{M}_{\text{step}} = \mathcal{M}_{\text{current}} & \text{at the interface} \\ \mathcal{M}_{\text{step}} = \mathcal{M}_{\text{target}} & \text{away from interface} \\ \mathcal{M}_{\text{step}} = \text{FMM}(\mathcal{M}_{\text{current}}, \mathcal{M}_{\text{target}}, h_{\text{grad}}) & \text{in between} \end{cases}$$

[1] Sethian, J.A., Proc. Nat. Acad. Sci., 1996

Constraint load balancing

- The first method is based on edge-weighted graph partitioning
- A graph of the element groups is built
- METIS is called on the master rank with $\omega_{\text{interface}} = 100\,\omega_{\text{inside}}$





Optimized transfer of cell groups

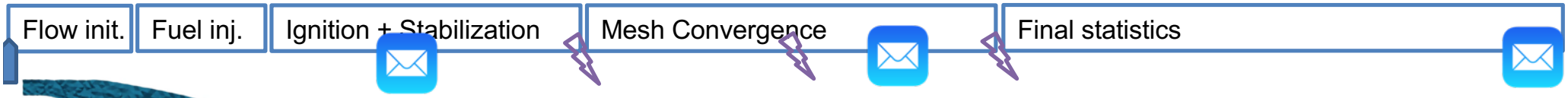
- Outer loop:
  - List all the sending operations (rank N to rank M)
  - Shuffle these sending operations
  - Perform as many parallel sending operations as possible

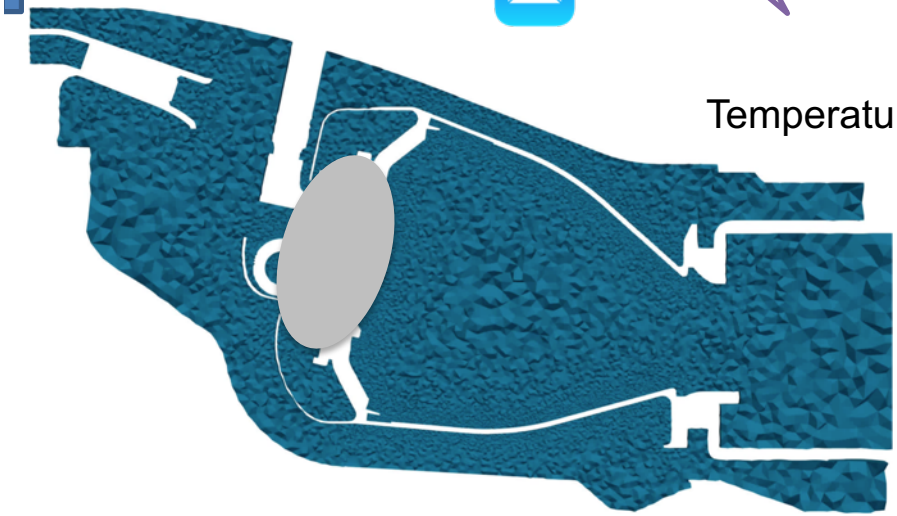
- Inner loop: rank N sends cell groups to rank M
  - **Pack** the cell groups on rank N in a single real array
  - **Non-blocking** send/recv of the pack
  - **Unpack** the cell groups on rank M
  - Rebuild **connectivity** on ranks N, M and connected ranks

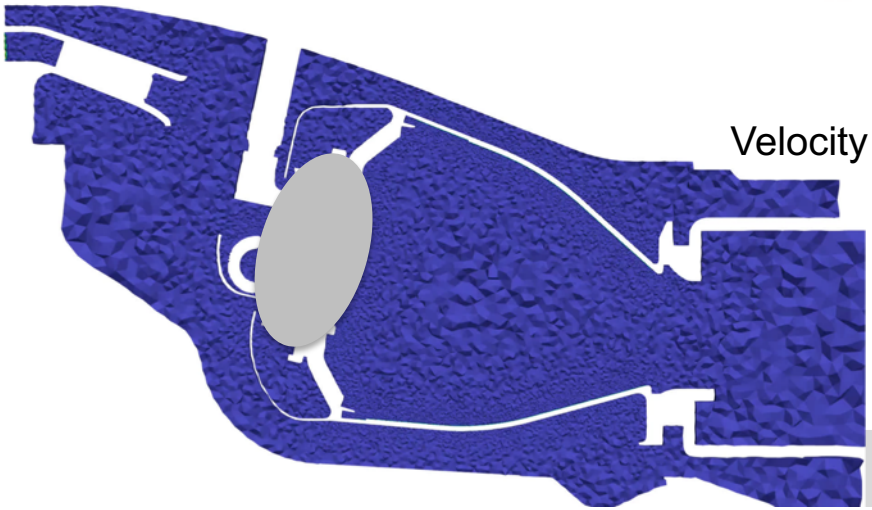
- Remarks
  - One cell group is a few MB => the MPI latency is still reasonable (a few 100 μs to 1 ms)
  - Packing/unpacking and rebuilding the connectivity are the time consuming algorithms

Example: user-independent LES of aeronautical burners

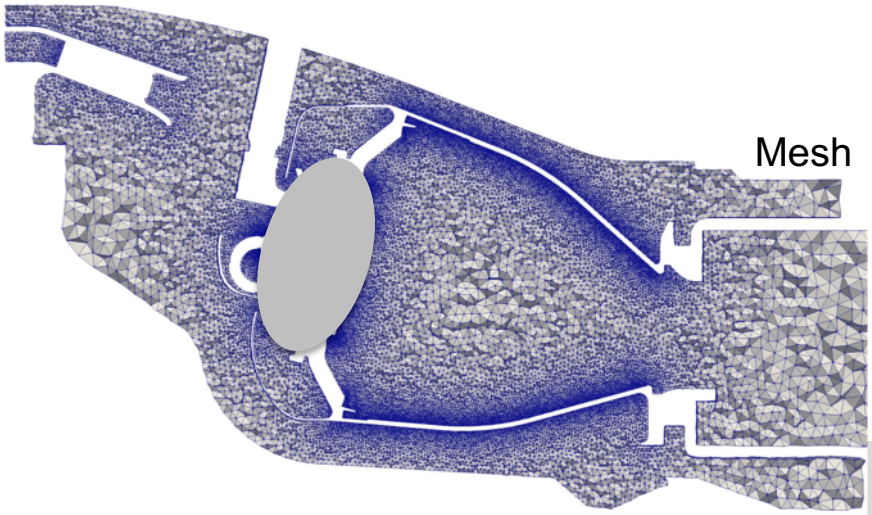
- 2018 TERATEC/Usine Digitale award
- First LES Design of Experiment performed at SAFRAN TECH in 2018
  - 9 combustion LES for injection system design (30h per run !)







- **"All in one run"** : Embedded run scenario (initialization, fuel injection, ignition, stabilization,…)
- **Automatic mesh convergence** included
- User-specified CPU budget



Outline

- Context & Motivation

- Parallel mesh adaptation
  - Principle
  - Load balancing and data transfer
  - Example

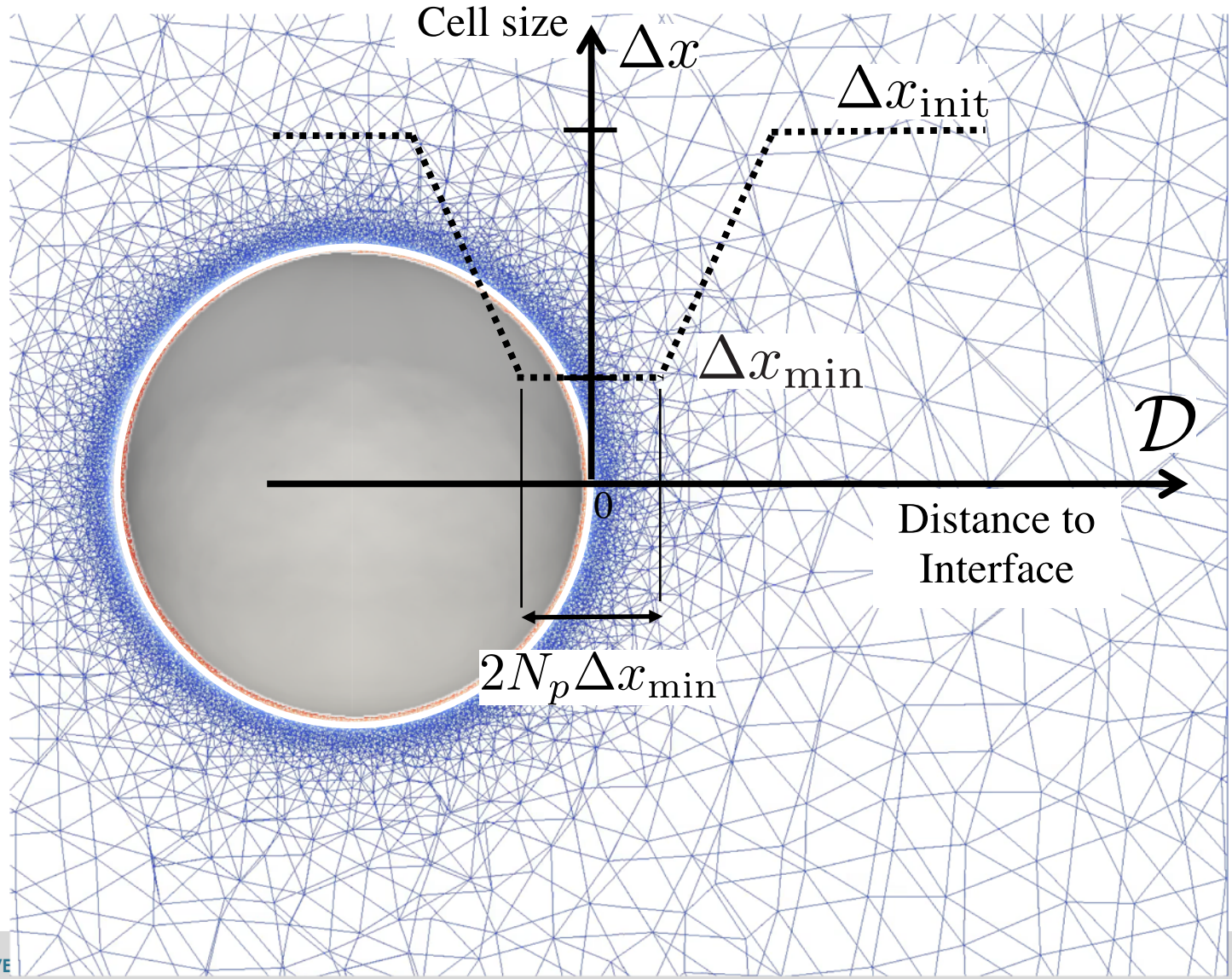
- **Dynamic mesh adaptation**
  - **Metric definition and adaptation triggering**
  - **Examples**
  - **Performances**

- Improvements to the dynamic mesh adaptation process

- Conclusions & perspectives

A simple metric definition for material interfaces

- The target metric is based on the distance to the interface (flame front or liquid/gas interface)



When to remesh?

