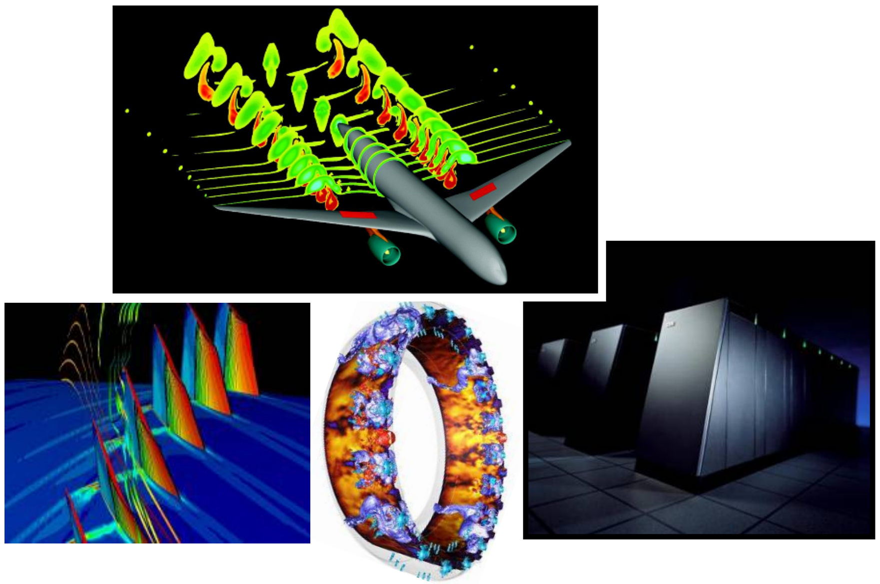
Toward petaflop numerical simulation on parallel hybrid architectures

Large Eddy Simulation and Multi-physics for Engine Computations on Massively Parallel Machines



F. Duchaine, G. Staffelbach,
The CFD TEAM and GlobC Team

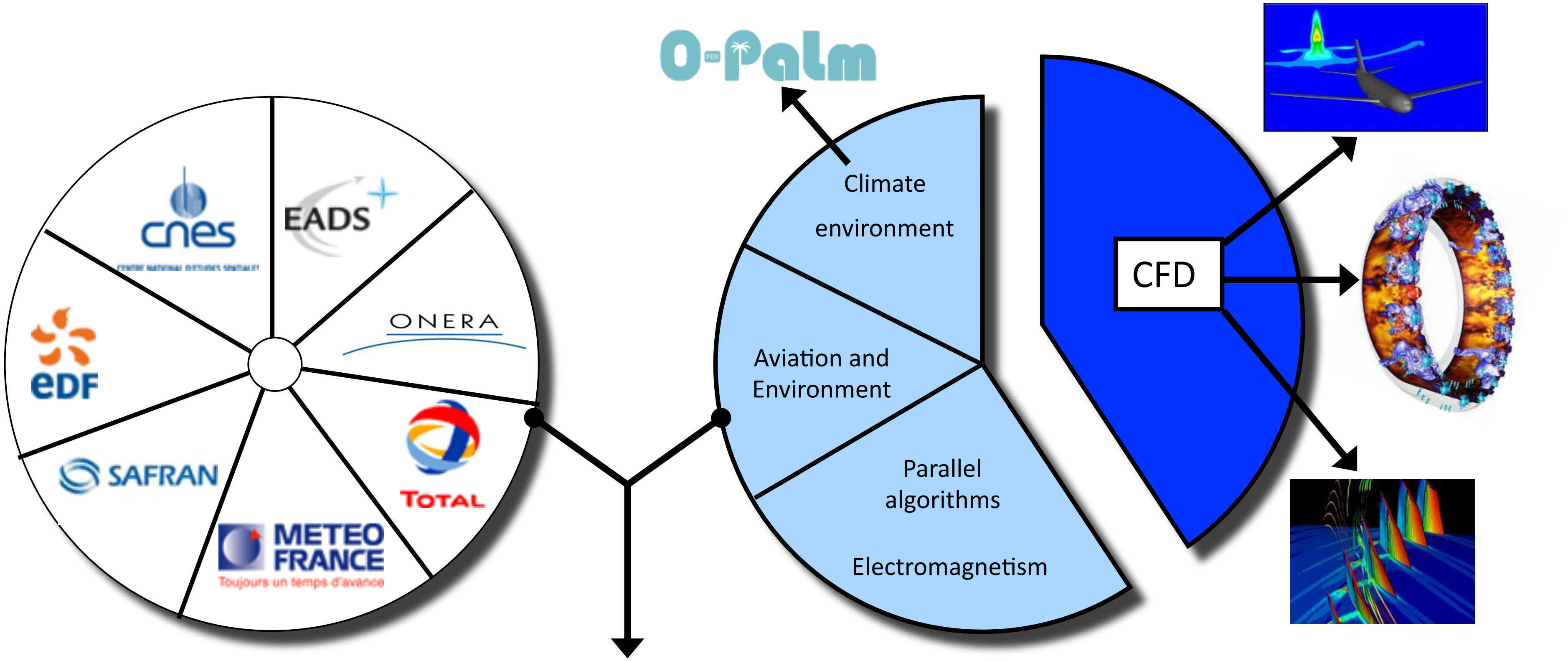
CERFACS, Toulouse

http://www.cerfacs.fr
Florent.duchaine@cerfacs.fr

What's CERFACS?

CERFACS has seven shareholders

One hundred people in 4 teams



- Expertise in scientific computation
- Access to large computational resources

To start, just remember two equations:

ENERGY ON EARTH TODAY = COMBUSTION

COMBUSTION IS PRODUCING MORE THAN 90 PERCENT OF THE ENERGY TODAY. THIS WILL DECREASE... BUT NOT TOMORROW

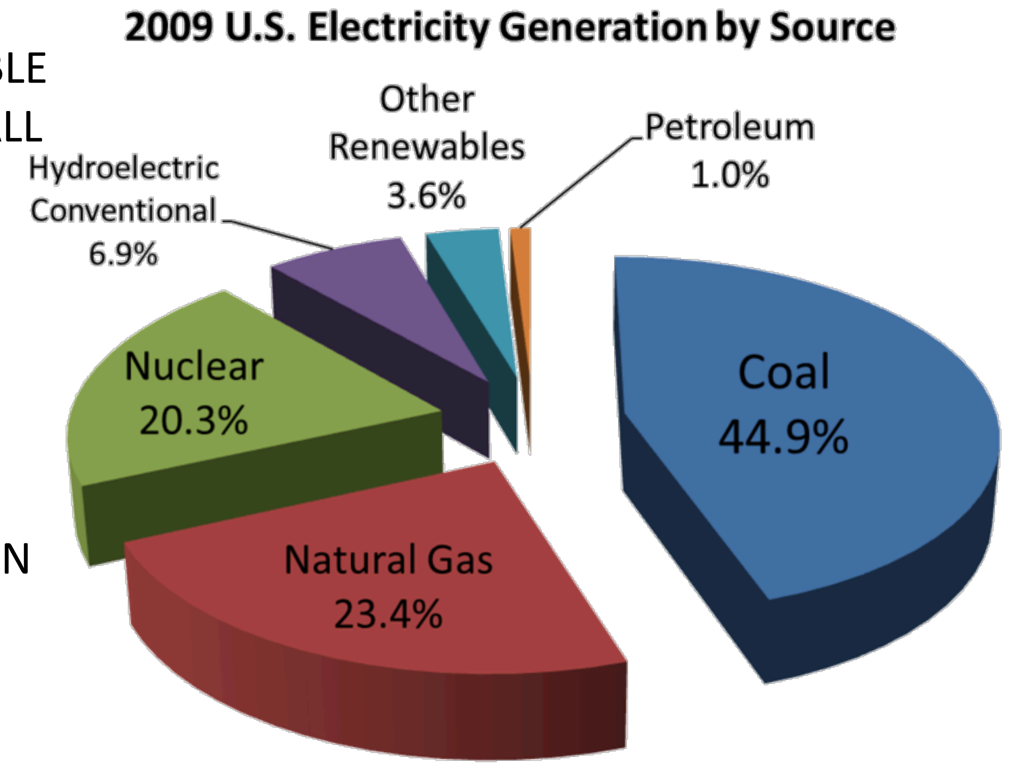
To start, just remember two equations:

ENERGY ON EARTH TODAY = COMBUSTION

ENERGY ON EARTH TOMORROW = COMBUSTION

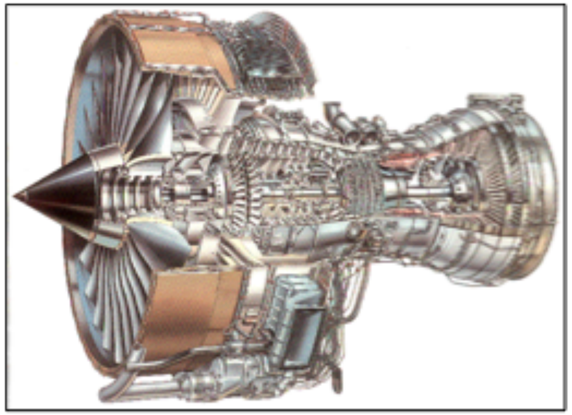
Climate change and energy market: 2010/2030

- TO CONTROL CLIMATE CHANGE, RENEWABLE ENERGIES MUST INCREASE FASTER THAN ALL OTHER SOURCES
- BUT THE GLOBAL DEMAND FOR ENERGY ALSO GROWS (TYPICALLY 2.6%) !
- THE ENERGY PRODUCTION BY COMBUSTION MUST ALSO INCREASE
- COMBUSTION SCIENCE MUST ALLOW THIS WITHOUT INCREASING EMISSIONS, WASTING FOSSIL FUELS OR MAKING CLIMATE CHANGE WORSE (!...)



Combustion is also dangerous

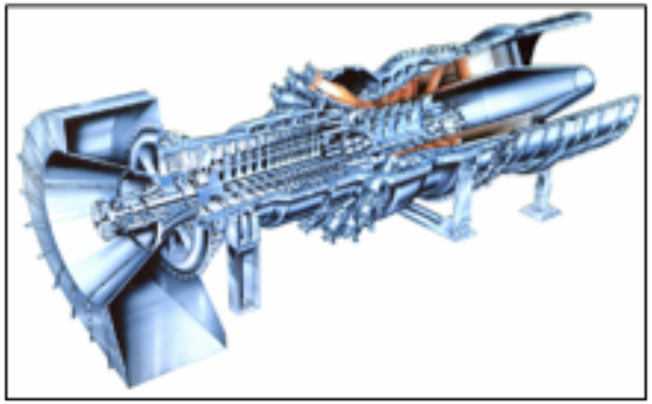
In the global energy strategy, gas turbines have a special role:



1) No other way to proper aircrafts

2) Highly efficient (60%), cheap, flexible system to produce electricity (Dec 2010 in France: no sun and wind -> for each windmill plant, you need a gas turbine)

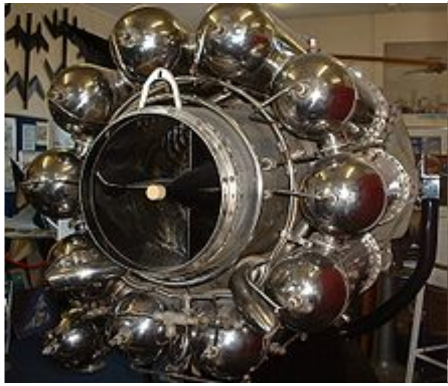
- The gas turbine market grows
- The regulations become tougher: European objectives to reduce pollutant emissions and noise
- Economical constraints: cut the engine cost (today, engines represent 30% of an aircraft cost)



=> Optimization is mandatory

Optimizing something which has been here for 70 years is difficult

• Compromises (efficiency, pollution, noise, stability, cost) are difficult to find and experimental costs too large to test all possible designs

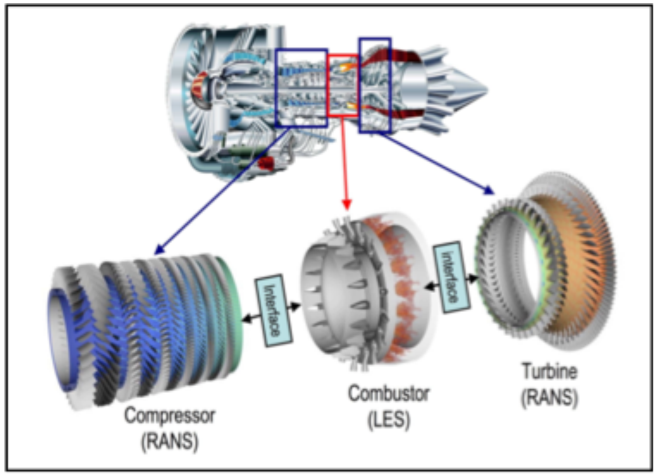


W2 engine

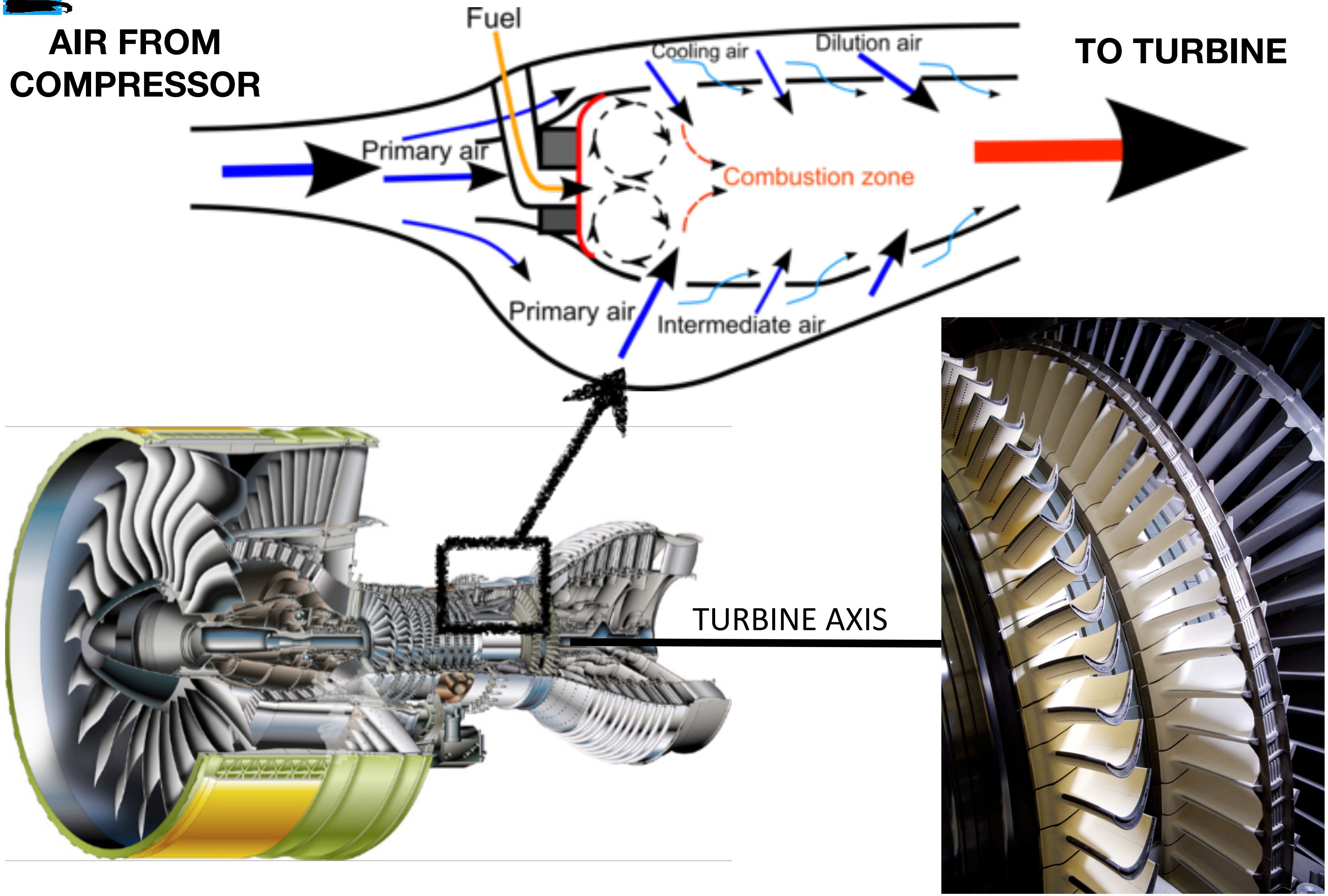
Sir F. Whittle

• Simulations has become essential: gas turbine companies now perform advanced simulations

• Since these simulations must optimize complete real systems, multi-components and multi-physics must be integrated (see ASCI CITS at Stanford)



Copyright Schlüter et al - Stanford CTR


AIR FROM
COMPRESSOR
Fuel
Primary air
Cooling air
Dilution air
Combustion zone
Primary air
Intermediate air
TO TURBINE
TURBINE AXIS

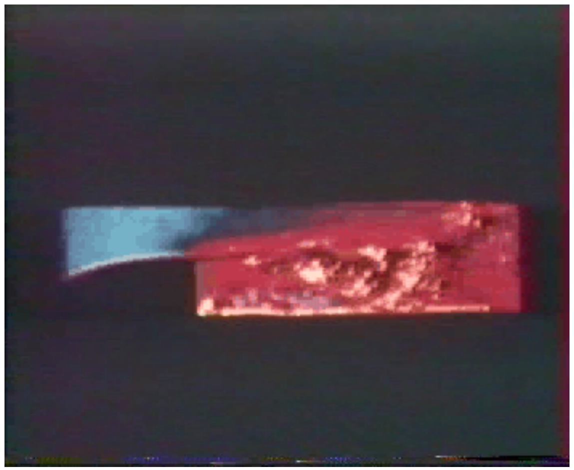
Two things a good combustion chamber should do:

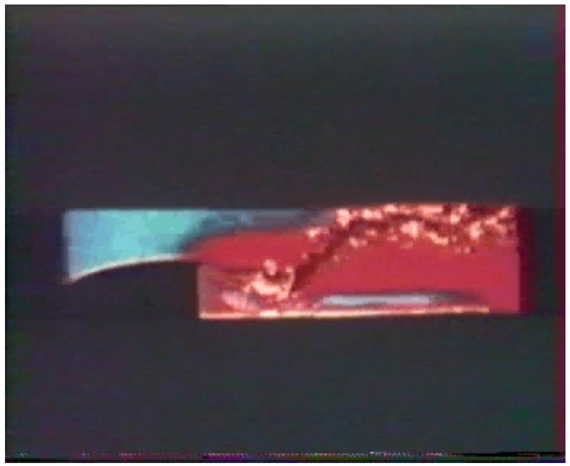
1) Not burn the turbine blades of the high pressure stator

⇒ not easy to do or to predict
⇒ requires combustion + heat transfer + radiation
⇒ need to be precise