- Metric definition

$$\mathcal{M} = \alpha \mathcal{M}_{\text{front}} + (1-\alpha)\mathcal{M}_{\text{background}}$$

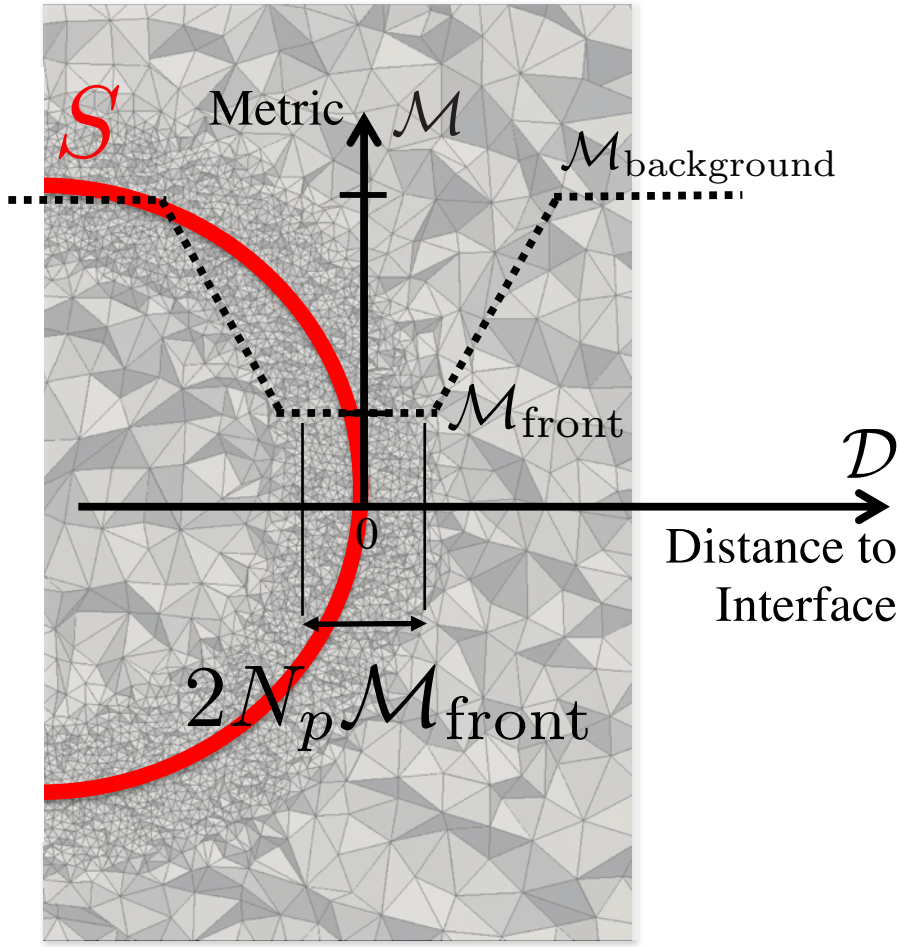
with
$$\left| \begin{array}{l} \alpha = f(S, N_p) \\ S : \text{Front sensor} \\ N_p : \text{Ncells propagation} \end{array} \right.$$

- **When to remesh ? [1,2]**
- Each $N_a \Delta t$

- Given N_p we can derive a **CFL-like condition for N_a**:

$$\text{CFL}^{\text{DMA}} = \frac{|\mathbf{u} \cdot \mathbf{n_\Phi}| * N_a \Delta t}{N_p \mathcal{M}_{\text{front}}}$$

V (numerator), dt (numerator), dx (denominator)

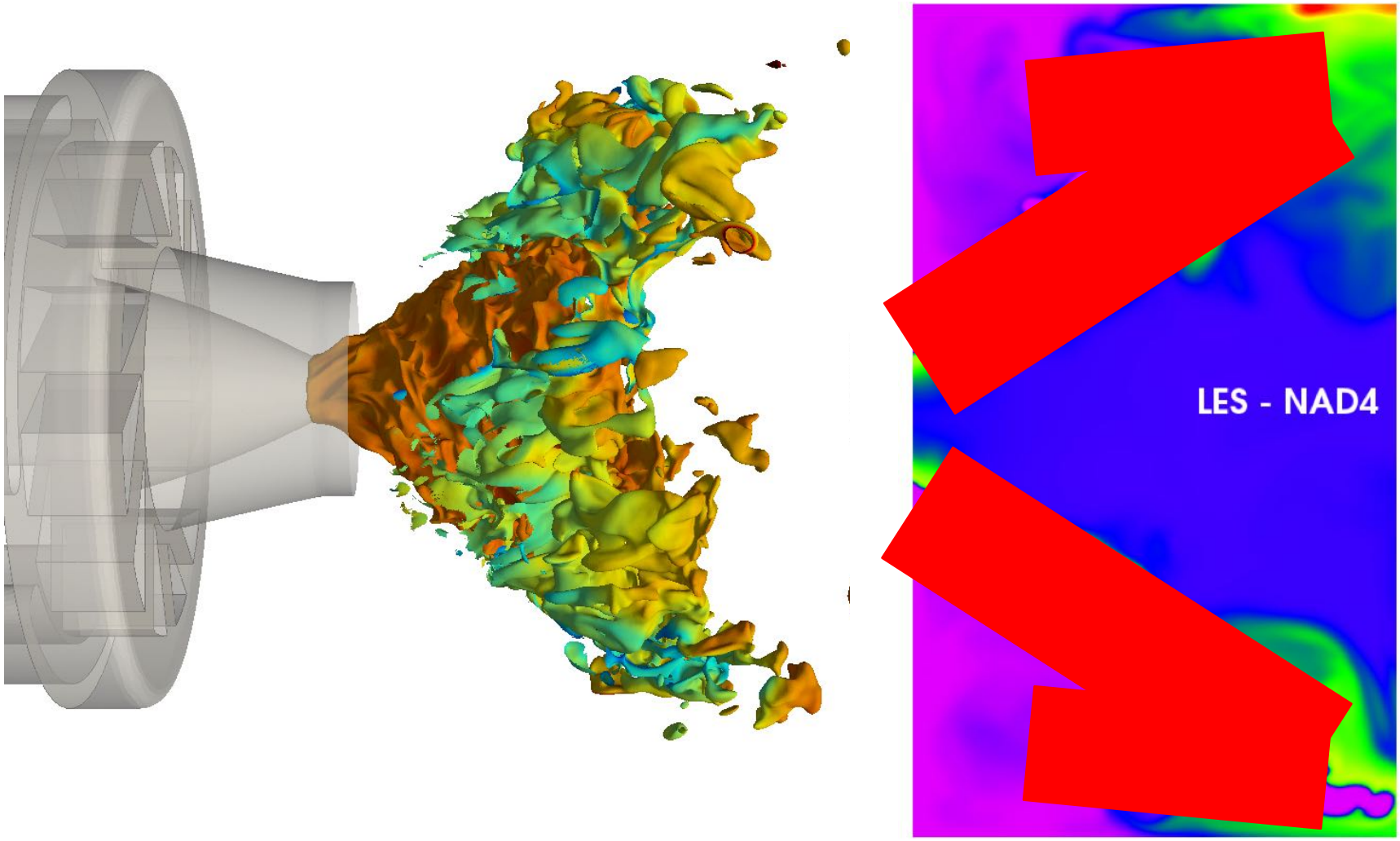


- How to choose N_p ? Depends on the interface dynamics

[1] Alauzet et al. IJNMF 2003 [2] Alauzet et al. JCP 2007

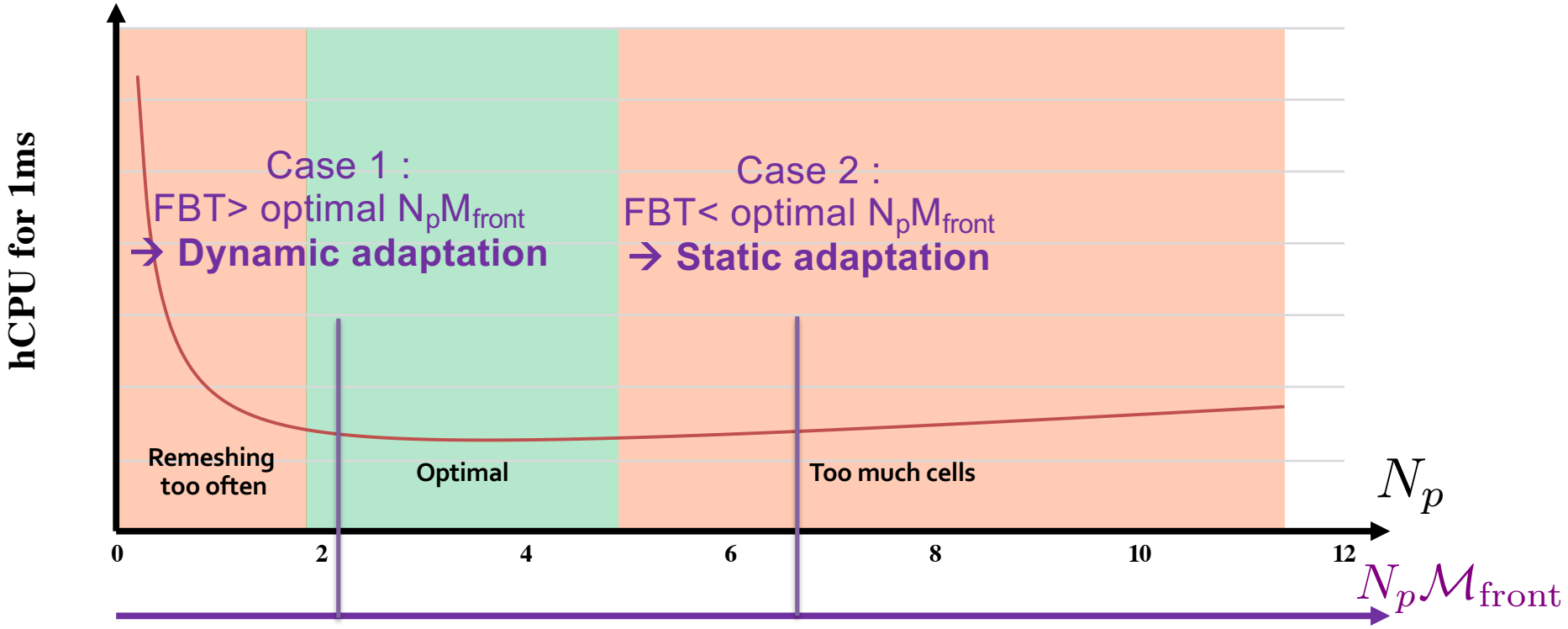
The Flame Brush Thickness

- The FBT can be defined as the region where the flame evolves
- The same can be defined for other material interfaces



How to choose the size of refined mesh region (N_p) ?

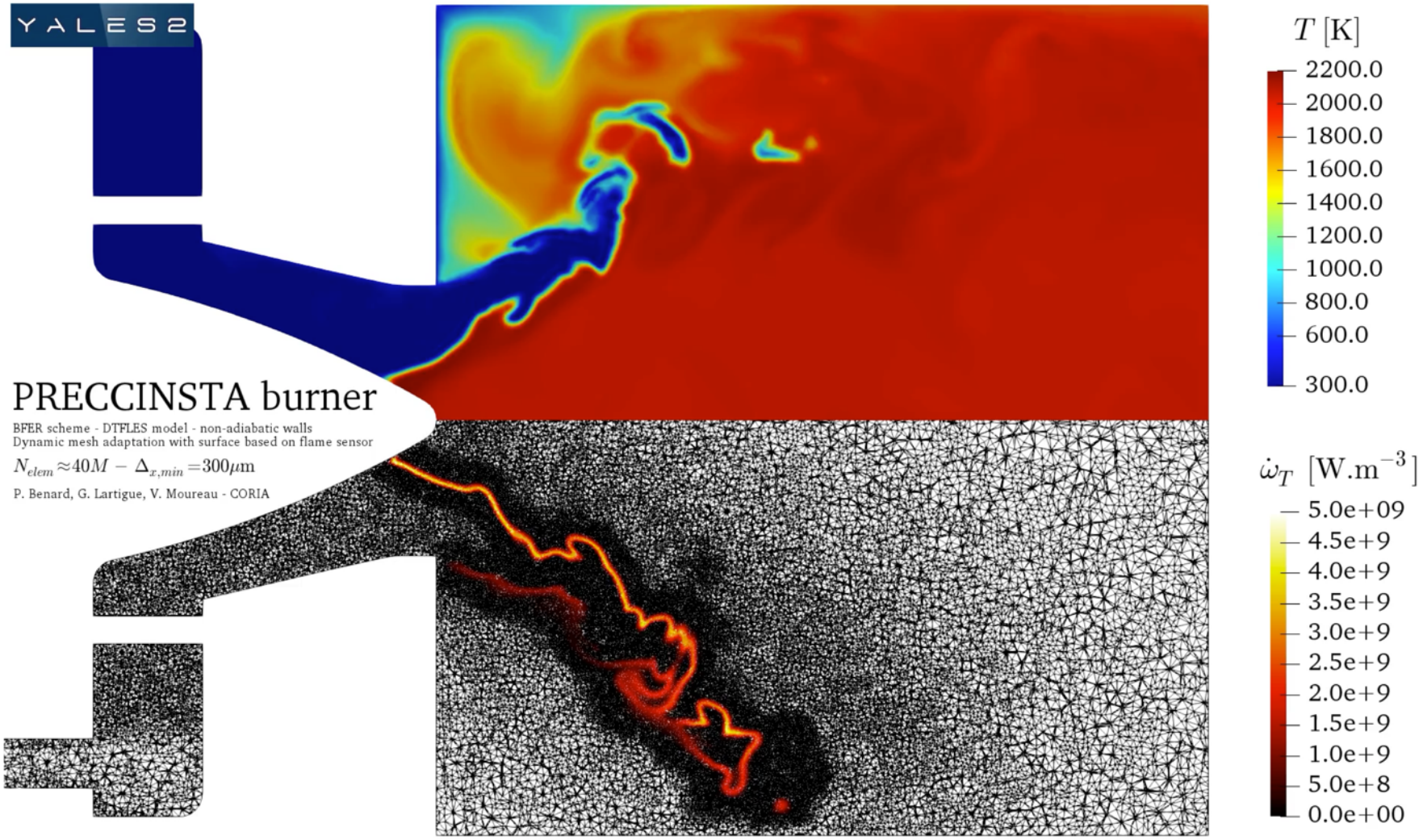
- Determine optimal N_p



- Dynamic or static adaptation ? Depends on the **Flame Brush Thickness (FBT)**
- /!\ Choice has to be considered at each new numerical setup (target cell size, etc…). N_p is constant in the following examples.

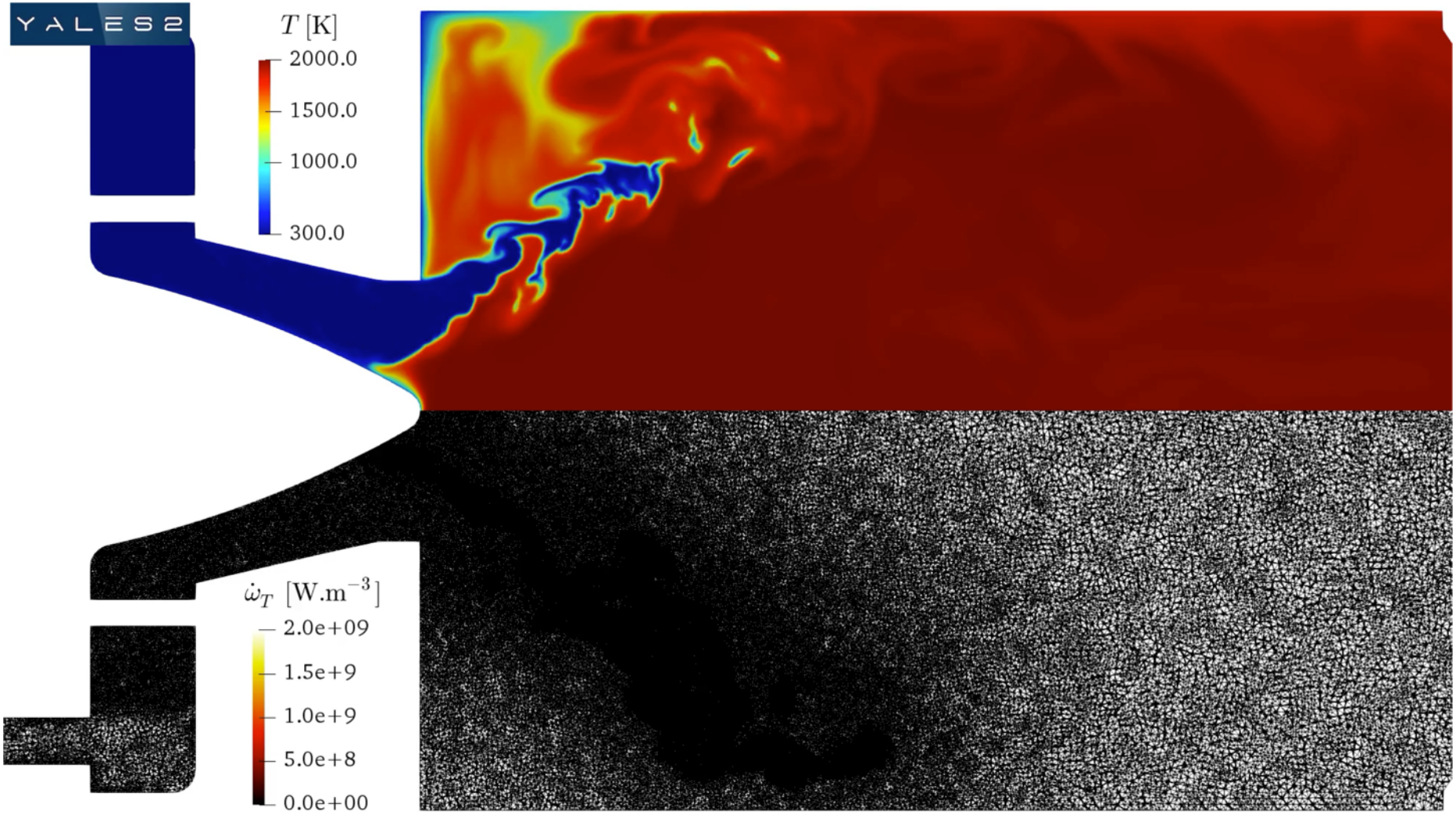
Application to a lean-premixed flame

- Metric definition based on a progress variable
- Same resolution as 110M (300 microns) but with 38M cells (**x3 speed-up**)
- **Volume/surface adaptation with MMG5.3 and YALES2 2018.11**