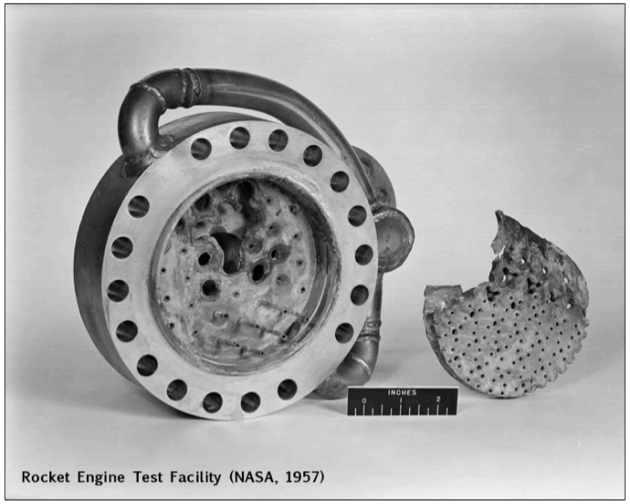
2) Be stable

⇒ difficult due to the coupling between combustion and acoustics (ex of the Berkeley backward facing step)

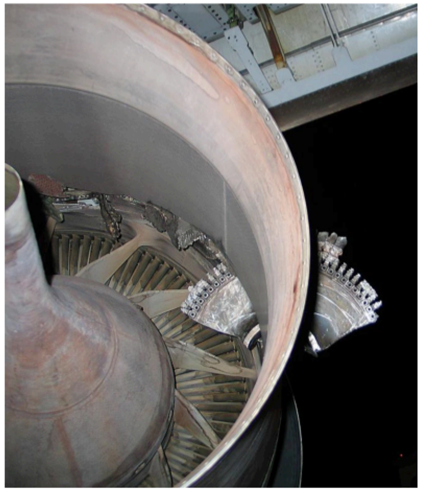


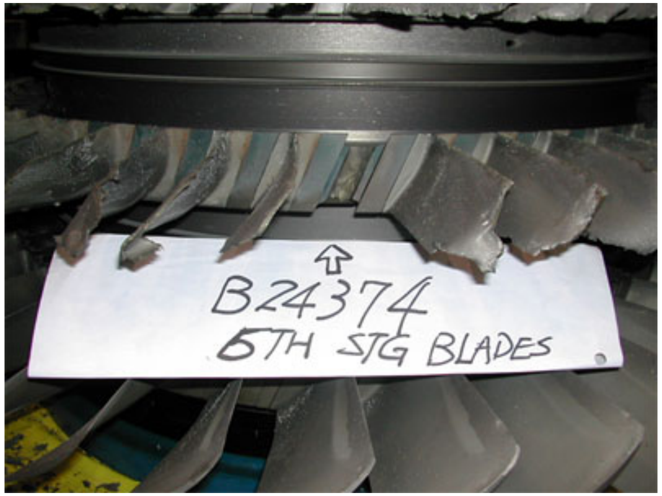


Objective: we want to predict these behaviors before it appends



Rocket Engine Test Facility (NASA, 1957)





Computational Fluid Dynamics & High Performance Computing

⇒ **need to be more precise: get SMALLER, refine the meshes because most important phenomena take place at very small scales (combustion, heat transfer, …) = Single codes with large meshes and CPU time**

⇒ **need to be more global: be LARGER, to compute a full engine with multi-physics and multi-components = multiple codes coupled on the same machine**

⇒ **finally, parametric studies / optimization / sensitivity analysis**

1 Numerical developments in CFD for HPC

- **CFD**
- **Flow solver examples**
- Speed**-up and Mesh-partitioning**
- **Communication, Impact on numerical solutions**
- **Applications to aeronautic challenges**

2 Code coupling

- **Why multi-physics simulations**
- **Physical and numerical issues**
- **HPC issues**
- **Applications to aeronautic challenges**

3 Conclusion and perspectives

Only CPU here, GPU are under investigation at CERFACS for CFD applications

1 Numerical developments in CFD for HPC

- **CFD**
- Flow solver examples
- Speed-up and Mesh-partitioning
- Communication, Impact on numerical solutions
- Applications to aeronautic challenges

2 Code coupling

- Why multi-physics simulations
- Physical and numerical issues
- HPC issues
- Applications to aeronautic challenges

3 Conclusion and perspectives

What is the status of CFD today?

• Computational Fluid Dynamics (CFD) describe the flow behavior, usually based on the Navier-Stokes equations,
• CFD is now an essential tool in industry for design and development,
• Strong industrial demands to tackle more and more complex flow phenomena.

On the one hand:
• CFD investigates modeled physical flows at a lower cost than “pure” experimental methods and can thus help complementing fundamental and industrial developments.
On the other hand:
• CFD is not yet always predictive for most industrial applications (complex geometries, high Reynolds numbers…).

→ Numerical codes require high-end computing platforms: increase resolution + take into account geometrical complexity

→ The term High Performance Computing (HPC) usually refers to (massively) parallel processing (also used as a synonym for supercomputing).

The physical limit of CFD: turbulence and the large range of flow scales

Aeronautical flows have a very high Reynolds number: $Re = \frac{\rho U L}{\mu} \Rightarrow N \propto (0{,}1\,Re)^{9/4}$

- Aircraft at cruise conditions:

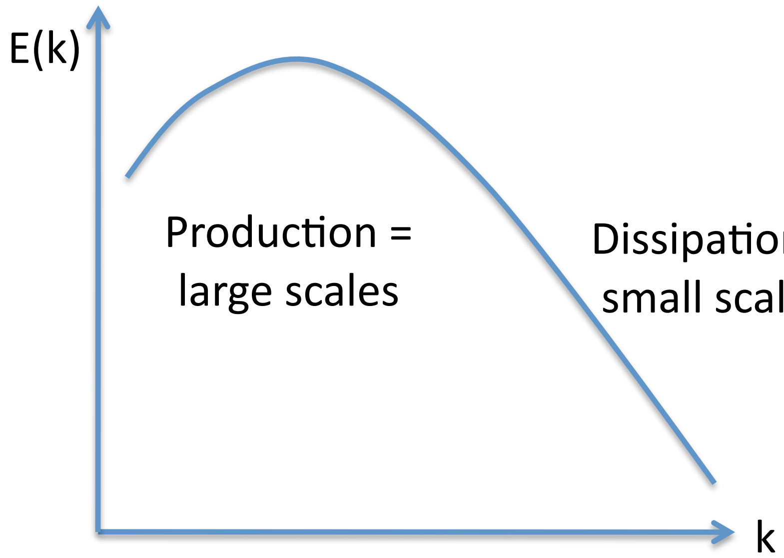
- Compr
- Combu

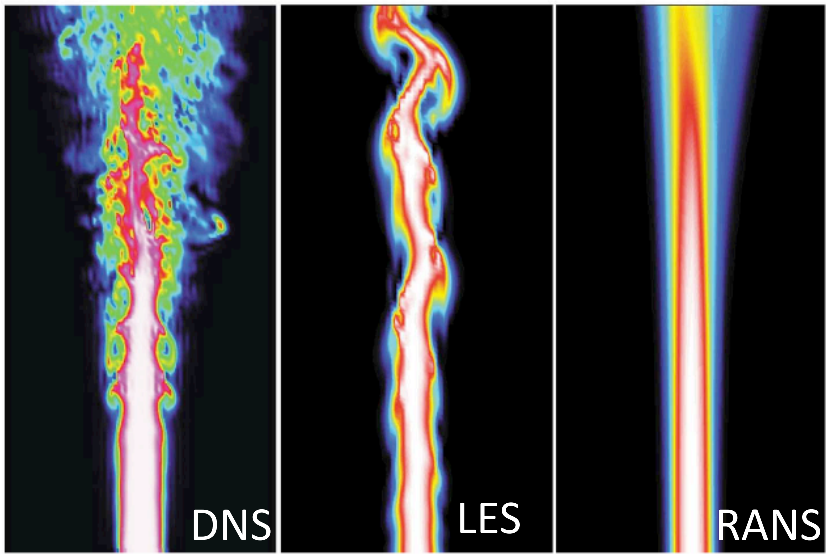
- Turbine

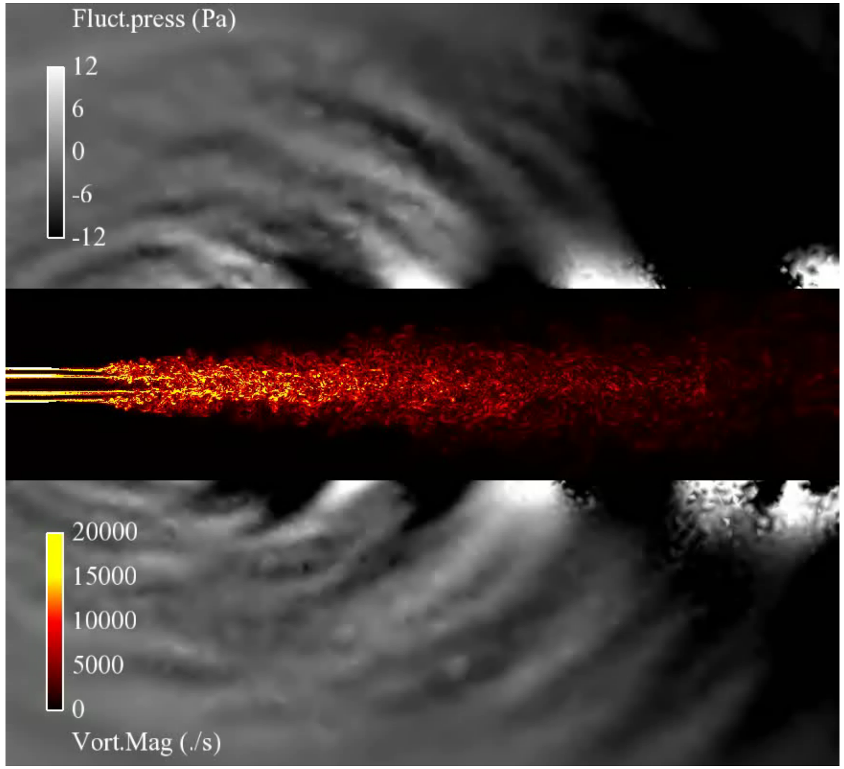
You need to do something to your set of governing equations to allow descent computing effort and take care of turbulence

Flow / Turbulence modeling

Turbulence:



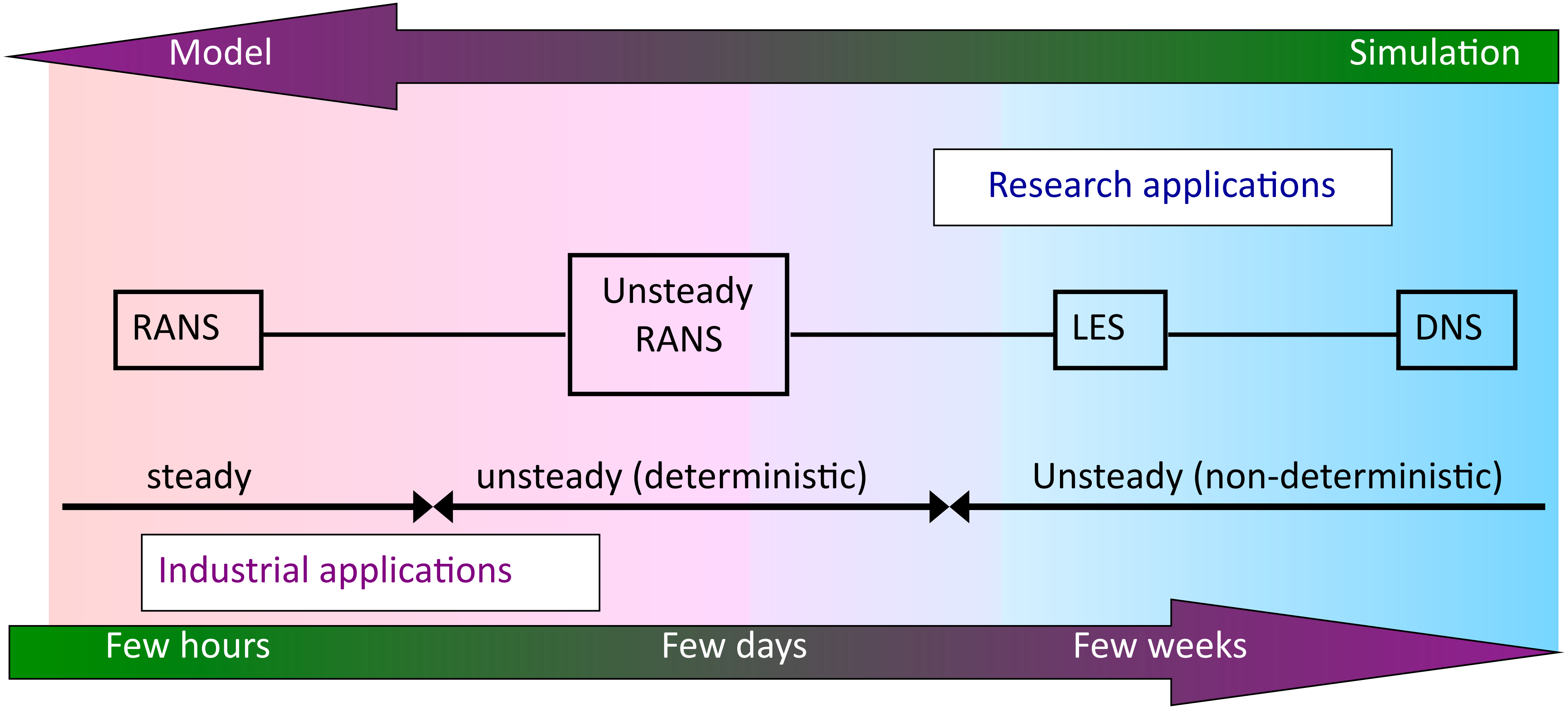




N. Lamarque (CERFACS)

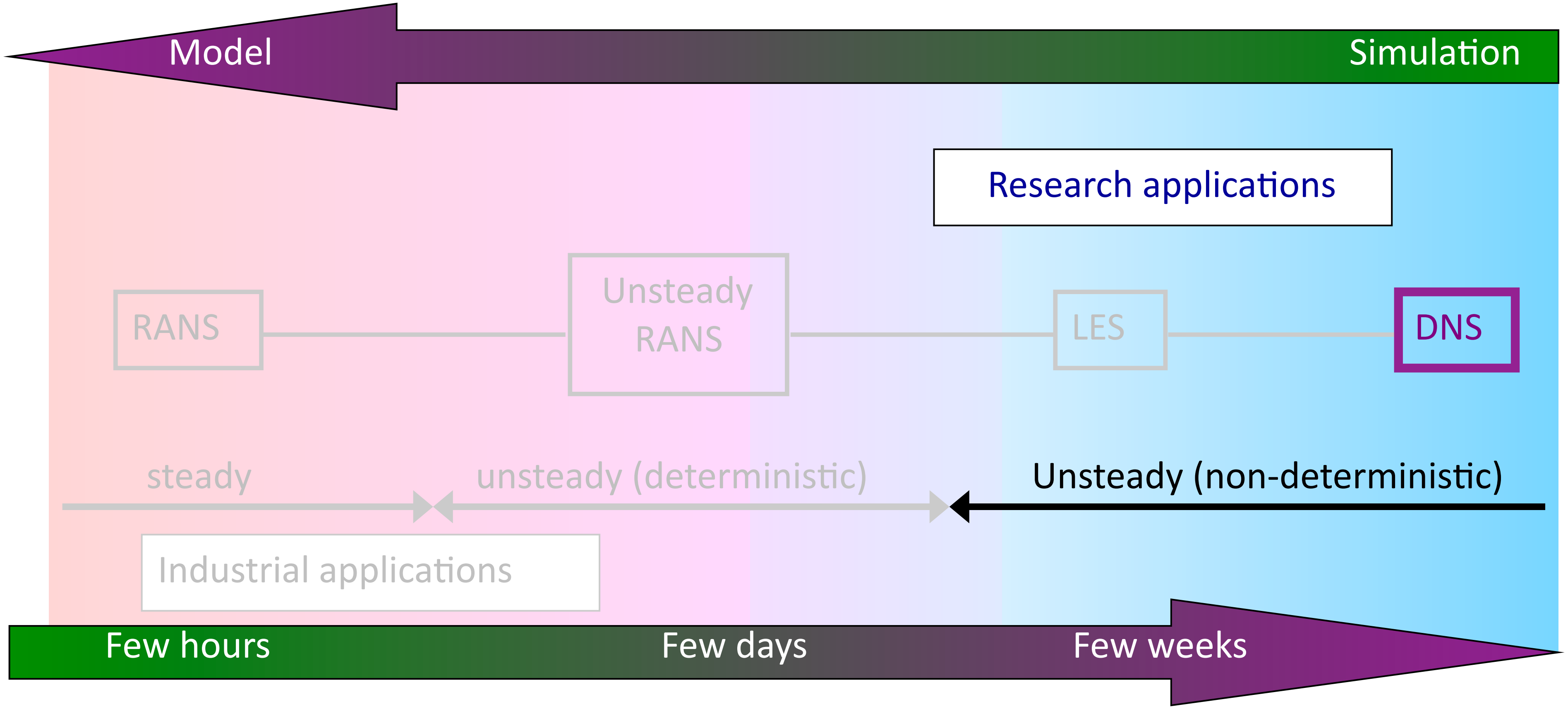
RANS: Reynolds-Averaged Navier Stokes
LES: Large Eddy Simulation
DNS: Direct Numerical Simulation

Overview of the computational methods



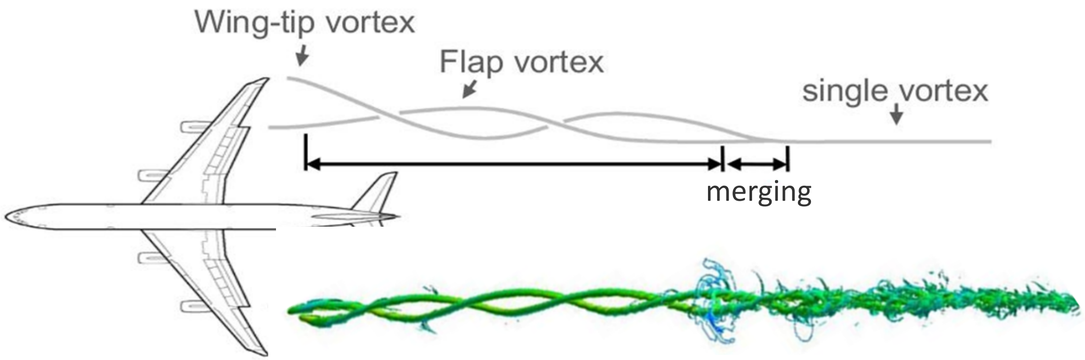
RANS: Reynolds-Averaged Navier Stokes
LES: Large Eddy Simulation
DNS: Direct Numerical Simulation

Overview of the computational methods



RANS: Reynolds-Averaged Navier Stokes
LES: Large Eddy Simulation
DNS: Direct Numerical Simulation

Examples of recent complex flow simulations: DNS

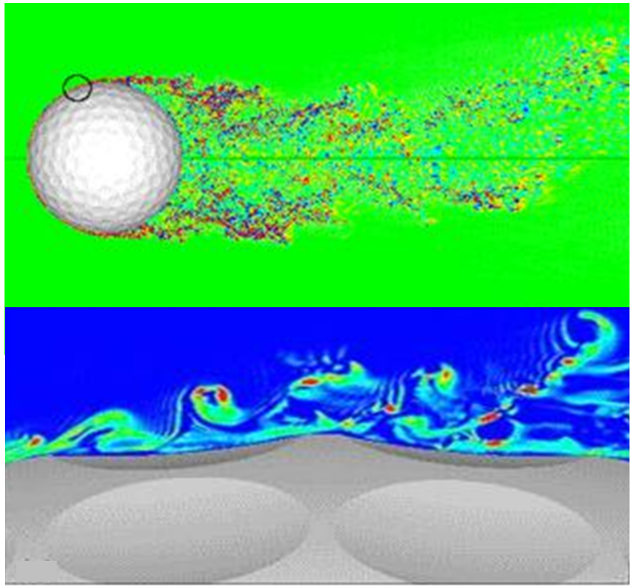


Simulation of wake vortex instabilities behind aircrafts (Nybelen et al., 2008):

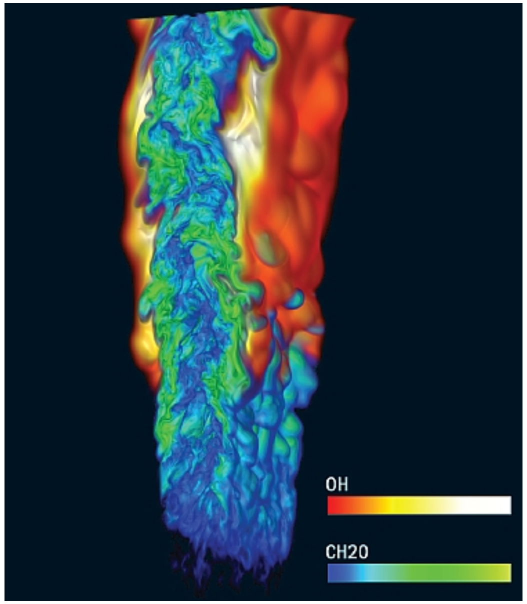
- DNS method (Re=10^4) with NTMIX,
- 110M cells (structured),
- 350 hours with 1024 computing cores (Blue Gene /L).