Application to a lean-premixed flame

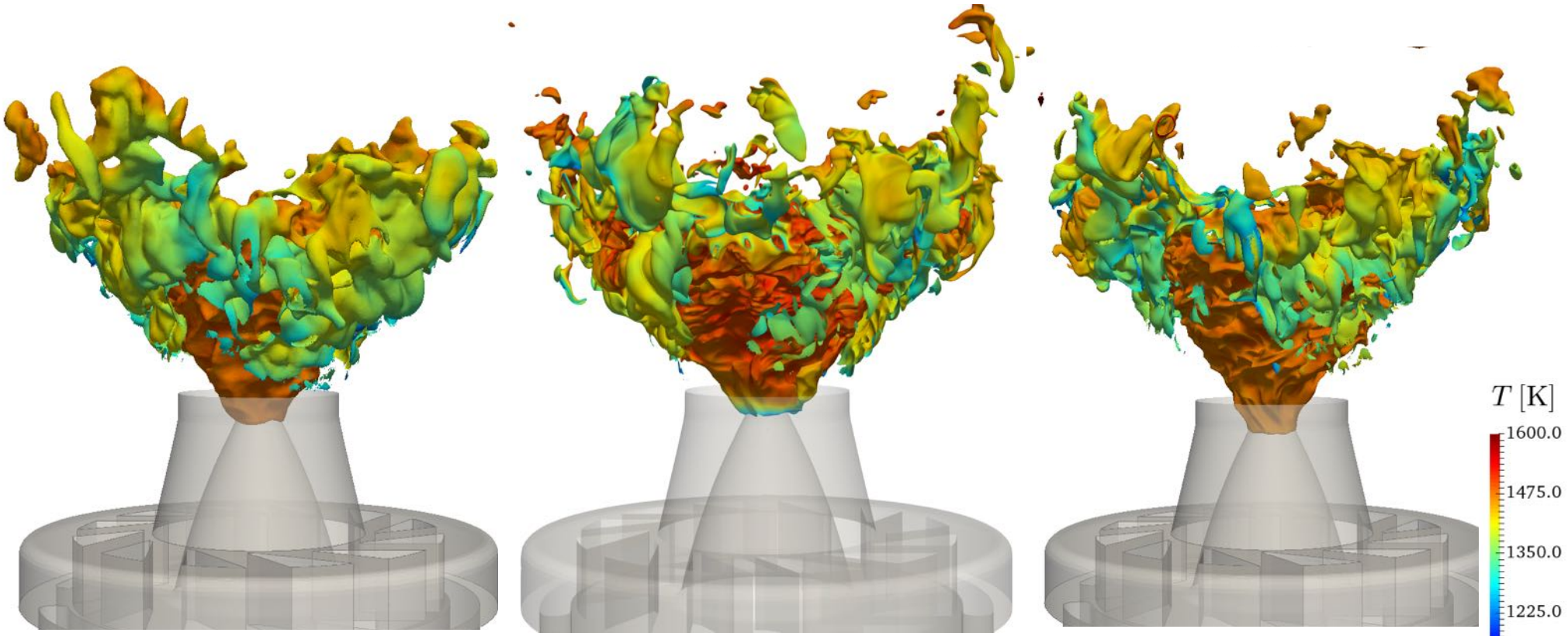
- Metric definition based on a progress variable
- Same resolution as 878M (150 microns) but with 260M cells (**x4.5 speed-up**)
- **Volume/surface adaptation with MMG5.3 and YALES2 2018.11**



[1] Mercier et al., ICNC 2019

Application to a lean-premixed flame

- Comparison of the dynamically adapted results to static cases [1]
- Progress variable iso-contour:



Static
110M static (300µm)

Dynamic
260M dynamic (150µm)

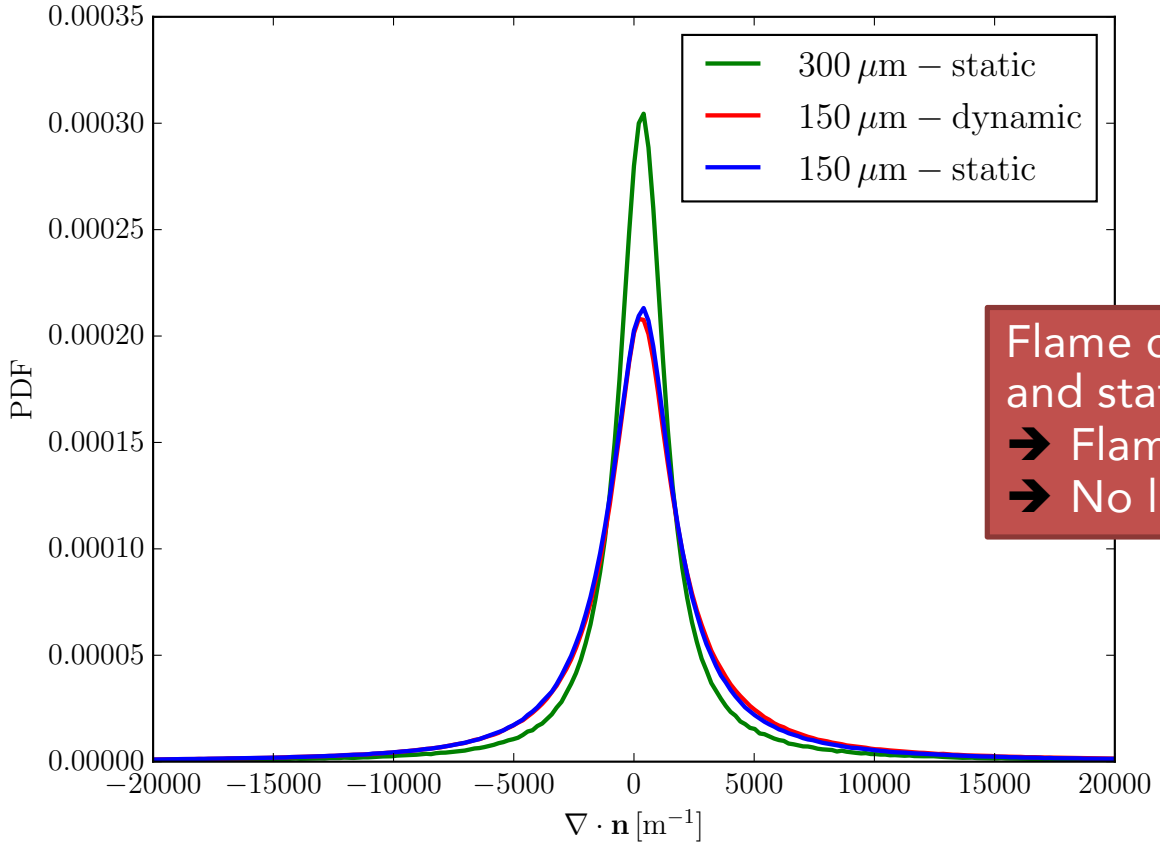
Static
878M static (150µm)

- The small-scale wrinkling depends only on the finest cell size

[1] Mercier et al., ICNC 2019

Application to a lean-premixed flame

- Is the flame wrinkling similar between 150µm **dynamic** and 150µm **static** ?
- Flame curvature distribution computed as the divergence of the isosurface c=0.7 normal

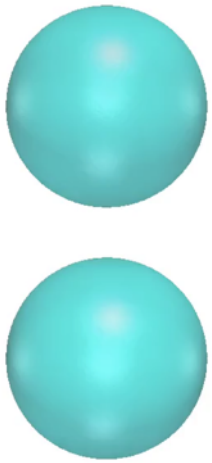


Flame curvature is similar between dynamic and static grids of same resolution
➔ Flame wrinkling is not altered
➔ No lack of generated turbulence

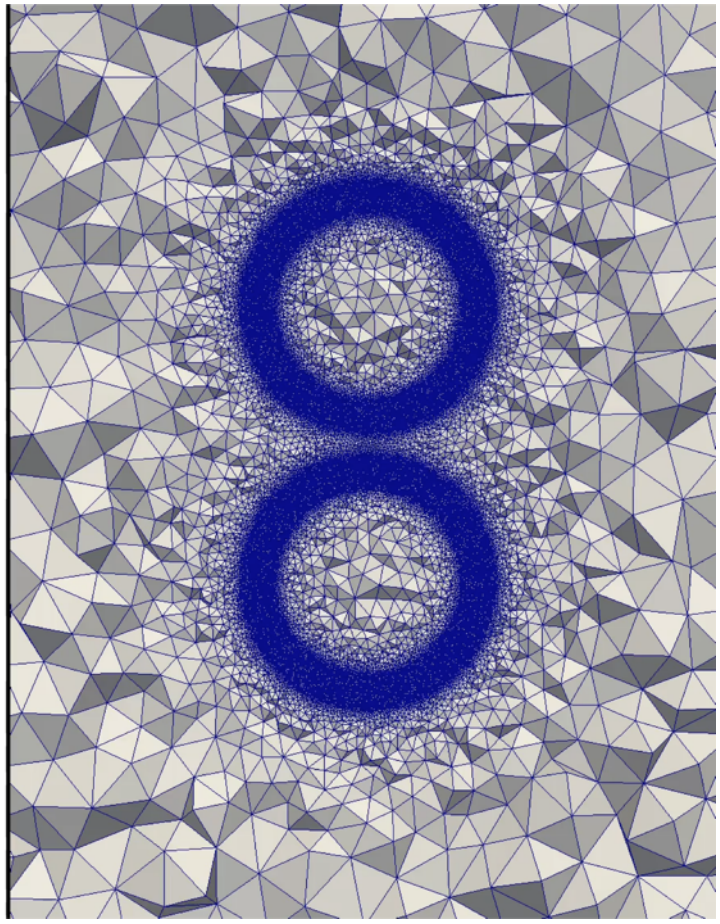
Application to gas/liquid interfaces

- 3D head-on collision of 2 equal-size droplets with reflexive separation [1]
- This is a very challenging case as dynamics is governed by surface tension [2]