Flow simulation around a dimpled sphere (Smith et al., 2008):

- DNS method (Re=10^5),
- 61M - 1200M cells,
- 300 hours with 500 computing cores.

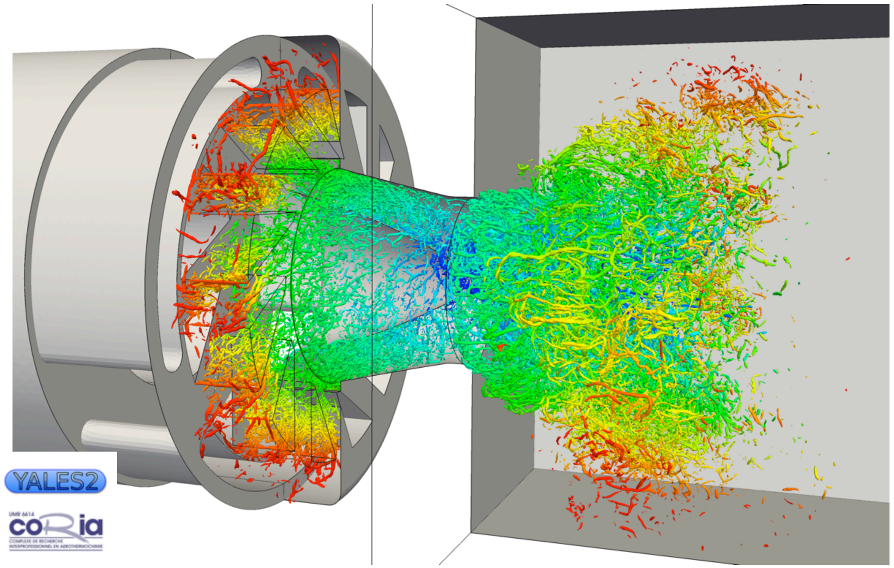


Examples of recent complex flow simulations: DNS in combustion

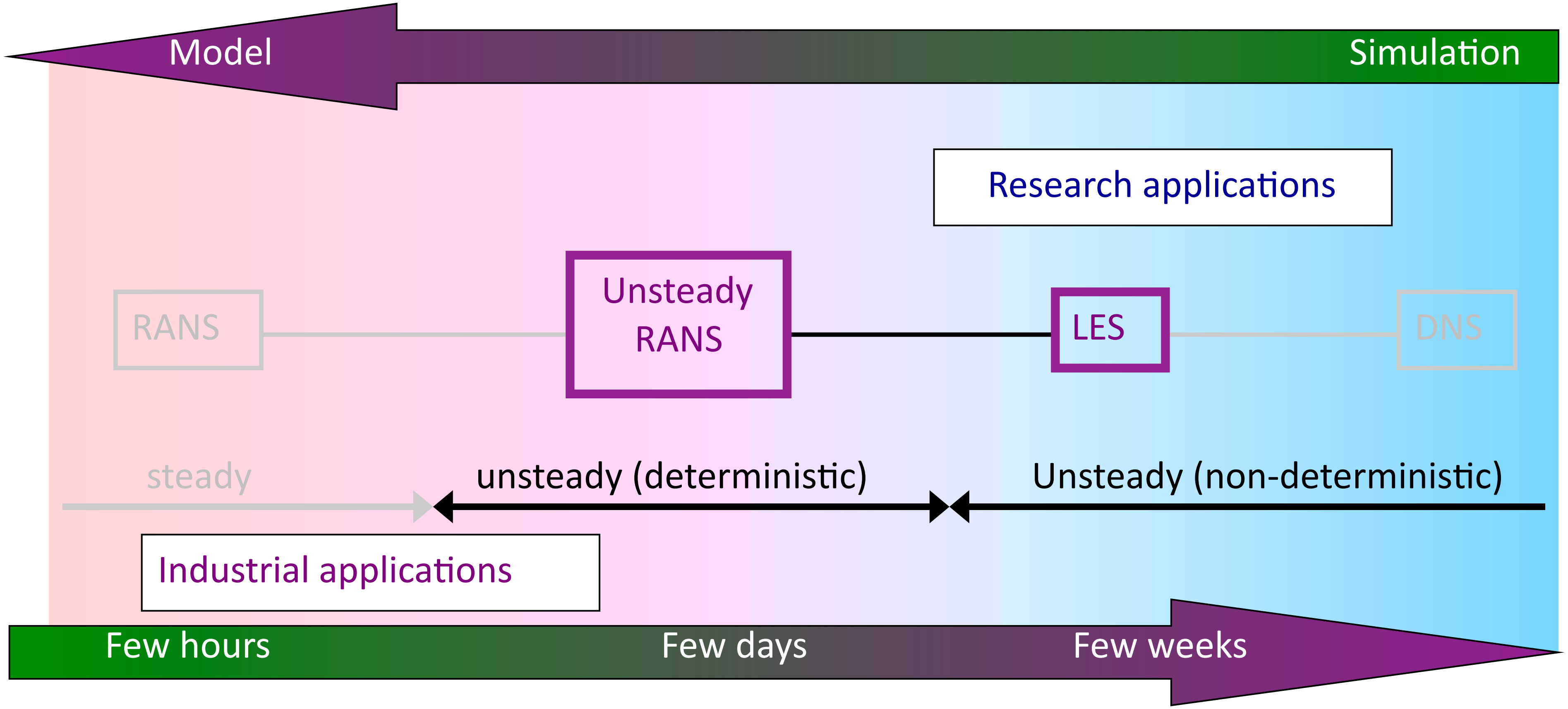


Simulation of a jet flame (Chen et al., 2009):
- DNS method (Re=10^4) with S3D,
- 1.8 B cells (structured), 167 reactions, 22 chemical species
- 250 hours with 30 000 computing cores (Cray XT)

Simulation of a simplified burner (Moureau et al., 2010):
- DNS method (Re=4.10^4) with YALES2,
- 2.6 B cells (unstructured)
- 80 hours with 16 400 computing cores (IBM blue Gene/P)



Overview of the computational methods

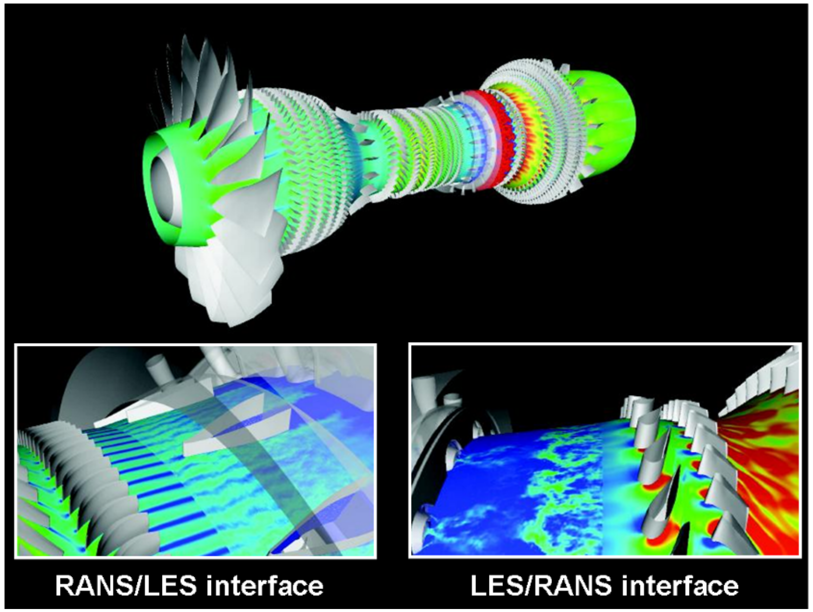


RANS: Reynolds-Averaged Navier Stokes
LES: Large Eddy Simulation
DNS: Direct Numerical Simulation

Examples of recent complex flow simulations: URANS/LES

Tip vortex noise simulation in a wind turbine (Arakawa et al., 2005):

- LES method,
- 320M cells (structured),
- 300 hours with 112 vector cores.



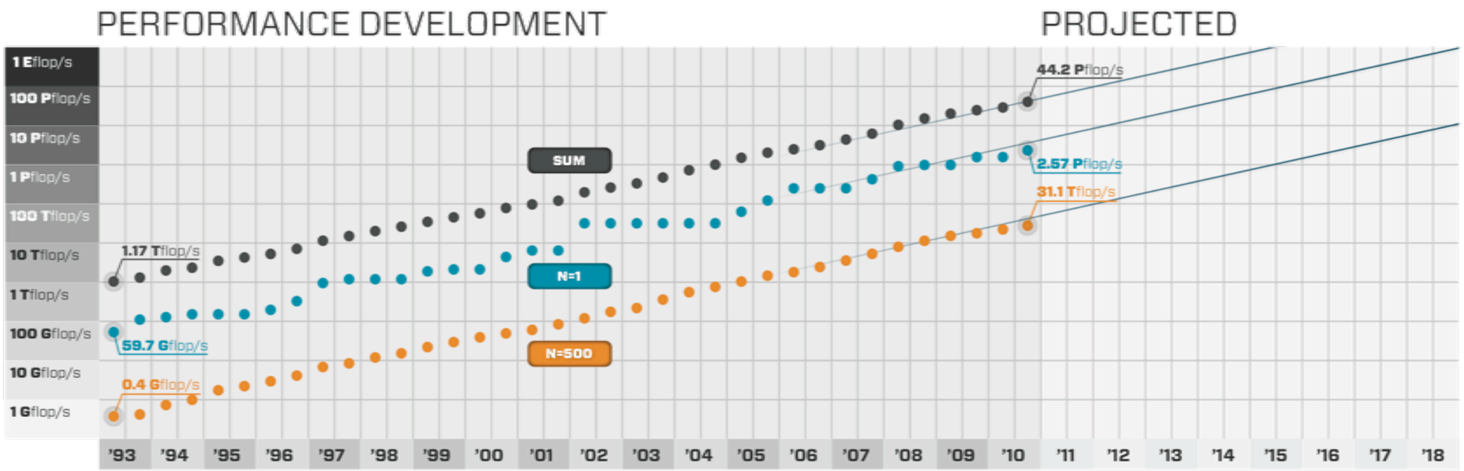
Whole gas turbine flow simulation (van der Weide, 2008):

- RANS/LES coupling method,
- 350M cells (unstructured/structured),
- 2600 hours with 1024 computing cores.

Overview of the most powerful computers in the world

- Numerical methods, such as DNS/LES/URANS are known since a long time,
- But enough computing power is available since only few years to apply them for industrial configurations.

	NAME/MANUFACTURER/COMPUTER	LOCATION	COUNTRY	CORES	R_{max} Pflop/s
1	**Tianhe-1A** NUDT 6-core Intel X5670 2.93 GHz + Nvidia M2050 GPU w/custom interconnect	NUDT/NSCC/Tianjin	China	186,368	2.57
2	**Jaguar** Cray XT-5 6-core AMD 2.6 GHz w/custom interconnect	DOE/SC/ORNL	USA	224,162	1.76
3	**Nebulae** Dawning TC3600 Blade Intel X5650 2.67 GHz, NVidia Tesla C2050 GPU w/ Iband	NSCS	China	120,640	1.27
4	**Tsubame 2.0** HP Proliant SL390s G7 nodes (Xeon X5670 2.93GHz), NVIDIA Tesla M2050 GPU w/Iband	TiTech	Japan	73,278	1.19
5	**Hopper** Cray XE-6 12-core AMD 2.1 GHz w/custom interconnect	DOE/SC/LBNL	USA	153,408	1.05



=> Use the potential of HPC (scalar machines) **for unsteady CFD solvers**

For the specific problem of Gas Turbines

• CFD and Massively parallel computer architectures offer a clear potential for time and cost reductions

• CFD modeling needs to be specifically addressed for the three components to be simulated:

=> Compressor – RANS / URANS / LES

=> Burner – LES

=> Turbine – RANS / URANS / LES

• Each component is the locus of distinct flow physics and adding multi-physics may greatly contribute to the predictions

=> Flow separation and transition

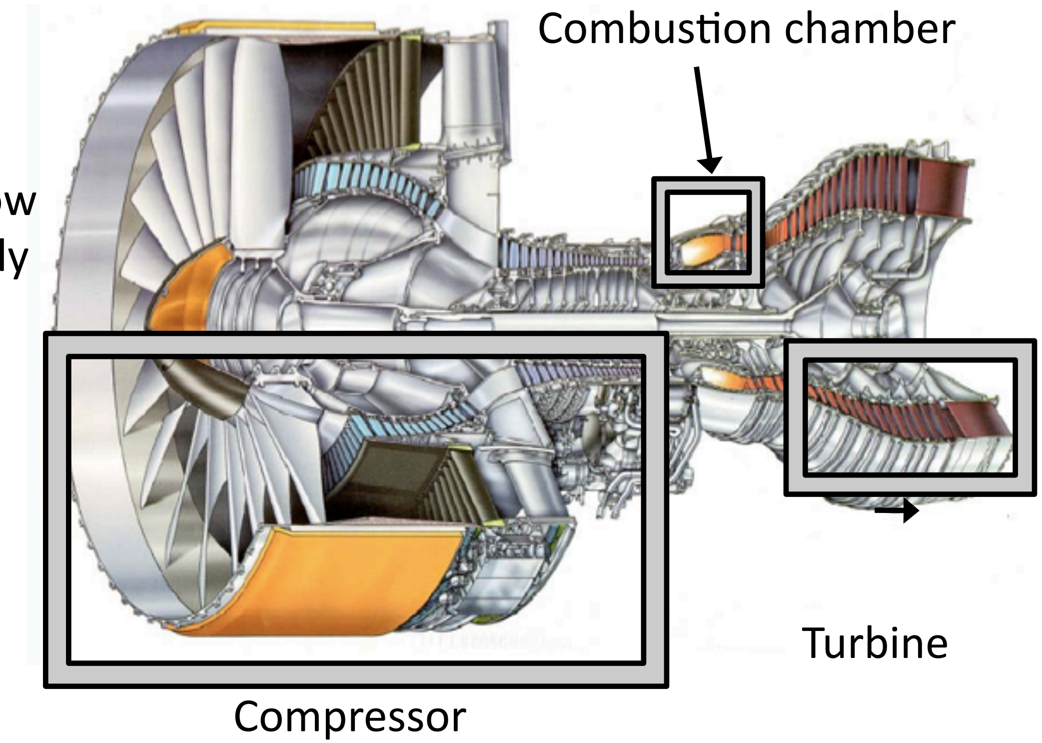
=> Multi-phase flows

=> Chemical reaction

=> Mixing, cooling

=> Heat transfer

…



<u>For the specific problem of Gas Turbines</u>: more computing cores available for

- Reduction in the restitution time (limitation due to efficiency),
- Increase the resolution, the geometrical complexity,
- Increase the size of the domain to take more physical phenomena into account,
- Increase the physical complexity with multi-physic and multi-component simulations

1 Numerical developments in CFD for HPC

- CFD
- **Flow solver examples**
- Speed-up and Mesh-partitioning
- Communication, Impact on numerical solutions
- Applications to aeronautic challenges

2 Code coupling

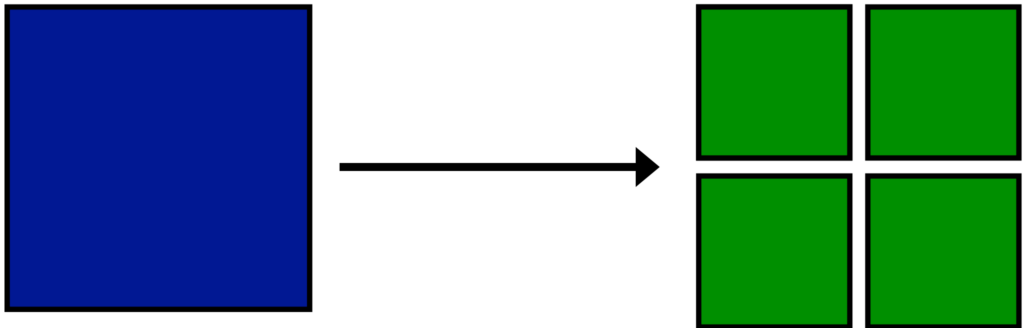
- Why multi-physics simulations
- Physical and numerical issues
- HPC issues
- Applications to aeronautic challenges

3 Conclusion and perspectives

Numerical developments in CFD for HPC – Flow solver examples

<u>What is the impact of massively parallel platforms?</u>

- HPC based on (massively) parallel is a new challenge for CFD flow solvers,
- Problem related to an efficient use of a large number of computing cores:
  - ➢ Mesh partitioning, load balancing, communication?
  - ➢ Impact on flow solvers implementation, numerical solutions?