YALES2



Time: 0.000000 s



$$\text{Water/Air}$$

$$\text{We} = 23$$

$$\text{Oh} = 0.0047$$

$$\Delta x_{\text{min}} = 6\mu\text{m}$$

$$\Delta t_{\text{adapt}} = 0.35\mu\text{s}$$

$$30\text{M cells}$$

$$420\text{ cores}$$

$$200,000\text{ total iterations}$$

$$10,000\text{ adaptation iterations}$$

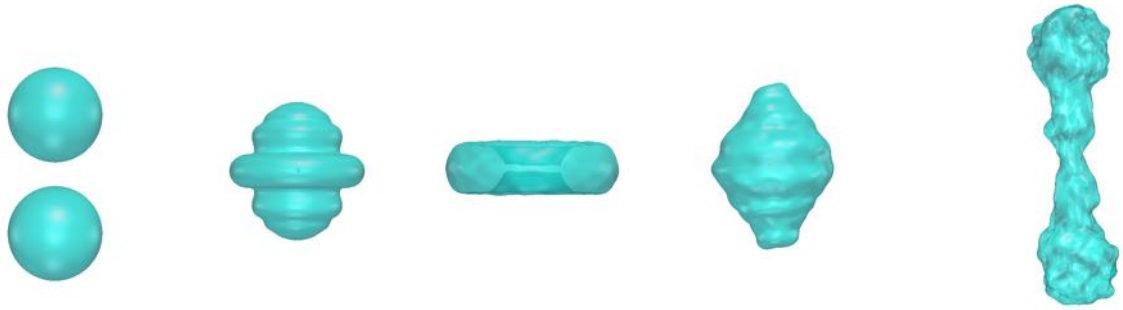
[1] Ashgriz, N., Poo, J.Y., JFM, 1990 [2] Janodet R. et al., ILASS, 2019

Application to gas/liquid interfaces

- 3D head-on collision of 2 equal-size droplets with reflexive separation [1]

Experiment [1]

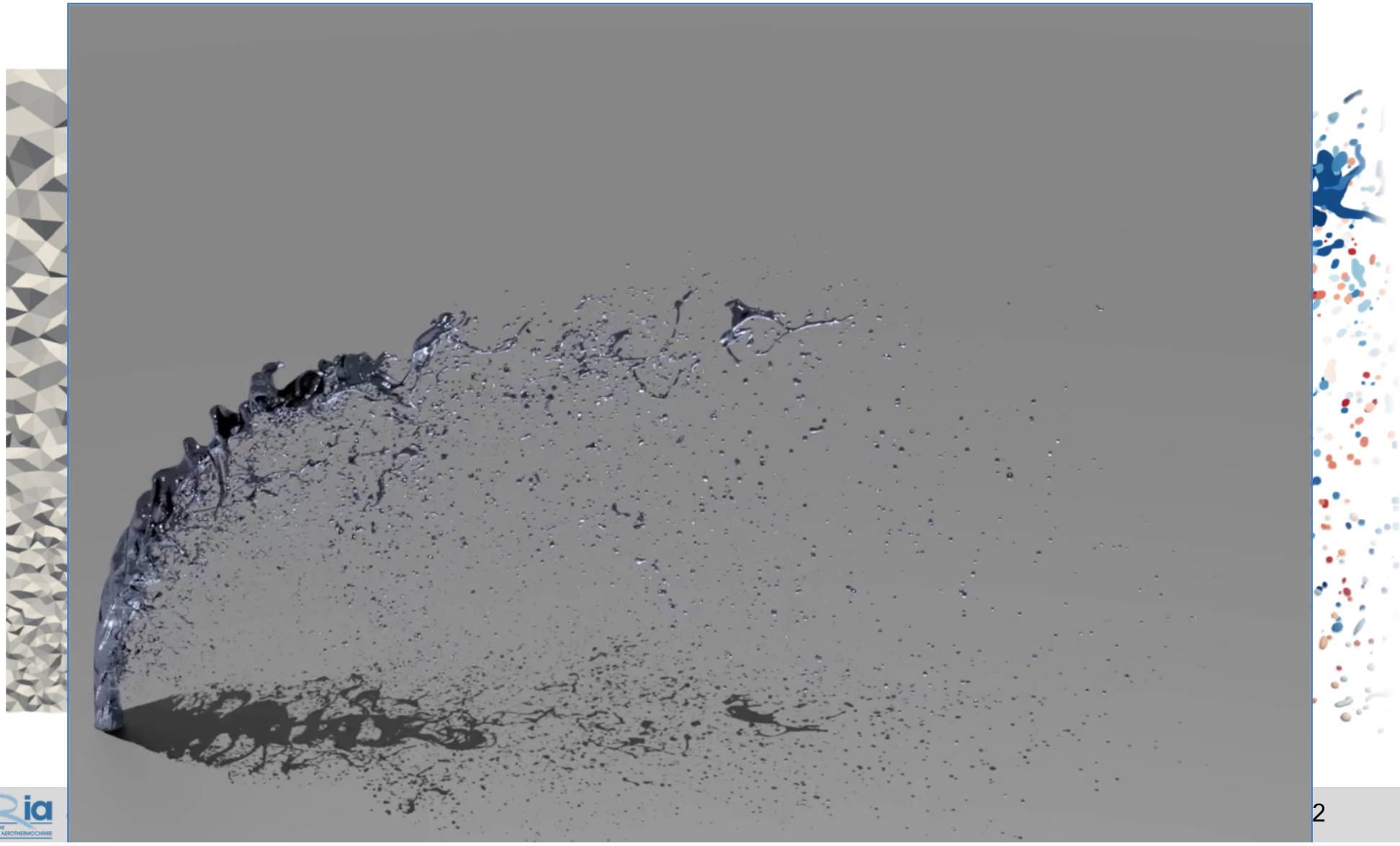
Simulation



[1] Ashgriz, N., Poo, J.Y., JFM, 1990

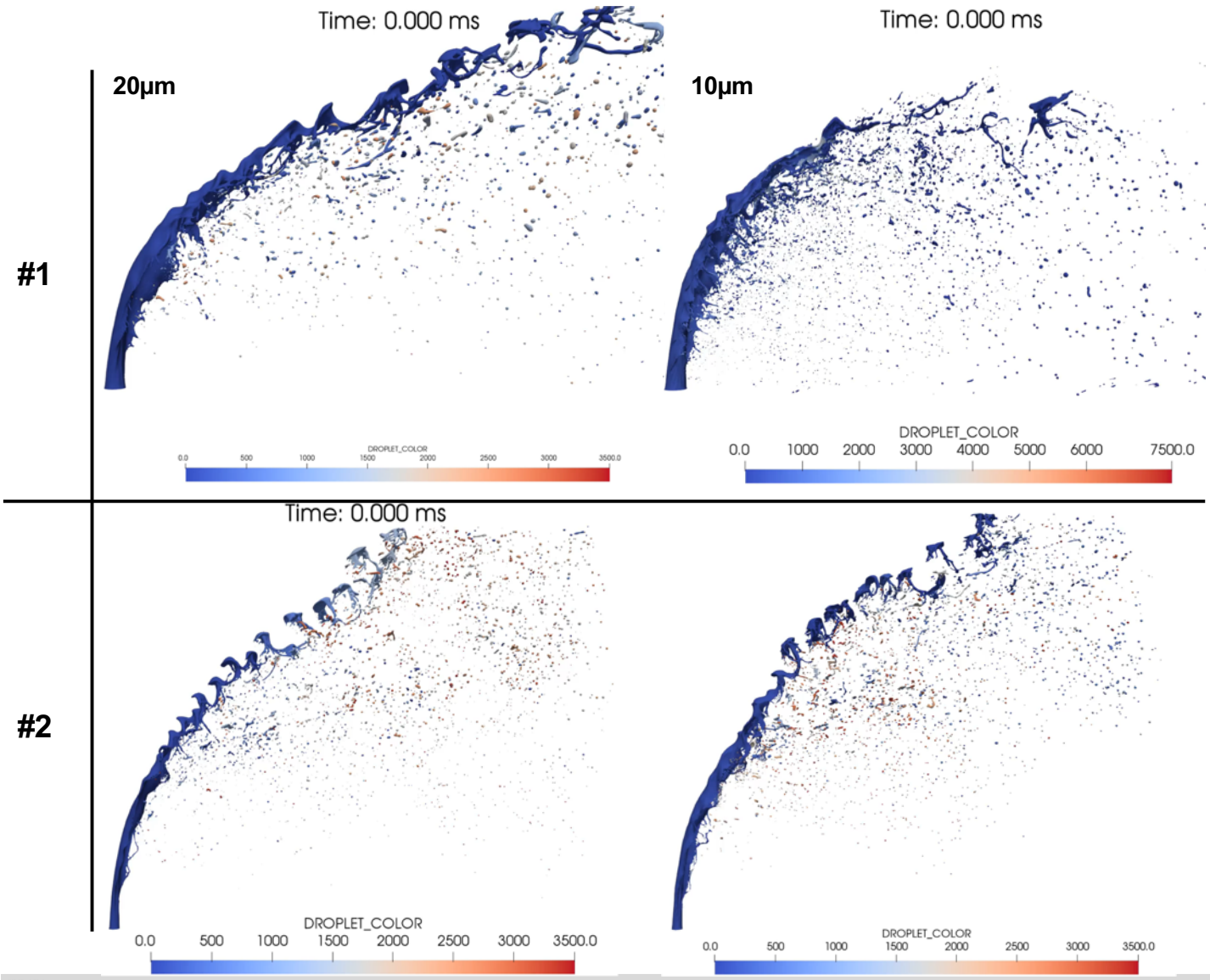
Application to gas/liquid interfaces

- High-Weber number example: kerosene jet-in-cross flow [1,2]
- Up to 1.6 billion tets on 8192 cores