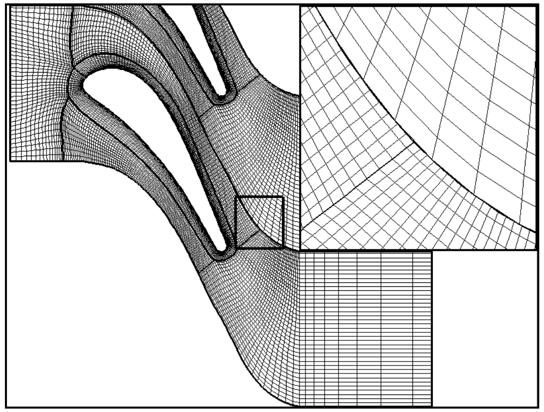


Original problem — New (more) adapted problem

<u>SPMD model:</u>
Single Program / Multiple Data

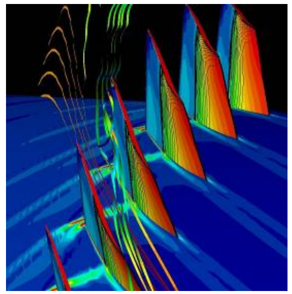
- Multidisciplinary team required for adapting flow solvers to HPC platforms,
- Work performed by scientists and computer experts with background on physics modelling, programming, hardware, HPC… and engineers for providing industrial configurations.

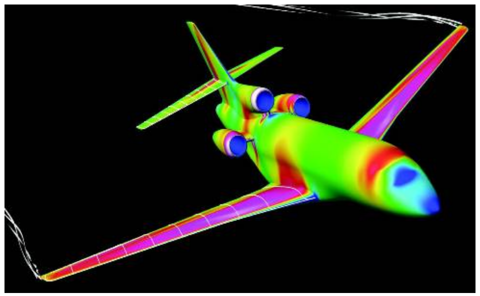
A structured multi-block flow solver: *elsA*



Multi-block structured grid
(Coincident/non-coincident interfaces)

- developed by ONERA[1,2] and CERFACS,
- vector and (massively) parallel capacities,
- *cell-centered approach, implicit in time,*
- Compressible finite volume formulation,
- *External/internal flow simulations and multi-disciplinary applications,*
  (Aerodynamics, aero-elasticity, aero-thermal, aero-acoustics…),
- *(U)RANS/LES and intermediate methods (TSM/DES),*
- *Mono-species (perfect gas or equilibrium real gas),*
- Languages: C++/Fortran/Python,
- SPMD approach.



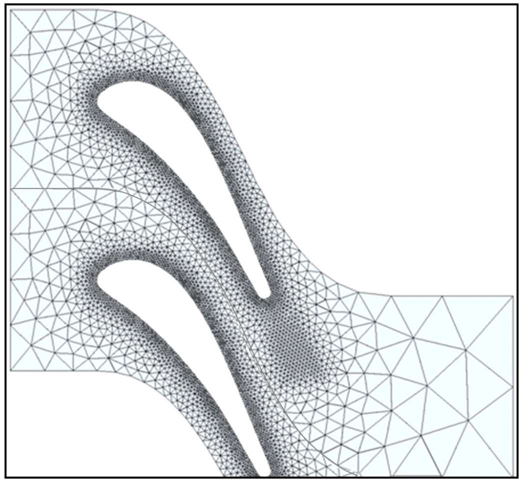


(1) Cambier, 2002
(2) Cambier, 2008

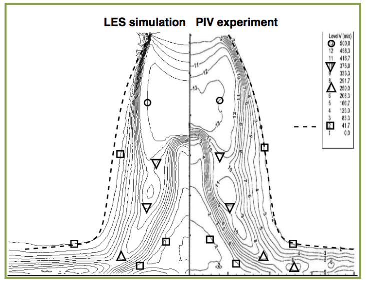
Numerical developments in CFD for HPC – Flow solver examples

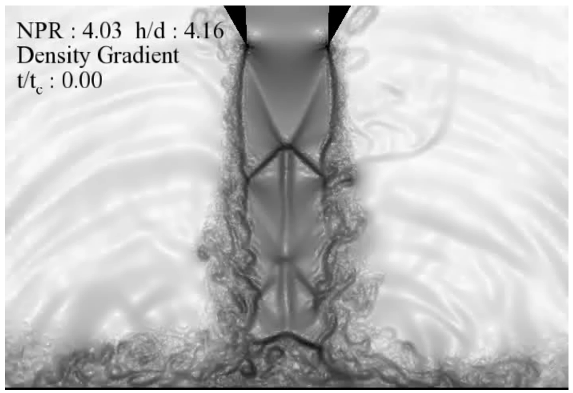
An unstructured flow solver: AVBP

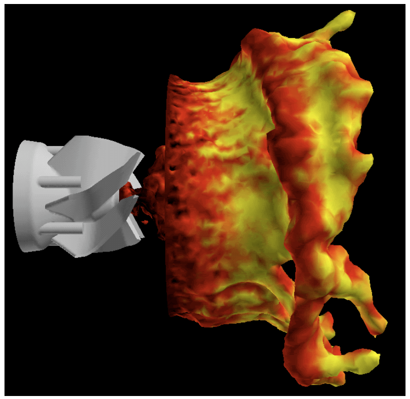
- Developed by CERFACS and IFP,
- External/internal flows,
- *Fully compressible turbulent reacting flows,*
- *DNS/LES approaches,*
- *Unstructured hexaedral, tetraedral, prisms & hybrid meshes,*
- Massively parallel,
- C/Fortran languages,
- SPMD approach.

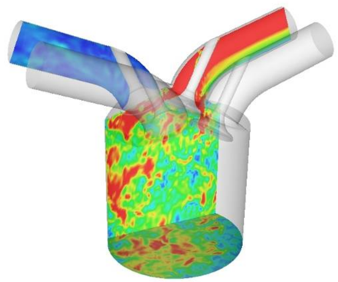


Unstructured grid (Coincident interfaces)









Estimation of the theoretical computing efficiency

- Computing efficiency is related to the computational time/communication ratio, load balancing…

- The speed-up is used to quantify the time reduction related to parallel computing (S=1 is the sequential time, S=N corresponds to a reduction by N…)

- Predict the computational time associated to a (massively) parallel simulation is essential for:
  - estimating the computational cost,
  - managing task scheduling.

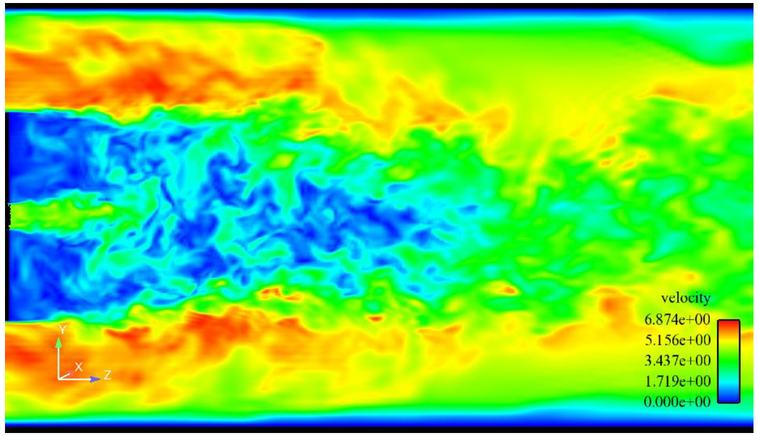
- Different methods can be used:
  - ideal efficiency,
  - Amdahl's law,
  - Extended Amdahl's law.

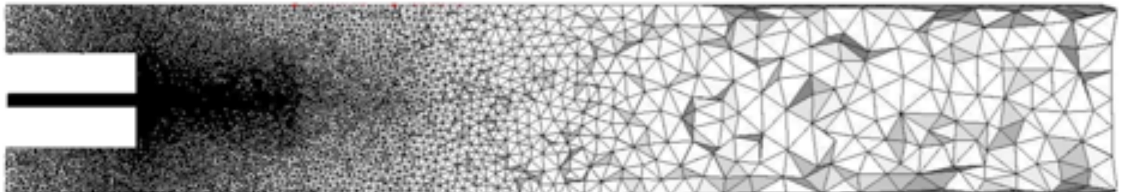
<u>Mesh partitioning for unstructured mesh :</u> splitting algorithms

Different partitioning algorithms (AVBP):

- RCB / RIB: geometric based algorithms,
- RGB: graph theory based algorithm,
- METIS[1]: multi-constraint multilevel graph partitioning.

(1) Karypis et al., 1998

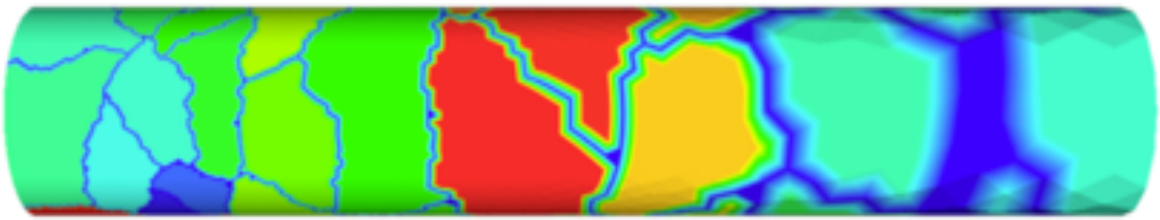




RIB
(32 cores)

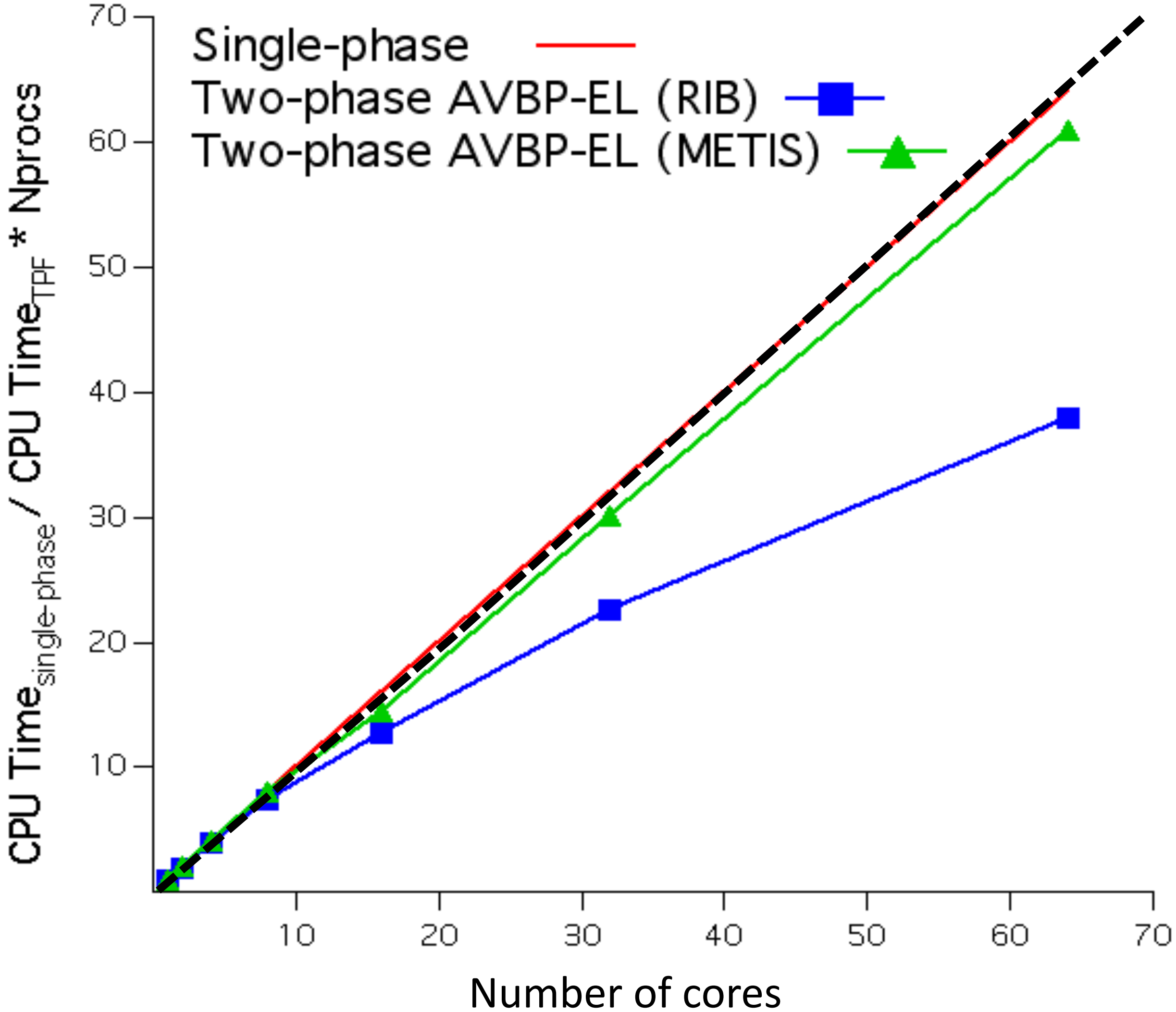
One constraint
(geometry)

METIS
K Way
(32 cores)



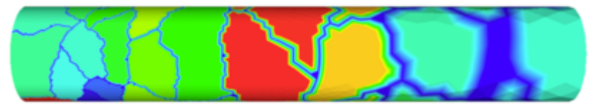
Two constraints
(graph + particles)

Mesh partitioning for unstructured mesh : load balancing



RIB

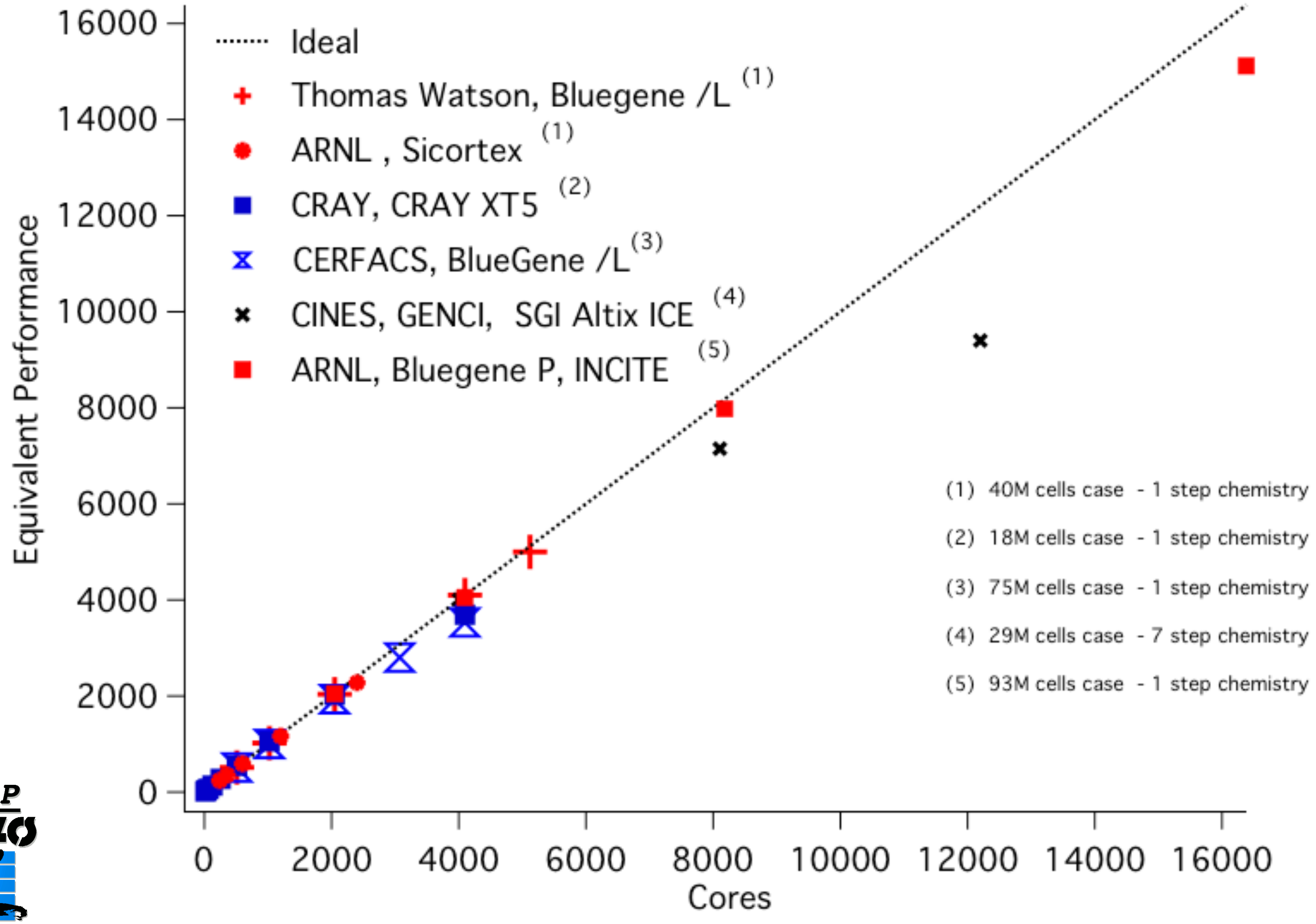
Partitioning on 32 cores



METIS

M. Garcia (CERFACS)

Garcia, PhD, 2009



G. Staffelbach (CERFACS)

1 Numerical developments in CFD for HPC

- CFD
- Flow solver examples
- Speed-up and Mesh-partitioning
- **Communication, Impact on numerical solutions**
- Applications to aeronautic challenges

2 Code coupling

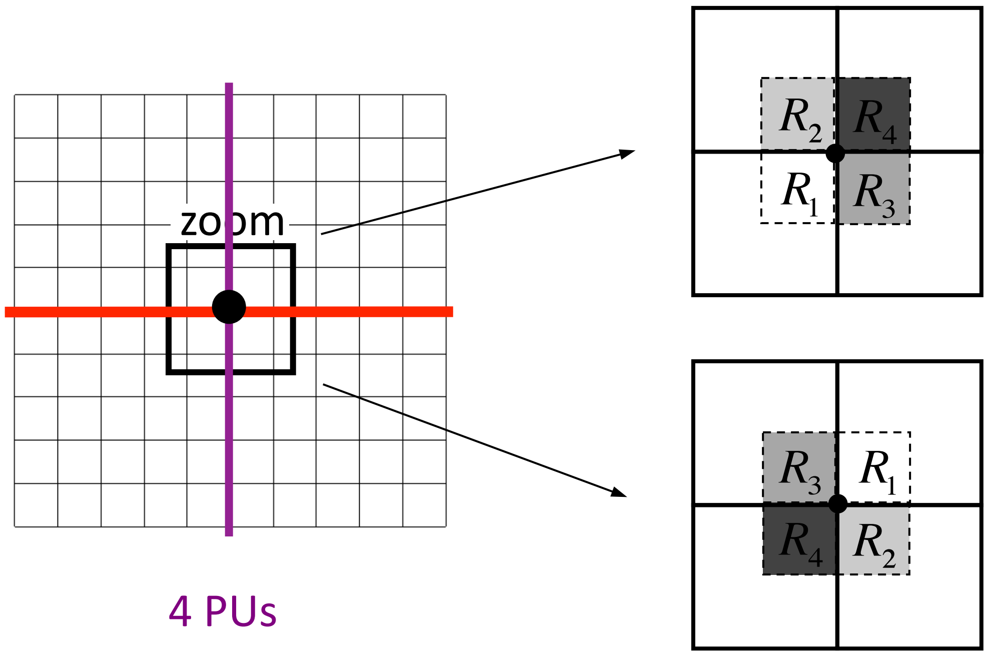
- Why multi-physics simulations
- Physical and numerical issues
- HPC issues
- Applications to aeronautic challenges

3 Conclusion and perspectives

<u>Communication strategy:</u> MPI non-blocking calls

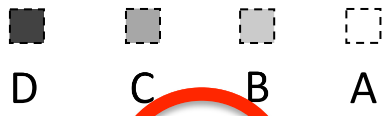
How to compute a residual at partition interfaces?

Like this…



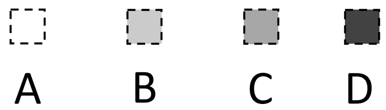
4 PUs

$$\frac{1}{4}\left\{R_4 + \left[R_3 + \left(R_2 + R_1\right)\right]\right\}$$



$$\neq$$

$$\frac{1}{4}\left\{R_1 + \left[R_2 + \left(R_3 + R_4\right)\right]\right\}$$



…or like this…

Problem: non-blocking communications induce a non-deterministic behavior

Impact of rounding errors on LES

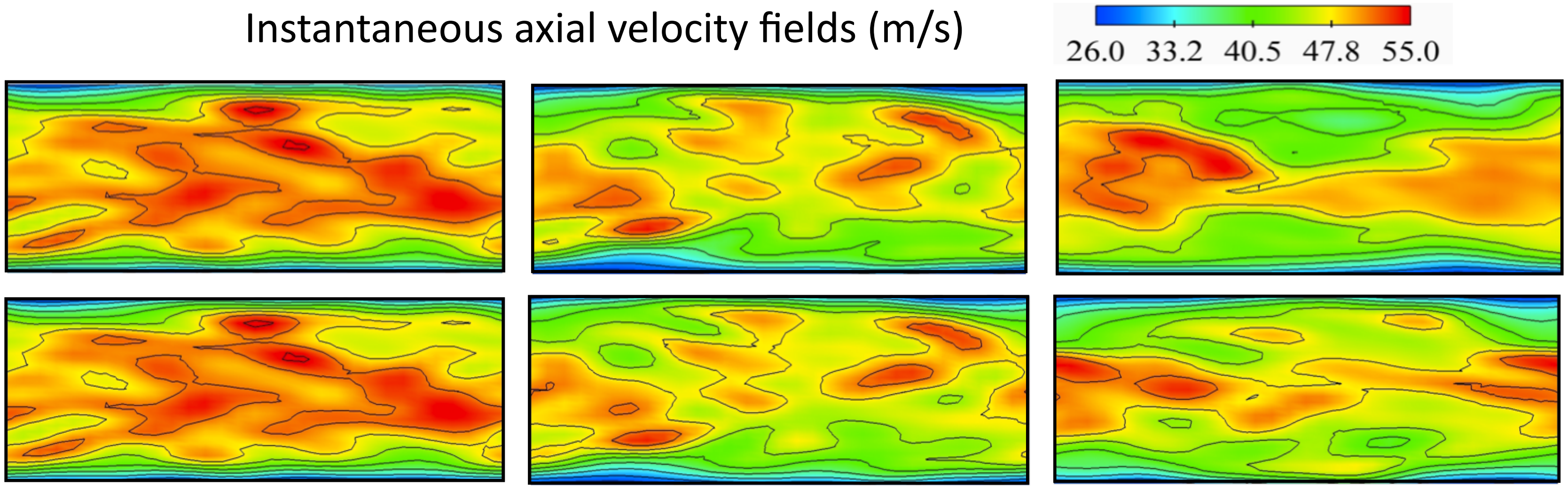
Consequence of the lack of associativity property (Floating point arithmetic):

- ➢ illustration on a temporally evolving turbulent channel (AVBP).

Instantaneous axial velocity fields (m/s)

4 PUs

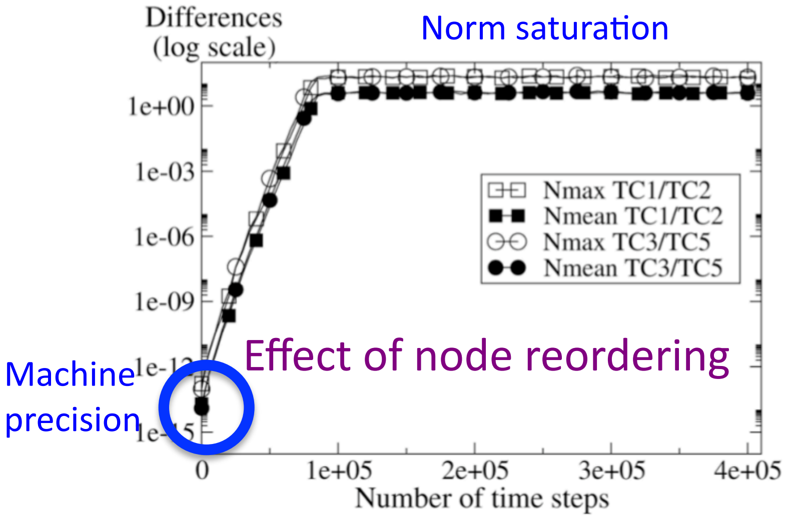
8 PUs

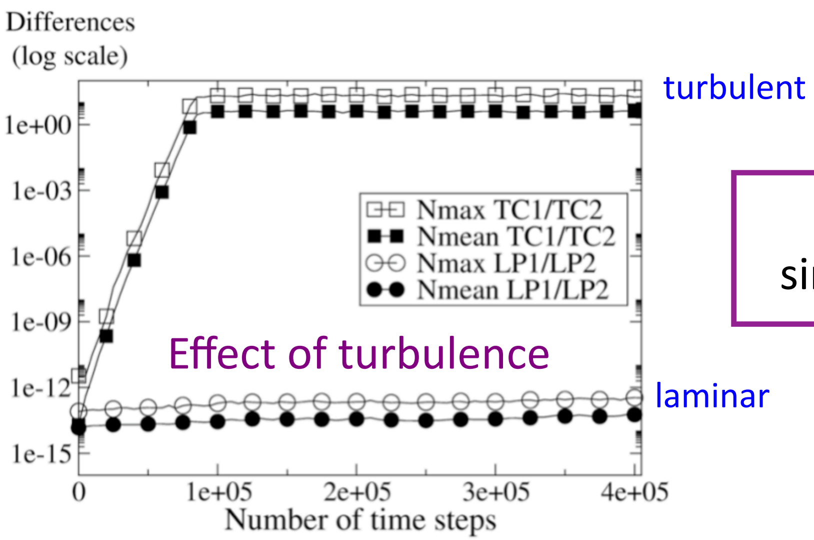


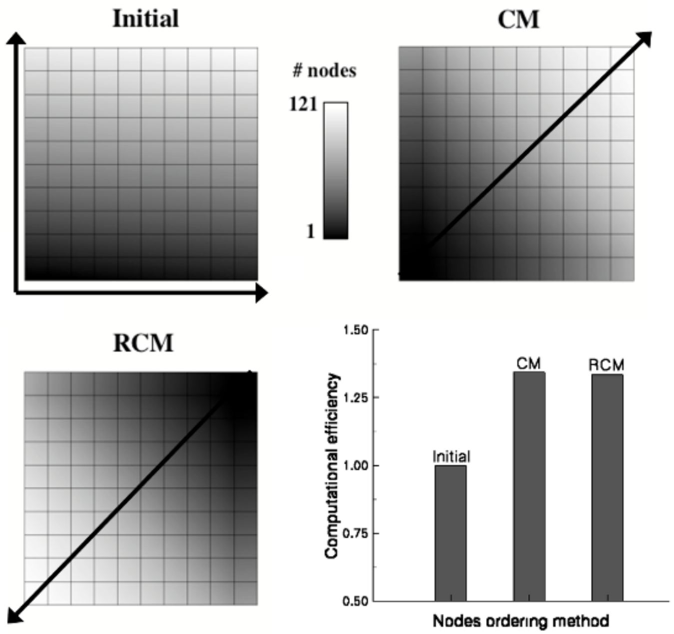
t_1 < t_2 < t_3

Senoner et al., 2008

Impact of rounding errors on LES







These results are not induced by parallel simulations (calc. performed with a single core!)

Senoner et al., 2008

Impact of mesh-partitioning

- Non-blocking communications participate to rounding errors (non-deterministic behavior),
- Blocking communications are good for deterministic behavior,
- Any sufficiently turbulent flow computed in LES exhibits significant sensitivity to small perturbations, leading to instantaneous solutions which can be totally different,
- The divergence of solutions is explained by 2 combined factors:
  - ➢ exponential separation of trajectories in turbulent flows,
  - ➢ propagation of rounding errors induced by domain partitioning and scheduling operations that can be different.
- Implicit stages done on a block basis can result in different convergence/instantaneous solutions:
  - ➢ in practice, this also impacts RANS convergence history… However since the solution is unique (?) and stationary there should be no degradation of solution observed.

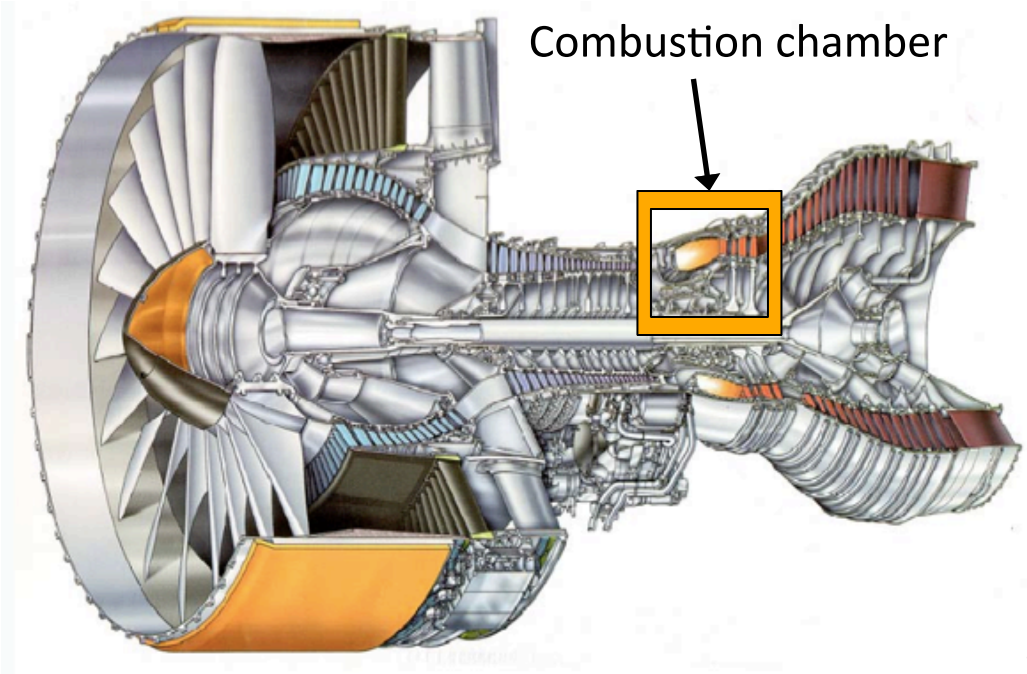
Validation of LES code after modifications may only be based on statistical fields!

Application to combustion chambers: objectives

Flow acceleration due to gas expansion/combustion:

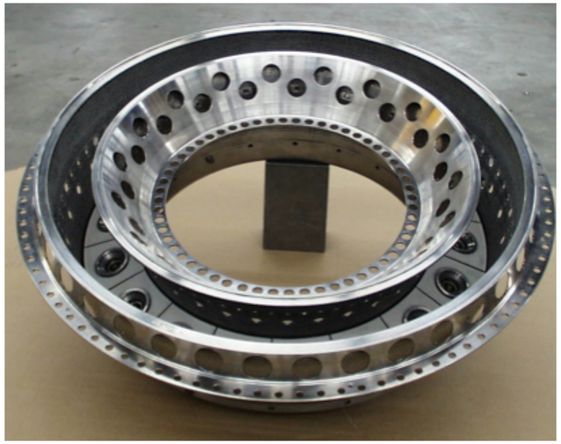
- Subject to thermo-acoustic oscillations (highly destructive and quasi unpredictable),
- Locus of pollutant formation,
- Strong thermal constraints…

=> Most recent pubications demonstrate the superiority of LES: i.e. captures the strong coupling between turbulence/mixing/combustion

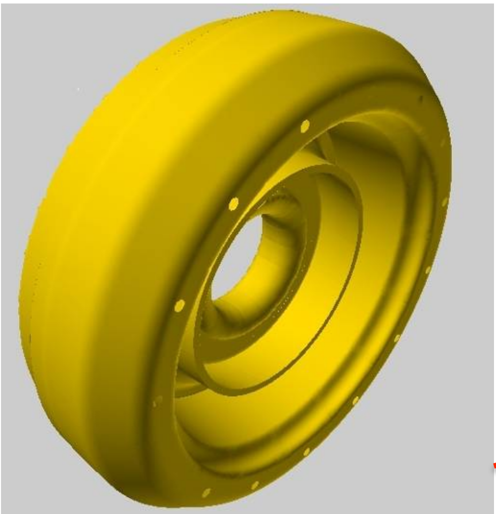


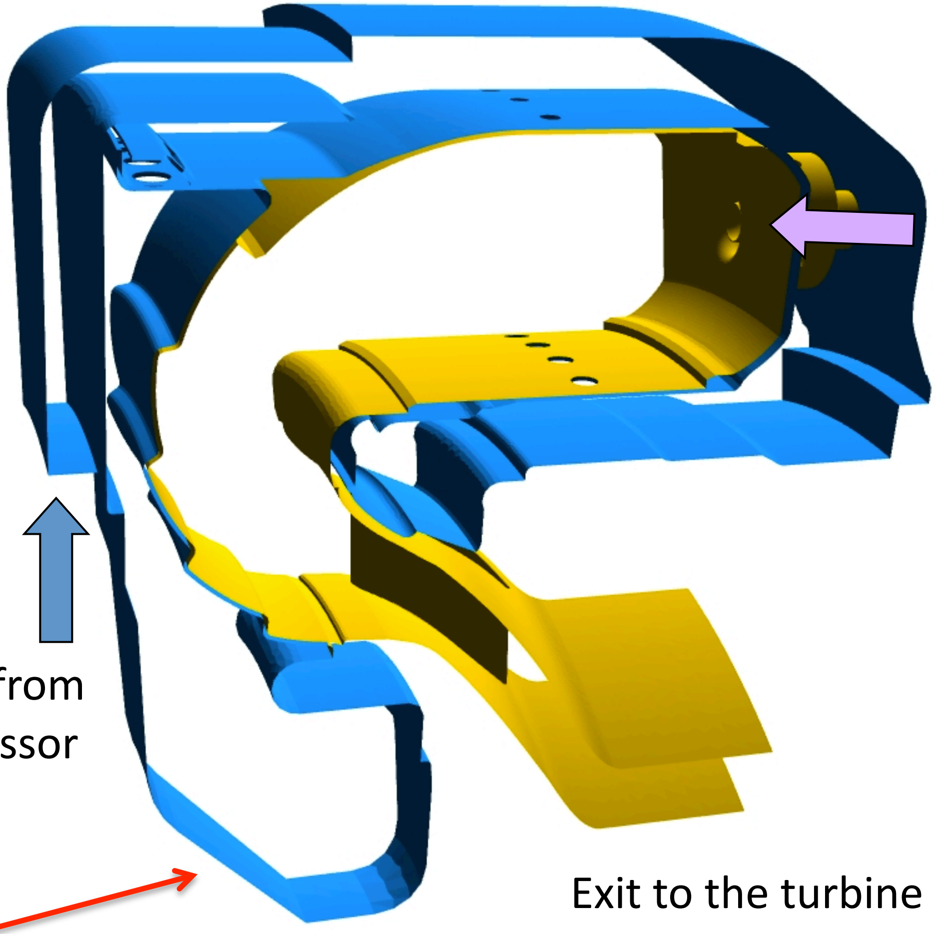
<u>Context</u>: highly complex geometry

<u>Target configuration</u>: an helicopter combustion chamber at cruise conditions.