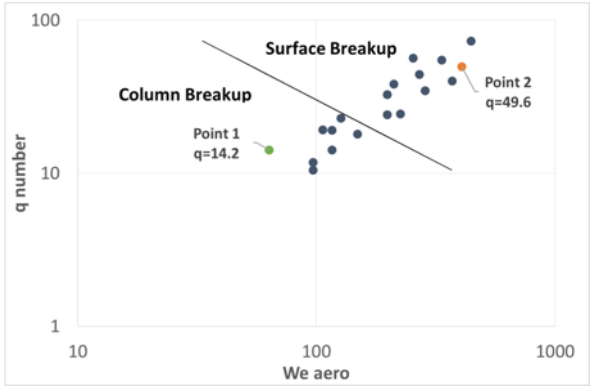
Application to gas/liquid interfaces

- Parametric study of the kerosene jet-in-cross flow [1,2]



$$q = \frac{\rho_l . V_l^2}{\rho_g . V_g^2}$$

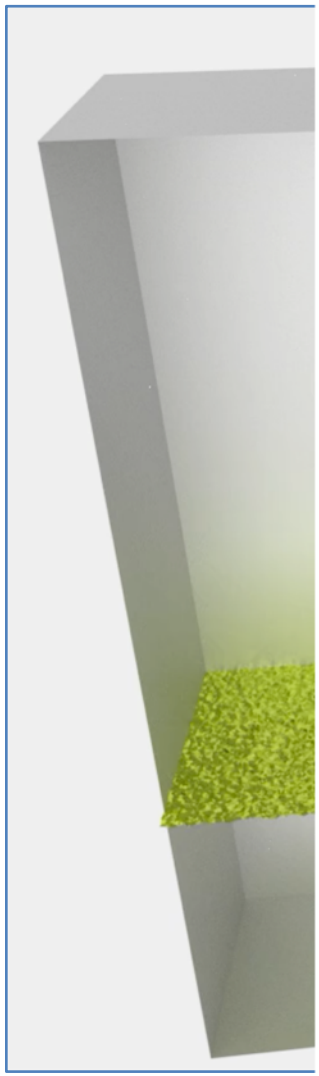
$$We_{aero} = \frac{\rho_g V_g^2 D_{inj}}{\sigma}$$

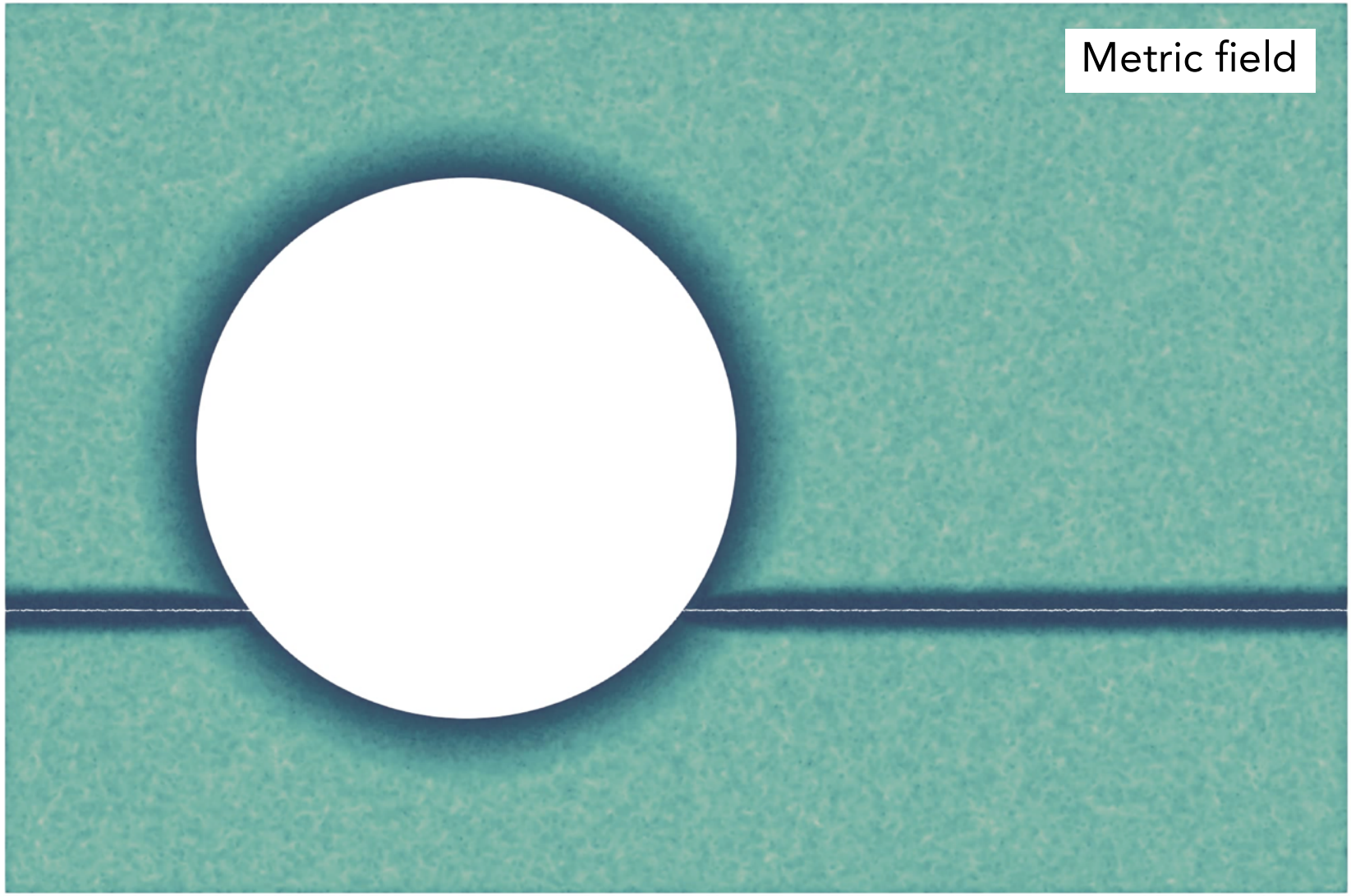


[1] Leparoux et al., ICLASS 2018
[2] Ragucci et al., Atomization and Sprays, 2007

Application to gas/liquid interfaces

- Simulation of oil churning [1,2] by M. Cailler, SAFRAN TECH
- 206 million tets on 1250 cores (Cobalt, CEA), adaptation cost = 42%