Annular burner





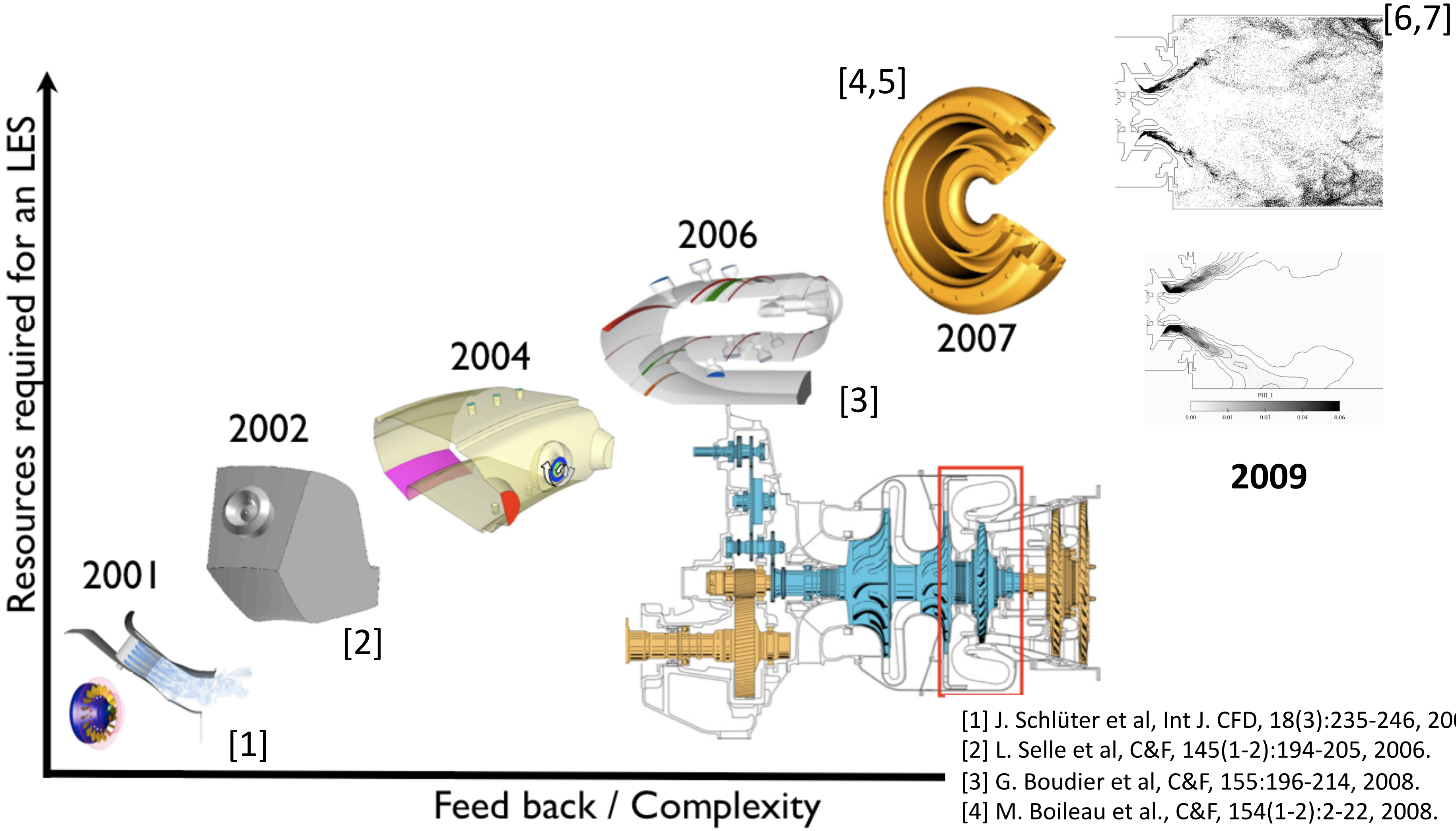
Fuel

Air coming from the compressor

Exit to the turbine

One sector

Context: highly complex geometry

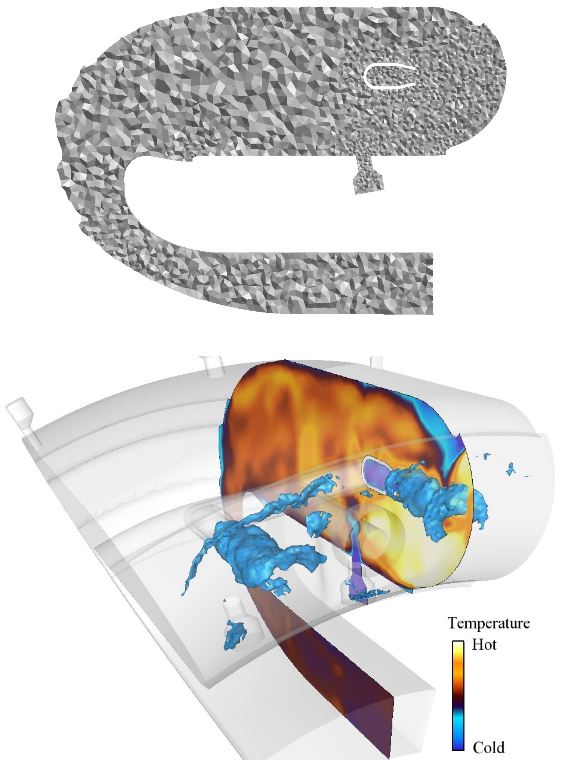


[1] J. Schlüter et al, Int J. CFD, 18(3):235-246, 2004.
[2] L. Selle et al, C&F, 145(1-2):194-205, 2006.
[3] G. Boudier et al, C&F, 155:196-214, 2008.
[4] M. Boileau et al., C&F, 154(1-2):2-22, 2008.
[5] G. Staffelbach et al., 32nd Symp., 2008.
[6] E. Riber et al, JCP, 228(2):539-564, 2009.
[7] F. Jaegle, PhD dissertation, INPT, 2009.

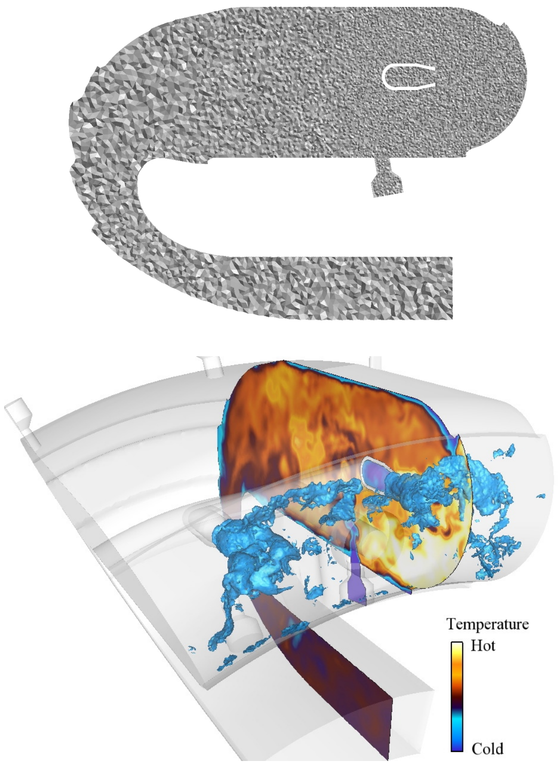
Effect of grid resolution: overview

LES in a single sector burner

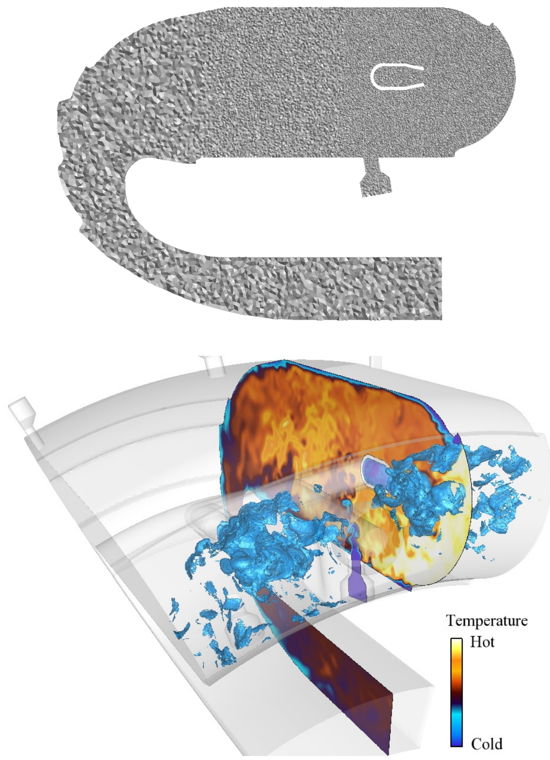
Increasing number of core
=>
Increasing mesh resolution



315 CPU hours

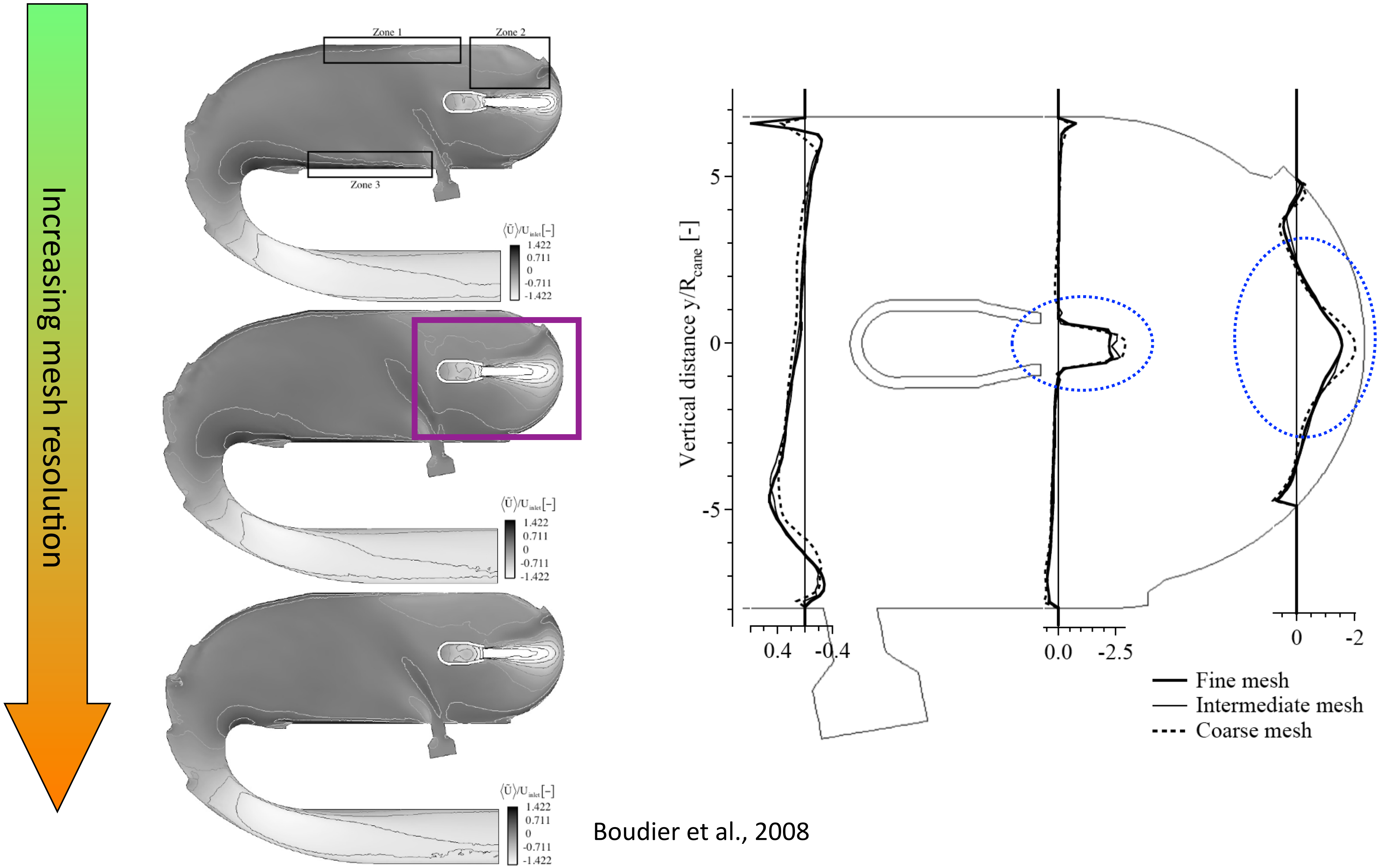


4,550 CPU hours



30,200 CPU hours

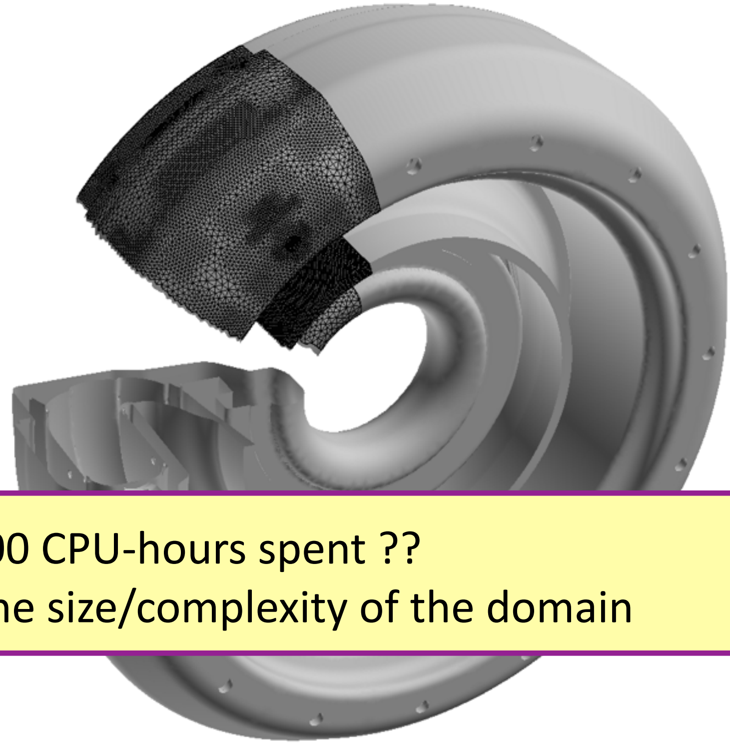
Effect of grid resolution: mean quantities



Boudier et al., 2008

Application to full annular burner: overview

- Numerical aspects:
  - 3D compressible LES (AVBP),
  - reactive Navier-Stokes solver,
  - TTGC convective scheme (3rd order),
  - Smagorinsky model[1],
  - NSCBC boundary conditions[2],
  - Initial conditions from statistically converged mono-sector results.



What do you get out of the 1,000,000 CPU-hours spent ??
Increasing the number of cores => Increasing the size/complexity of the domain

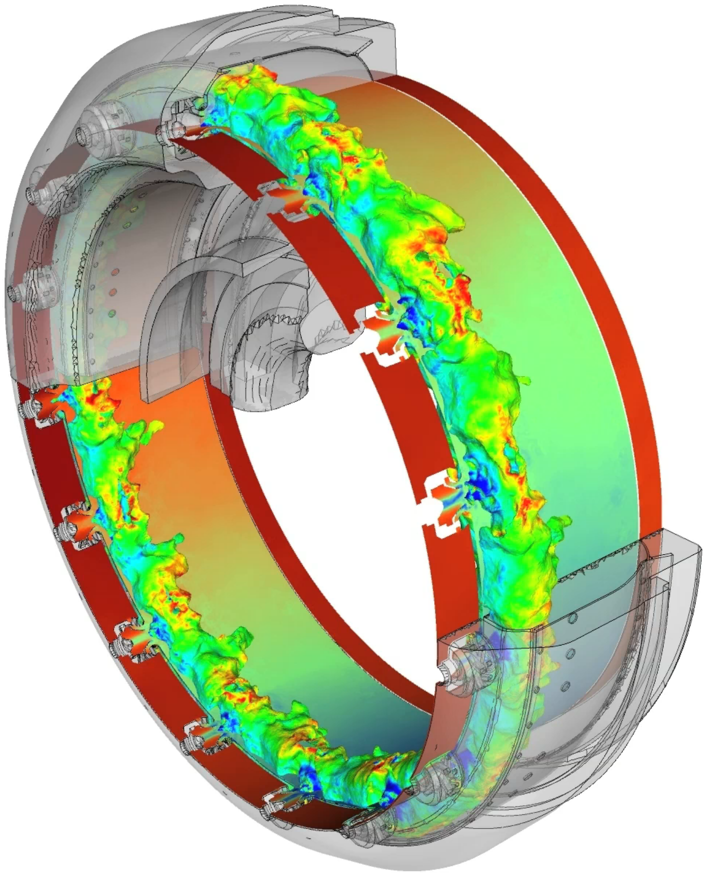
- Chemical aspects:
  - JP10 1-step fitted mechanism (surrogate for kerosen[3])
  - Dynamic Flame Thickening[4].

G. Staffelbach et al., 2010
G. Boudier et al., IJ Aeroacoustic, 2007

(1) Smagorinsky et al., 1963
(2) Poinsot et al., 1992
(3) Légier et al., 2001
(4) Colin et al., 2000

Application to full annular burner: impact on pressure and temperature

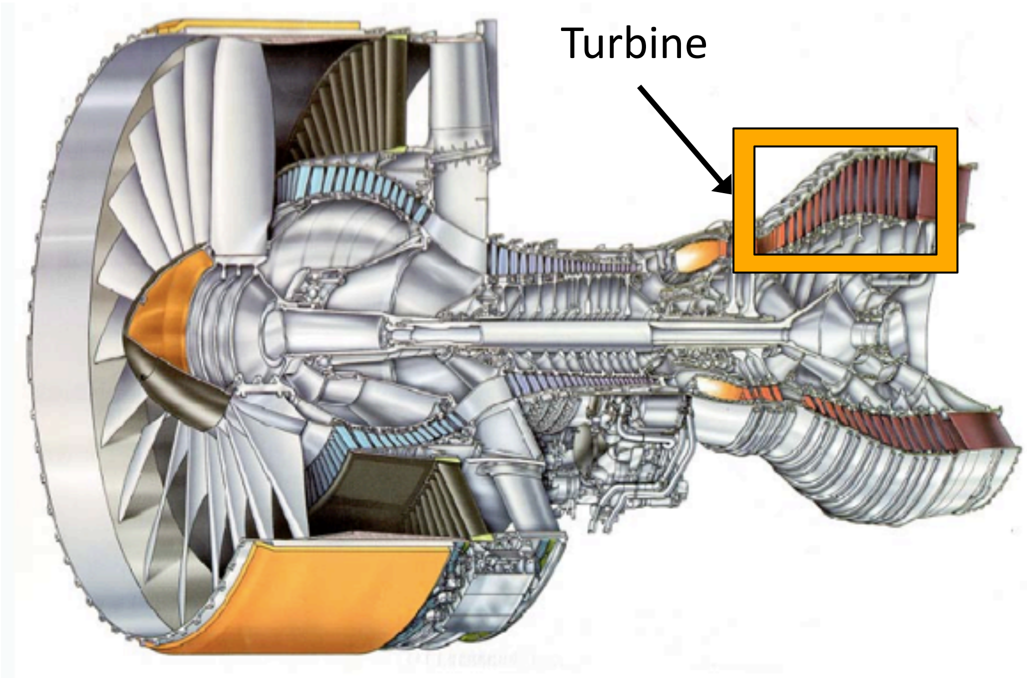
P. Wolf (CERFACS)



- Temporal evolution of pressure typical of the expression of two counter-rotating pressure waves: self-sustained azimuthal thermo-acoustic instability.
- Unexpected implication of the instability: azimuthal oscillation of combustion and the temperature field.

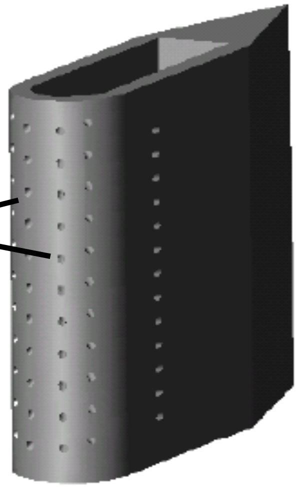
Application to turbines: Objectives

- Unsteady flows are still not well understood,
- Main reasons are also computational cost, size and complexity of the configurations,
- Challenges today for turbine designers is the prediction of heat transfer:
  - a 15 K difference on the temperature prediction leads to a reduction of its life duration by a factor 2,
  - (U)RANS methods are not adapted to complex flows.