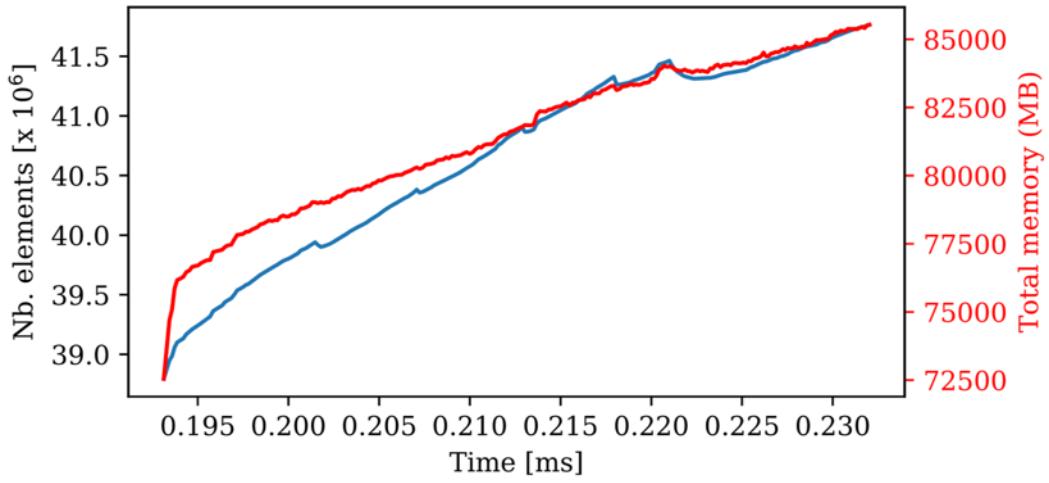


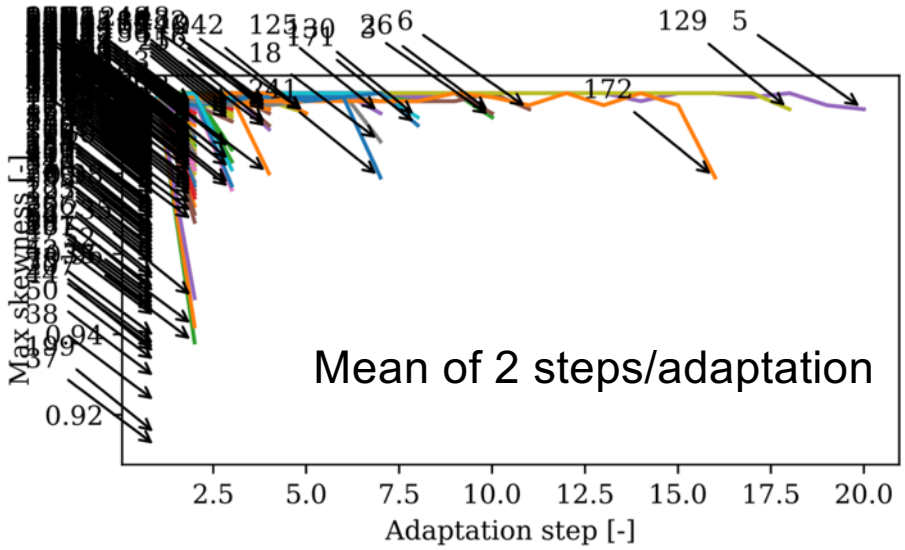


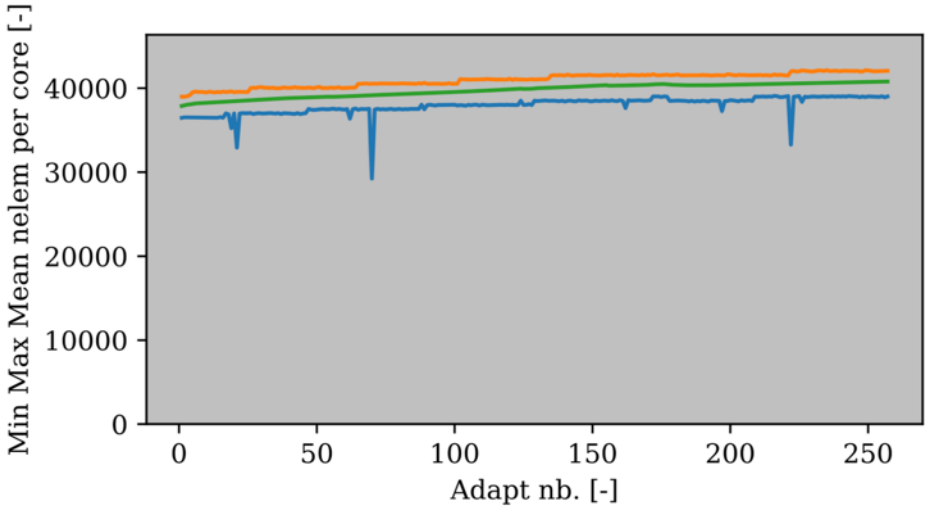
[1] Cailler et al., ICMF, 2019
[2] Leprince et al., Advances in Tribology, 2012

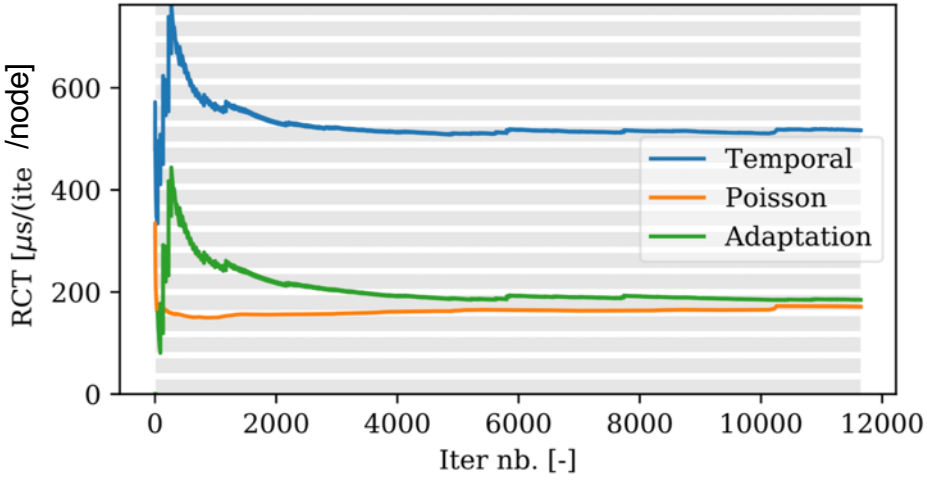
Performances

- Two-phase flow simulation on 1024 cores (Intel Xeon Skylake 2.4GHz)



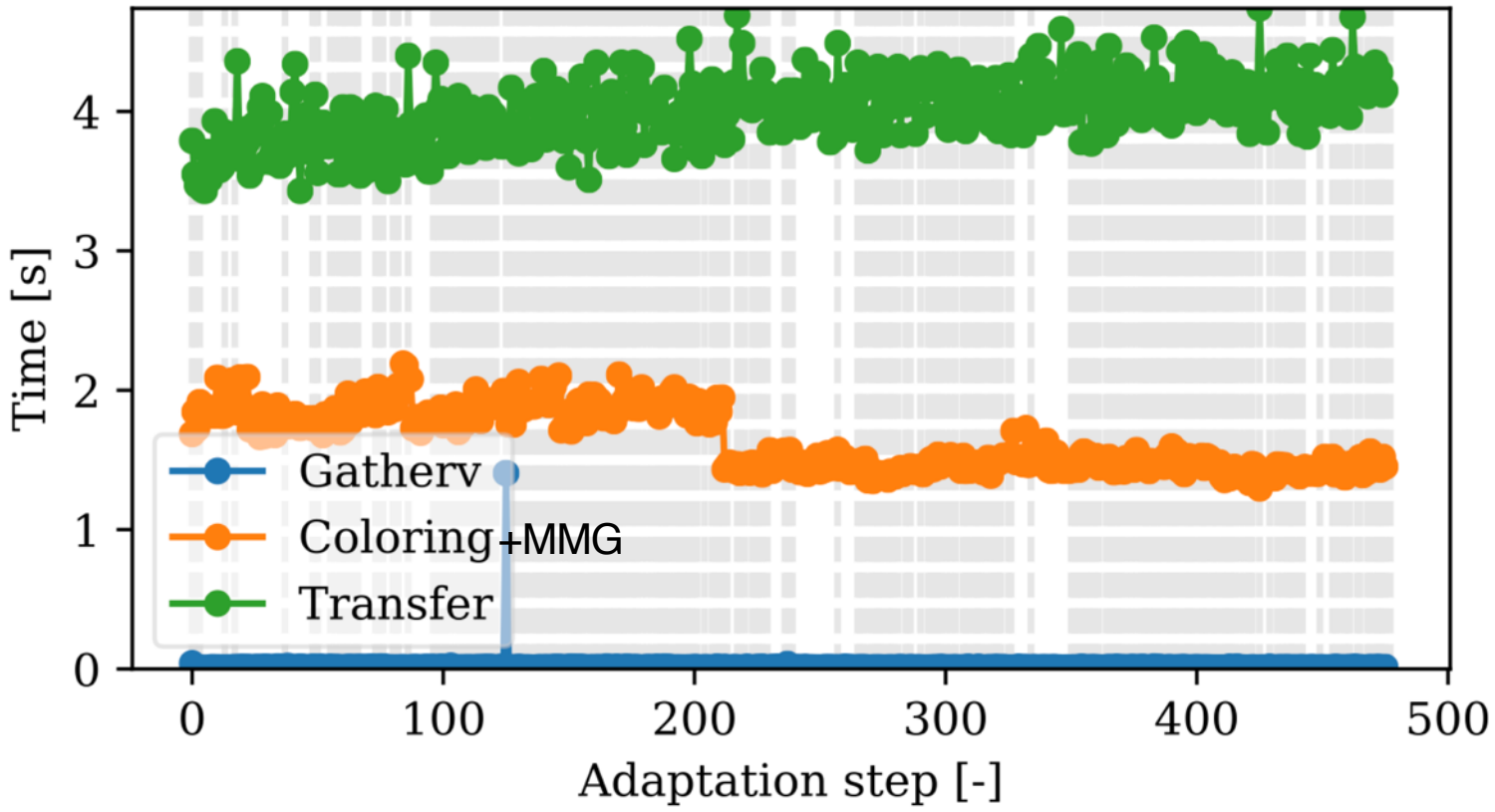






Performances

- Two-phase flow simulation on 1024 cores (Intel Xeon Skylake 2.4GHz)
- Cost of graph coloring+MMG and data transfer in the adaptation steps



600 µs/node

200 µs/node

- Despite packing/unpacking send/recv and fully parallel transfers, the total data transfer cost is around 600 µs/node, which is non negligible

Outline

- Context & Motivation

- Parallel mesh adaptation
  - Principle
  - Load balancing and data transfer
  - Example

- Dynamic mesh adaptation
  - Metric definition and adaptation triggering
  - Examples
  - Performances

- **Improvements to the dynamic mesh adaptation process**

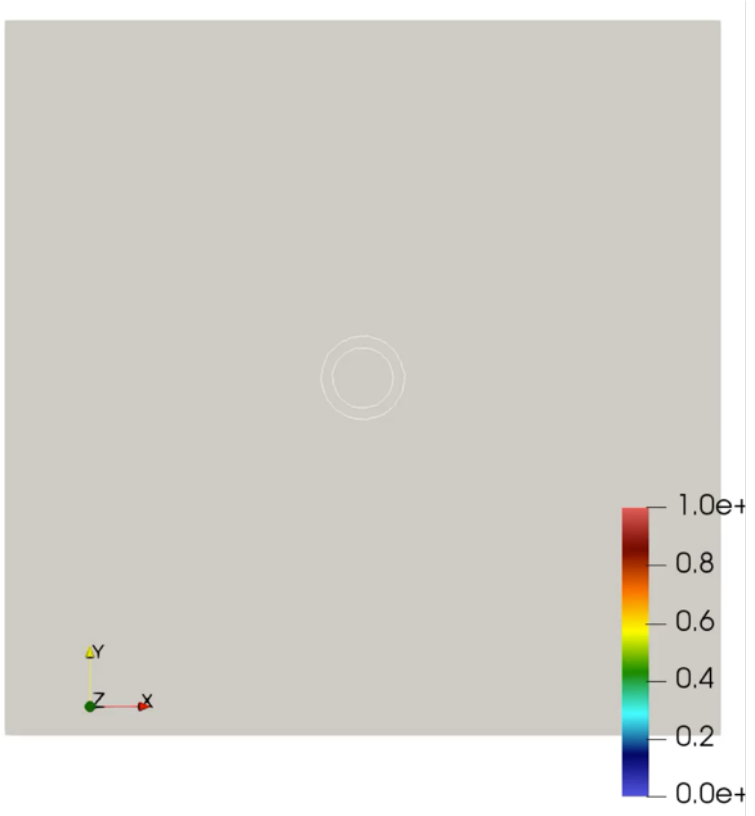
- Conclusions & perspectives

Metric error-based adaptation

- Aim: better control of the adaptation triggering
- Relative metric error

$$\epsilon = \max\left(\left|\frac{\mathcal{M}_{\text{current}} - \mathcal{M}_{\text{target}}}{\mathcal{M}_{\text{target}}}\right|\right)$$