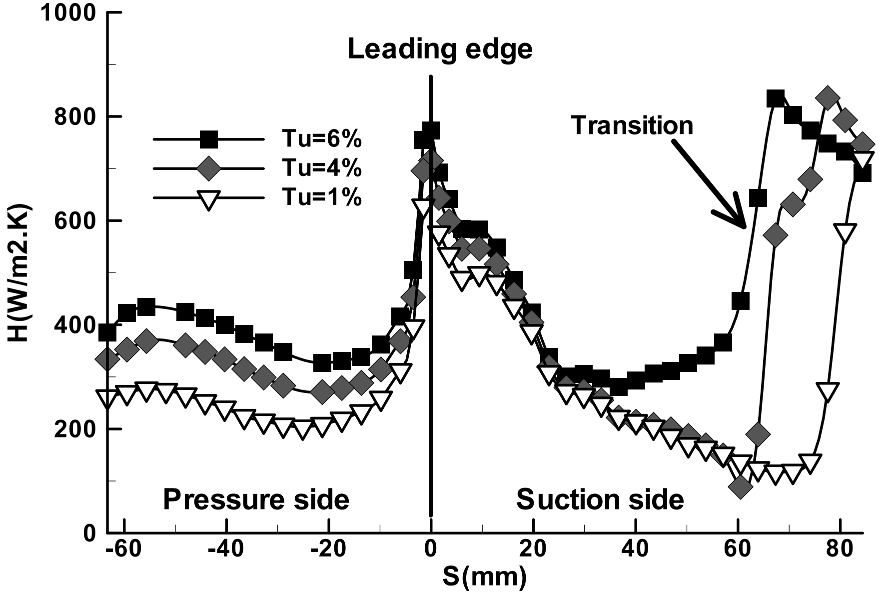
Aerothermal in turbine blades: overview

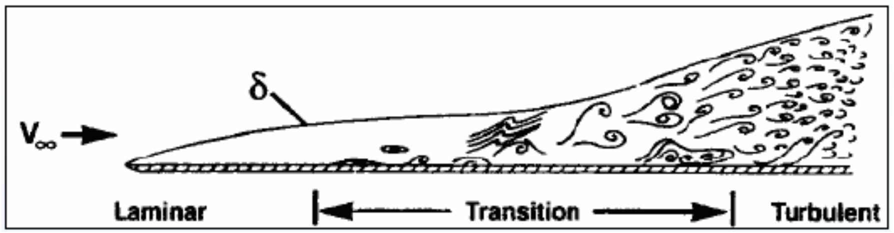
film cooling
holes



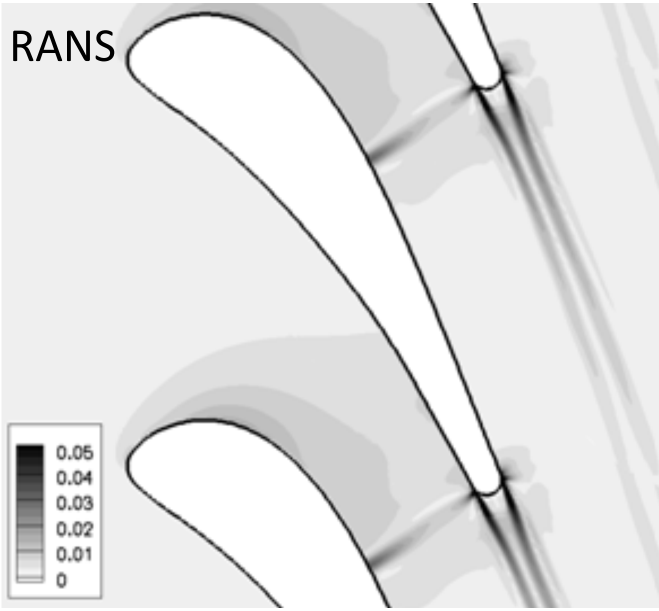
Complex flows that can not be efficiently computed with a (U)RANS method:

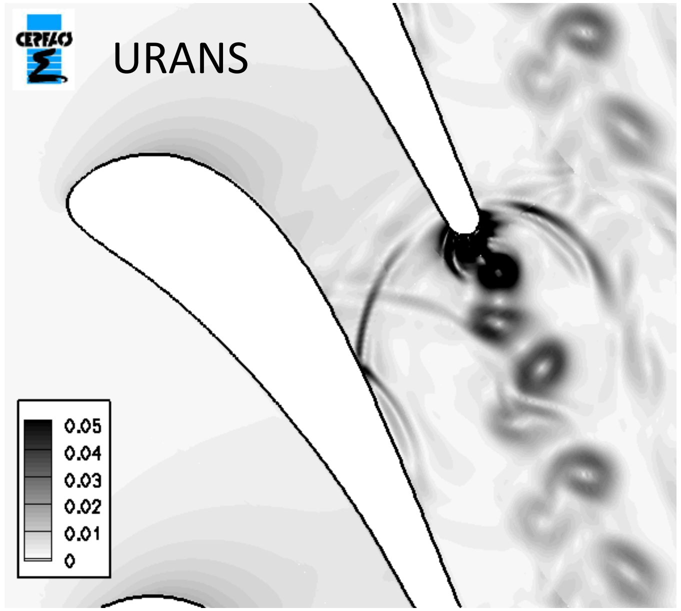
- laminar to turbulent transition,
- hot spot incoming from the combustion chamber,
- aero-thermal interactions (adiabatic is not true).



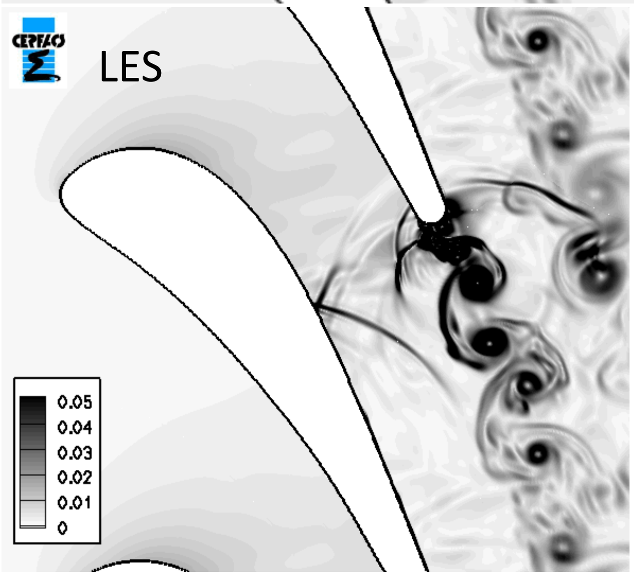


Impact on unsteady aerodynamic performance





T. Léonard et al., in ASME Turbo-Expo, Glasgow, 2010.

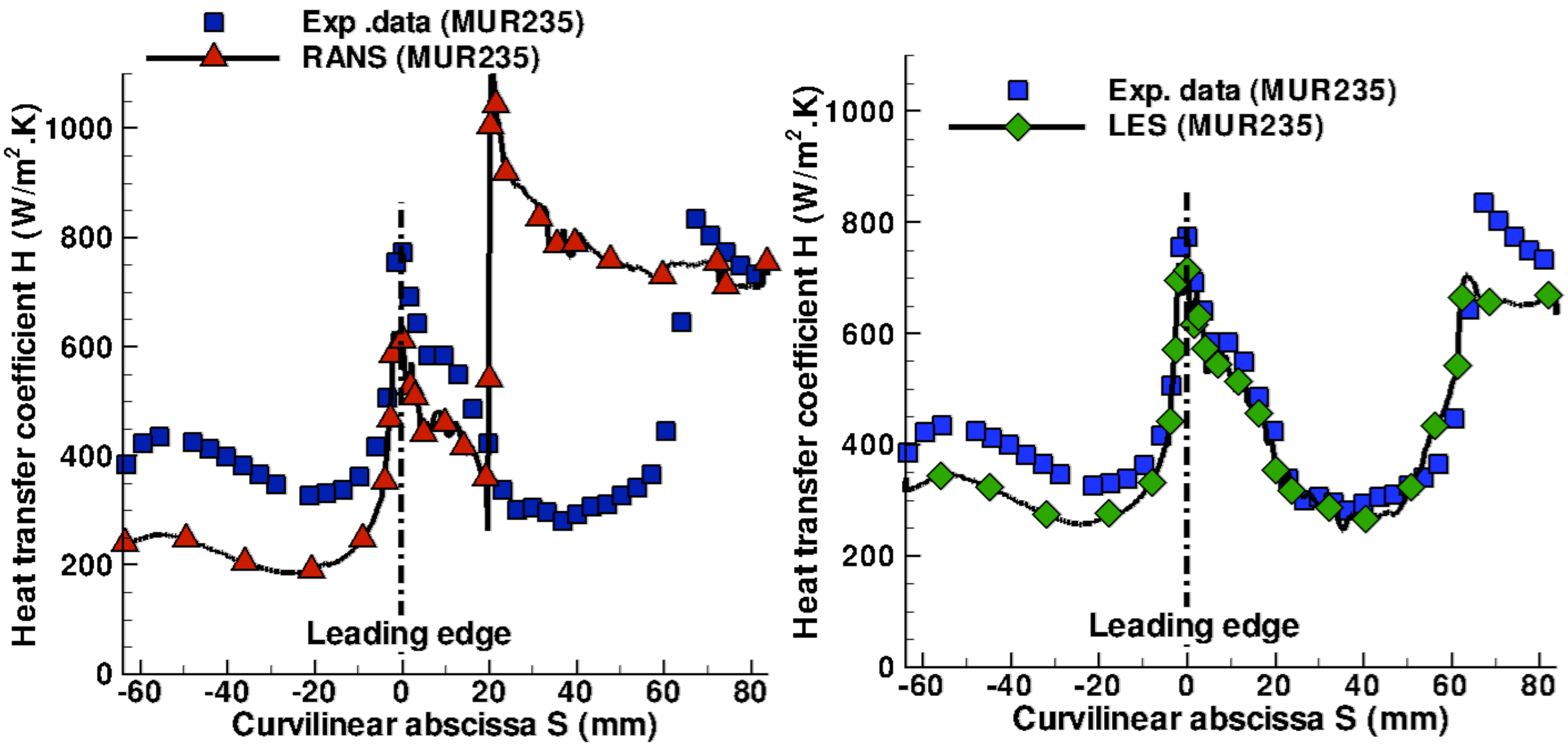


Instantaneous gradp flow field

- RANS predicts a non-physical shock-wave,
- URANS predicts the vortex shedding but flow features are damped by artificial viscosity,
- LES demonstrates its capacity to transport flow vortices and acoustic waves.

Impact on aerothermal predictions

E. Collado (CERFACS)



Unsteadiness and turbulence modelisations are crucial !

➔ LES is a good candidate, HPC required (29 M cells)

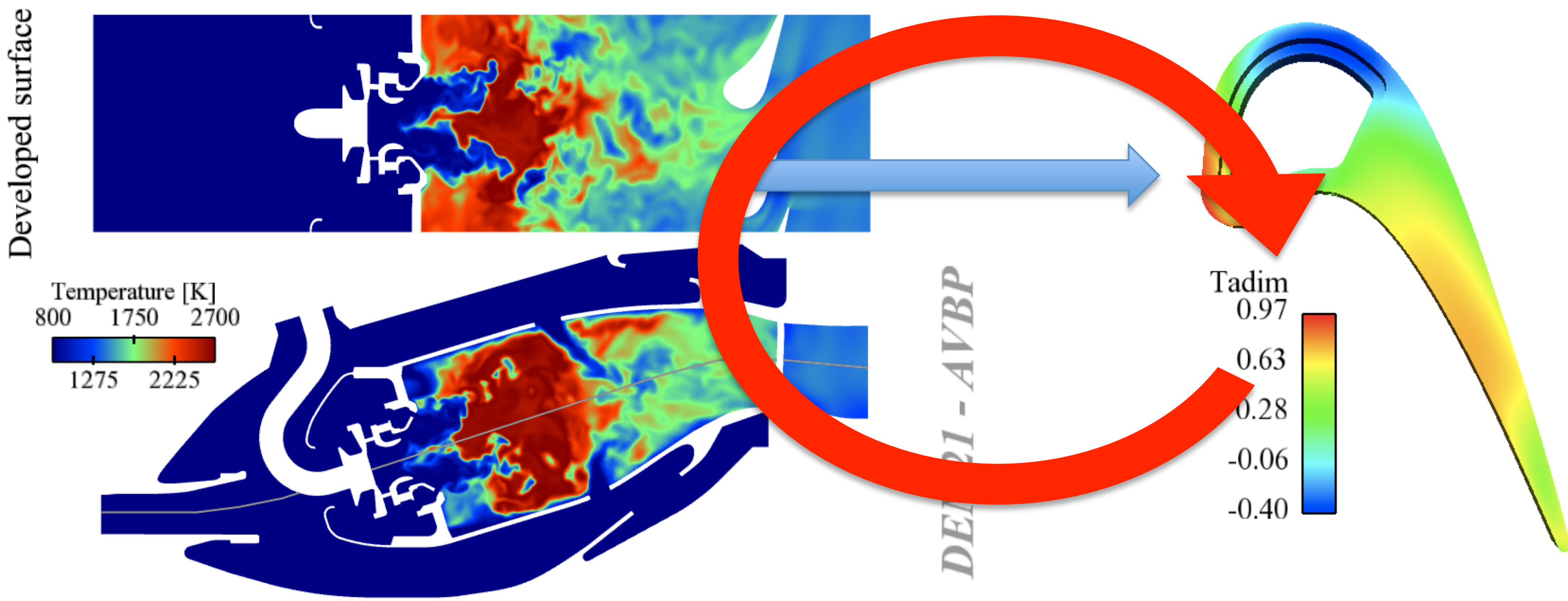
Complex geometries: unsteadiness, mixing, …

F. Duchaine (CERFACS)

Why multi-physics simulations?

• Pure CFD cannot give all relevant information for comprehension and design: heat transfer, vibration, solid deformation …

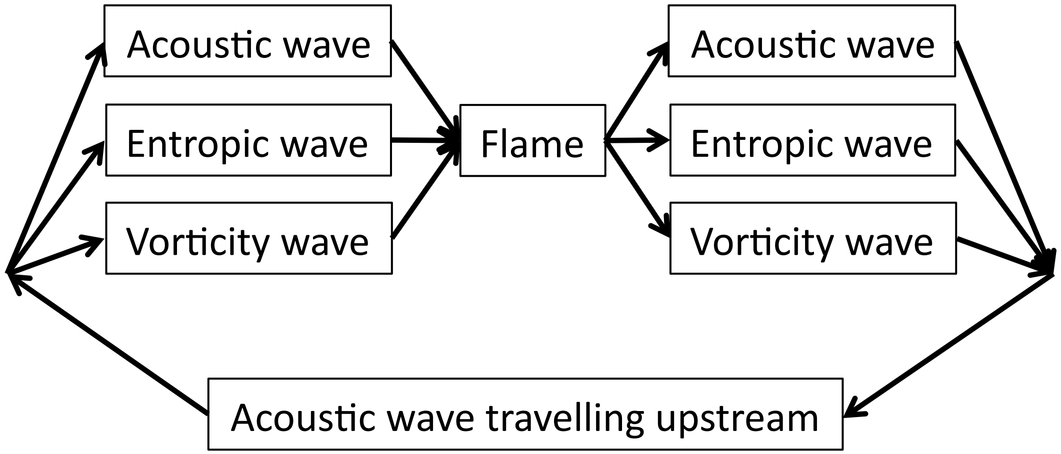
• The challenge: combine all the important information of different physics in one simulation

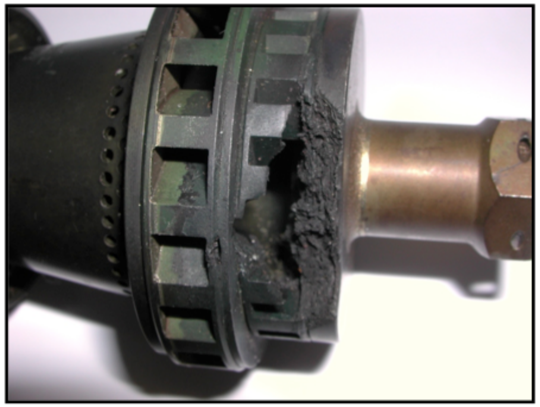


Why multi-physics simulations?

- Enhance predictions: +15 K on a turbine blade = life time / 2

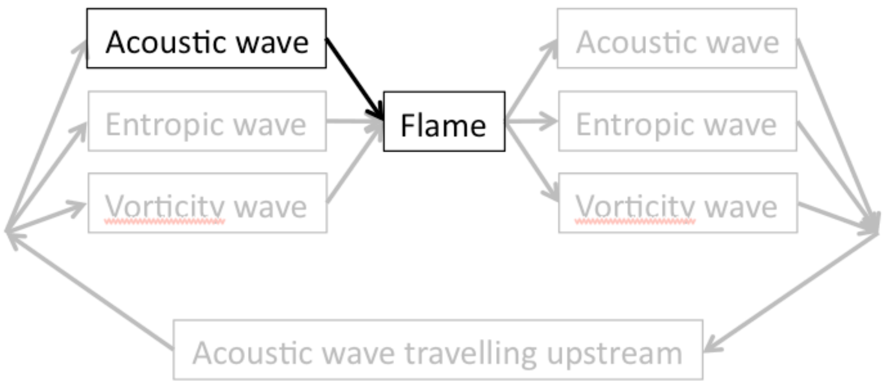
- Better comprehension and reproduction of the physics, example: thermo-acoustic instabilities

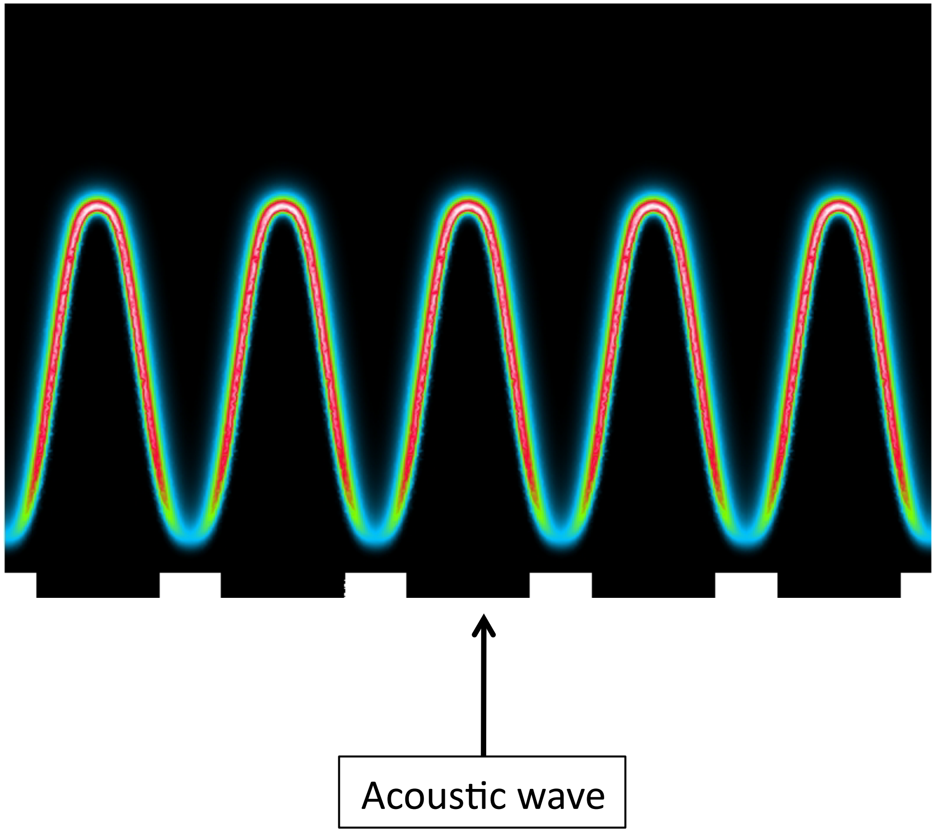




Why multi-physics simulations?