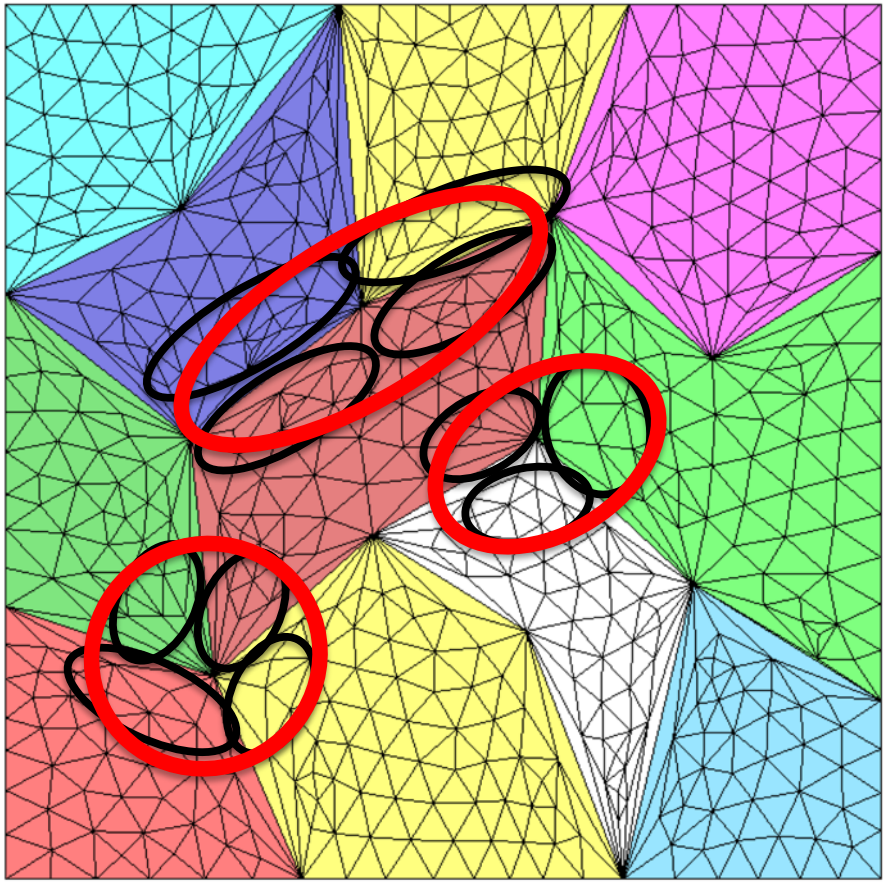
- Triggering: $\epsilon > 0.5$ in the refined region
- This criterion simply states that the metric must keep its value in the refined region
- Example: water droplet convection



Improved error-based load balancing

- The load balancing step can be enhanced by addressing the bad quality cells and the metric errors via agglomeration coloring

- Proposed coloring algorithm
  - Identify all cell groups with bad quality cells and metric error
  - For all flagged cell groups
    - Pick a bad quality cell group and agglomerate with connected other bad quality cell group
    - Stop if large enough
  - For all flagged colors
    - Repeat agglomeration with unflagged connected cell groups
    - Stop if large enough
  - Agglomerate the rest until reaching the number of colors (or ranks)

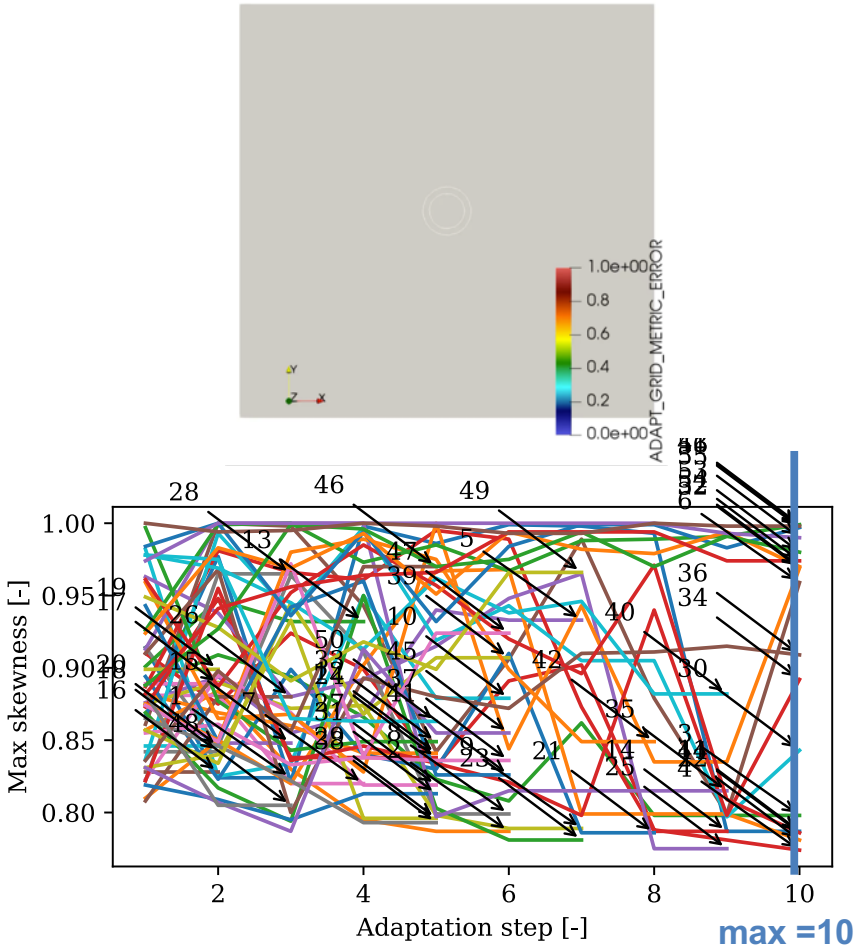


- The mesh is not perfectly balanced but contiguous (if initial mesh is). METIS is still called at the final step to ensure load balance

Performances of new strategy

- Convection of water droplet

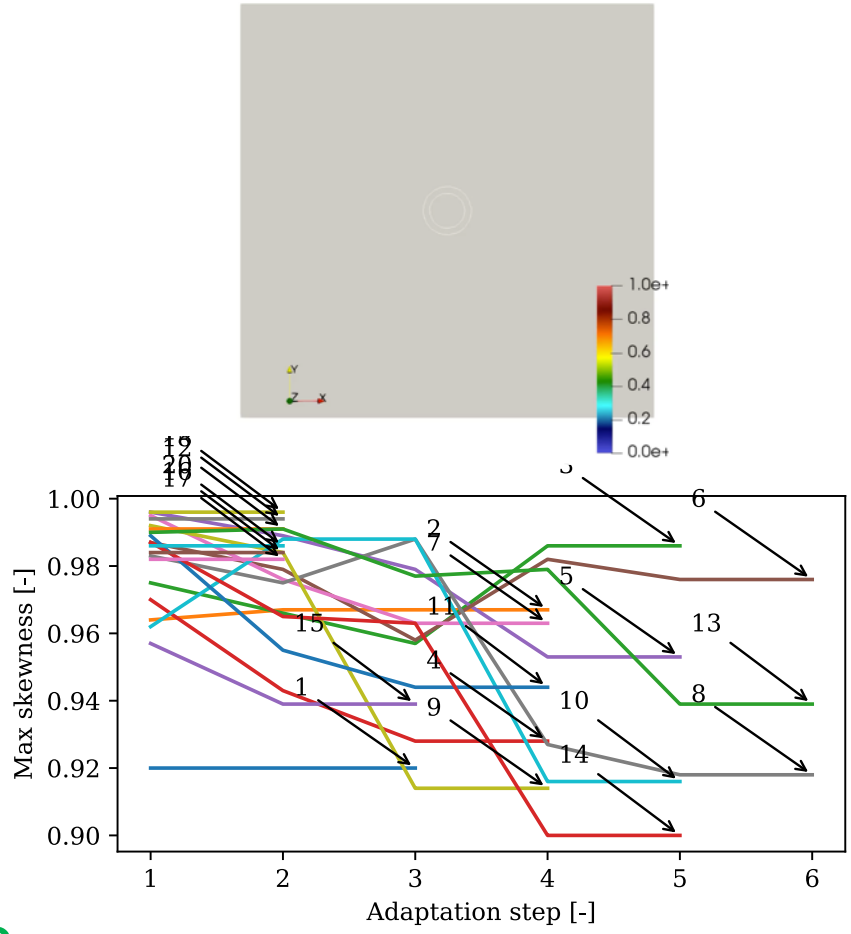
Previous adaptation strategy



Adaptation cost: 90 µs/iter/node

-75%

Metric error trigerring
Error-based load balancing



Adaptation cost: 25 µs/iter/node

Outline

- Context & Motivation
- Parallel mesh adaptation
  - Principle
  - Load balancing and data transfer
  - Example
- Dynamic mesh adaptation
  - Metric definition and adaptation triggering
  - Examples
  - Performances
- Improvements to the dynamic mesh adaptation process
- **Conclusions & perspectives**

Conclusions and prospects

- Dynamic mesh adaptation of material interfaces on distributed memory systems is feasible and is a solution to achieve more accuracy for a given cell count

- It has been employed for many industrial applications in 2018 and 2019 and has proven to be mandatory for certain cases

- Many theoretical and numerical developments are still required
  - Modeling of space/time commutation errors in LES
  - Sub-grid scale models for front capturing methods
  - Performance model for better time control of adaptation
  - Better load balancing algorithms

- A further step could be parallel 4D finite-volume adaptive methods. Does someone has a good 4D simplex-based remesher?

Acknowledgments

- PhDs and postdocs of the YALES2 team
- A. Pushkarev, G. Balarac – LEGI, Grenoble
- CPU hours from PRACE, GENCI and CRIANN