- Better comprehension and reproduction of the physics, example: thermo-acoustic instabilities

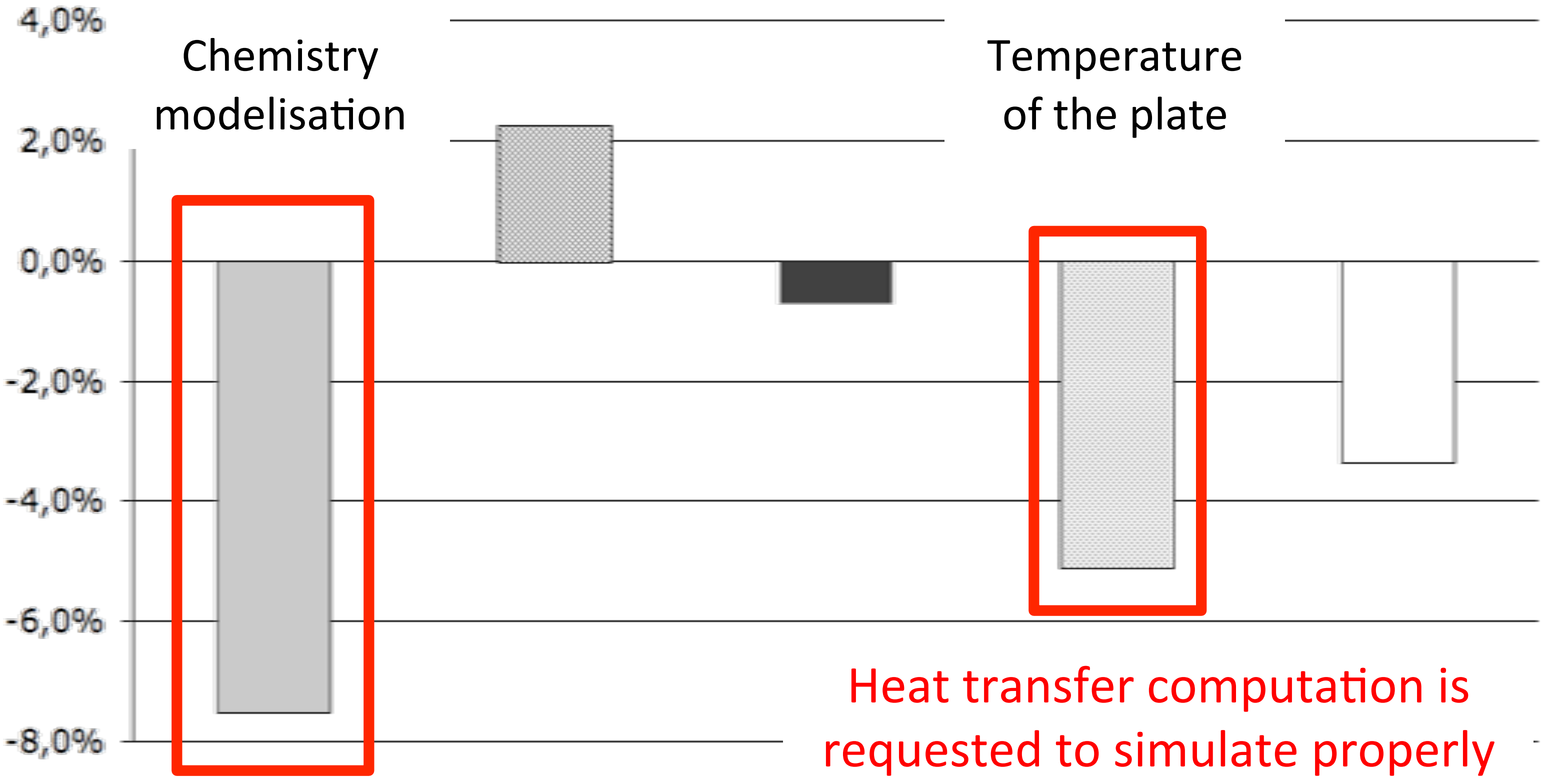


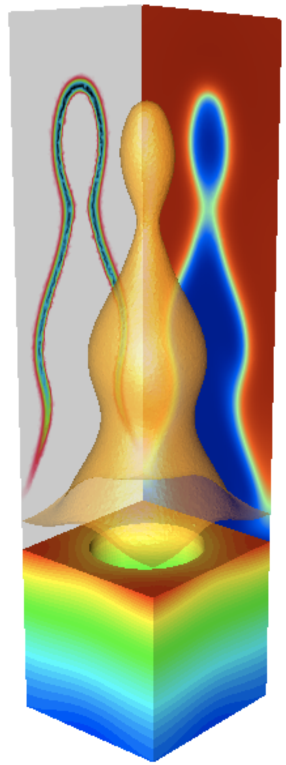


F. Duchaine (IMFT)

Why multi-physics simulations?

- Sensitivity analysis of flame response to acoustic perturbations based on CFD



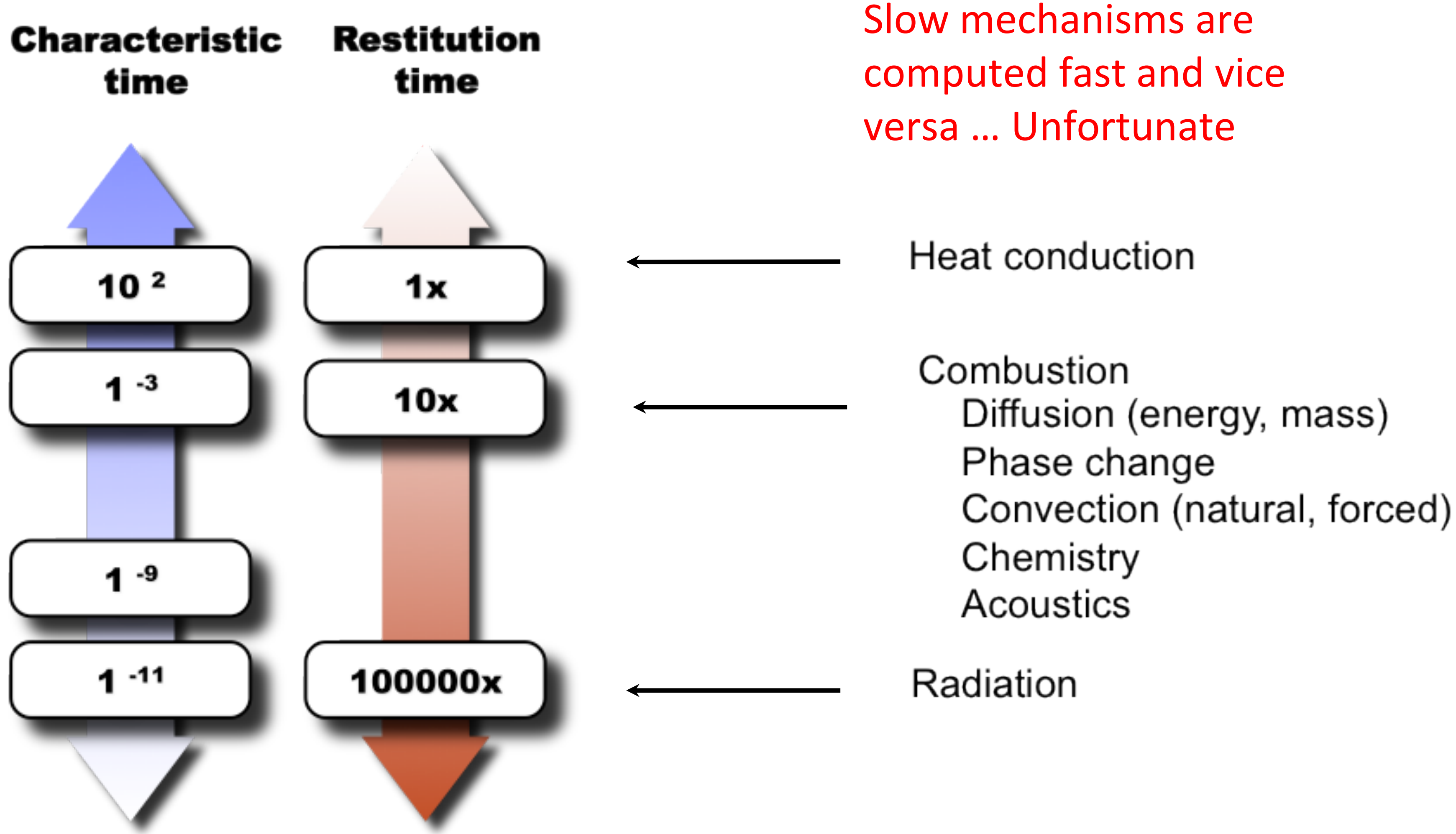


Heat transfer computation is requested to simulate properly the physic of the system

F. Duchaine, F. Boudy, D. Durox and T. Poinsot. Sensitivity analysis of transfer functions of laminar flames. Combustion and Flame. In press

- In a combustion chamber: heat transfer by convection – conduction and radiation

Slow mechanisms are computed fast and vice versa … Unfortunate



J. Amaya (CERFACS)

- Types of problems

  ⇒ (Quasi) Steady state problem: cruise regime …

  ⇒ Unsteady fluid problem: ignition phase, extinction …

  ⇒ Unsteady solid problem: change in the cruise regime

- To ensure a reasonable restitution time, all these problems require a specific treatment because: it is still too expensive to compute a transient evolution in a fluid on a time characteristic of a solid

- How to do multiphysics:

⇒ write a single monolithic code

☺ Very efficient if well done for HPC,
☹ Demanding for the development,
☹ Not always possible,
☹ Difficulties for evolutions,
☹ Does not preserve the effort made in code development

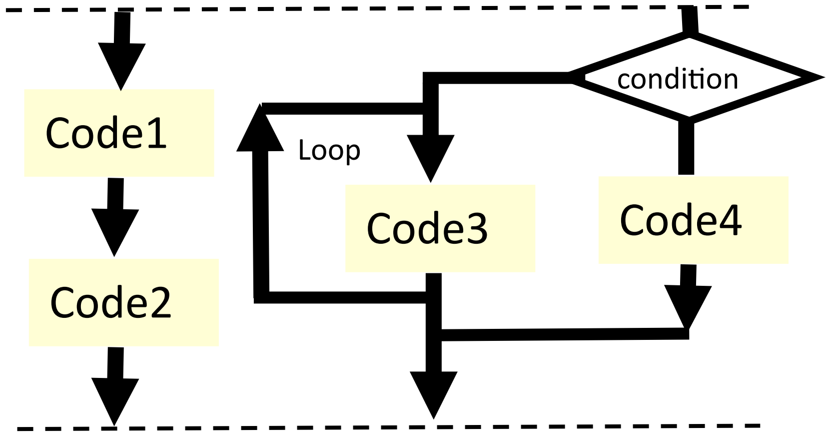
⇒ Use your favorite communication protocol to make different exchange data

☺ Preserve existing codes,
☺ Very efficient if well done,
☹ Demanding for the development (more than 2 codes …),
☹ Difficulties for evolutions,
☹ Some numerical difficulties to solve

⇒ Use a coupler

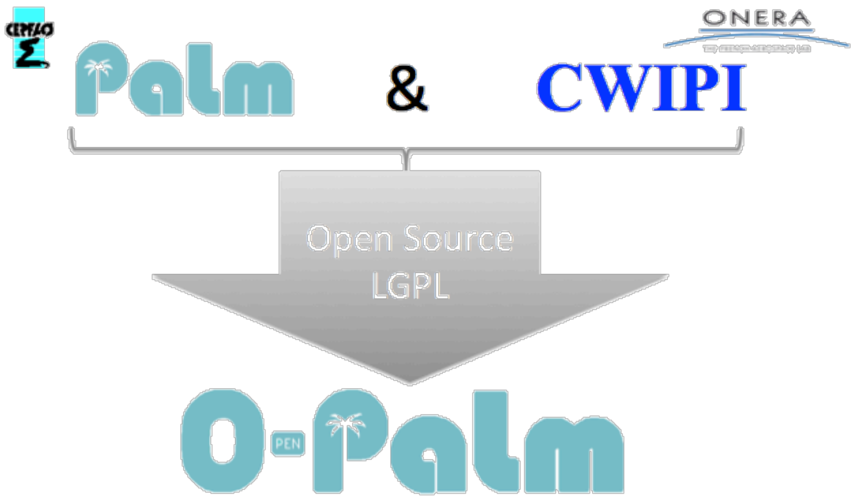
☺ Preserve existing codes,
☺ Very efficient if well done,
☹ Some numerical difficulties to solve

- A coupler is a library of functionalities that facilitate the exchange of data between existing codes,
- PALM (C/fortran/MPI2 and MPI1) is a dynamic coupler of parallel codes developed since 1996 at CERFACS



➔ Recent developments for multi platforms simulations with IP transfer

- Open-PALM since 2011



- A dynamic code coupler
  - ✓ Designed to do very complex algorithms,
  - ✓ With a high level of flexibility,
  - ✓ Make communications between parallel codes running at the same time or not

⇒ Code coupling,

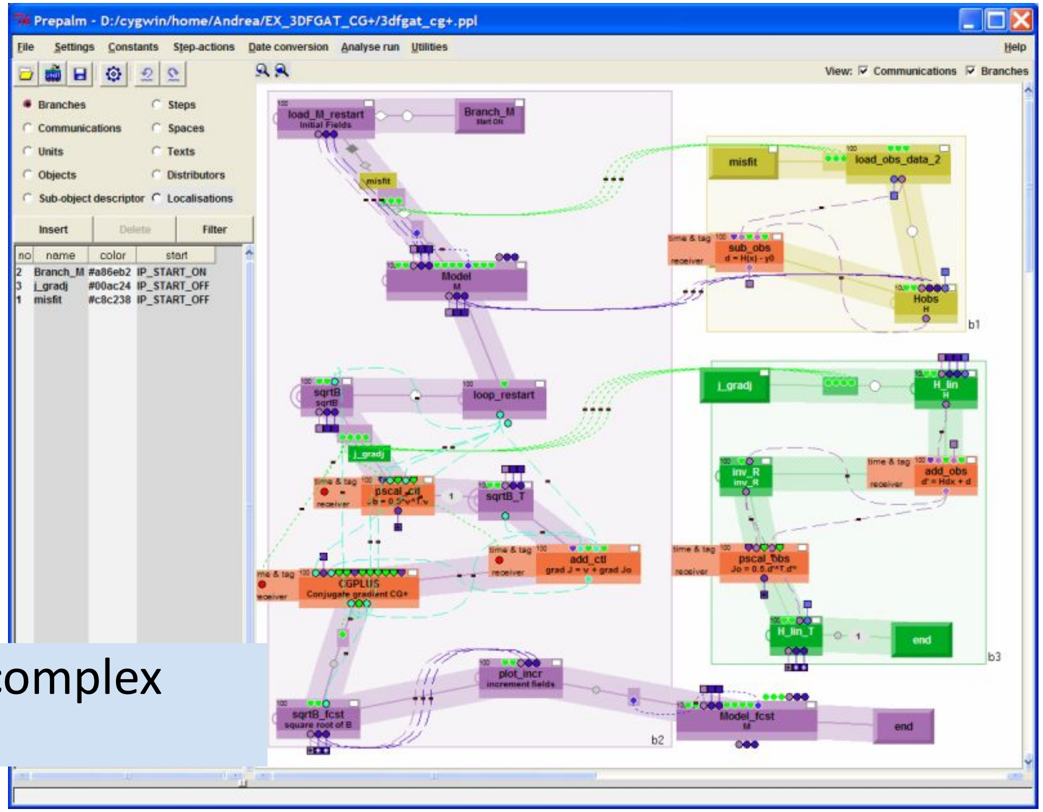
⇒ Data assimilation,

⇒ Optimization environment,

⇒ Post-processing,

⇒ UQ,

⇒ …



PALM facilitates the prototyping of complex applications

To perform multiphysic simulations in gas turbine environment with a coupler, it is necessary to address:

- Physical and numerical issues
  - ⇒ boundary conditions,
  - ⇒ exchange frequencies (due to time marching codes),
  - ⇒ consistency,
  - ⇒ stability,
  - ⇒ …
- Computational issues for HPC
  - ⇒ data exchange,
  - ⇒ performance,
  - ⇒ scalability

1 Numerical developments in CFD for HPC

- CFD
- Flow solver examples
- Speed-up and Mesh-partitioning
- Communication, Impact on numerical solutions
- Applications to aeronautic challenges

2 Code coupling

- Why multi-physics simulations
- **Physical and numerical issues**
- HPC issues
- Applications to aeronautic challenges

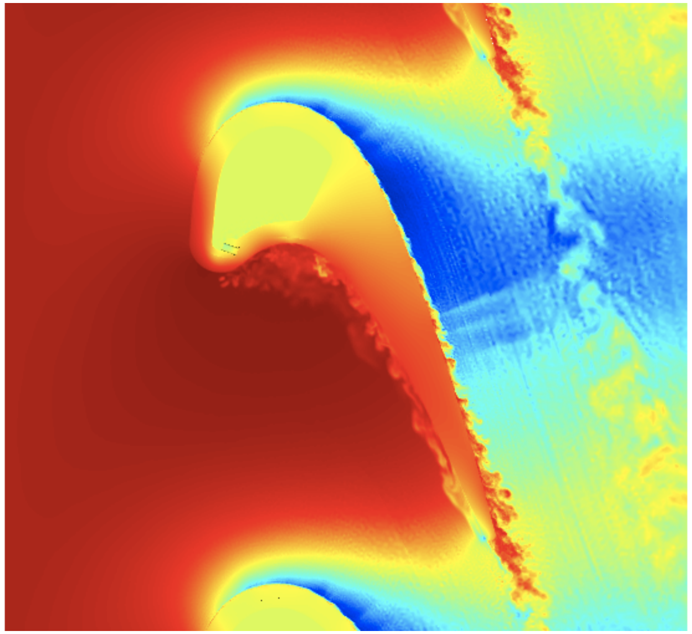
3 Conclusion and perspectives

Example of a coupling methodology to reach a steady state in a conduction / convection problem based on a LES solver and a transient conduction code

⇒ Strategy to accelerate the thermal convergence of the coupled problem,

⇒ Definition of the variable to exchange,

⇒ Combinations of these two points



F. Duchaine, A. Corpron , V. Moureau , F. Nicoud , T. Poinsot. Development and assessment of a coupled strategy for conjugate heat transfer with Large Eddy Simulation. Application to a cooled turbine blade. International Journal of Heat and Fluid Flow, 30(6) : 1129-1141, 2009

- Analysis of characteristic times :

  Convection time in the fluid: $\tau_F = L_F / v$

  Diffusion in a solid: $\tau_S = L_S^2 / D$

Generally, $\tau_f \ll \tau_s$: the thermal convergence is driven by the solid

- Lets imagine that between each synchronization of the solvers,

  the flow is advanced in time of a quantity $\alpha_F \tau_F$

  and the solid is advanced of a time $\alpha_S \tau_S$

- Synchronization in physical time imposes: $\alpha_F \tau_F = \alpha_S \tau_S$

The convergence speed is drastically limited by the cost of the CFD computation

- To accelerate the convergence, we propose to impose a synchronization in the characteristic times: $\alpha_F = \alpha_S = \alpha$

- At the fluid / solid interface, $\left\{ \begin{array}{l} \text{the temperature } T \\ \text{the heat flux } \Phi \end{array} \right.$ are continuous

- For a fully coupled unsteady problem (with $\alpha_F \tau_F = \alpha_S \tau_S$), the stability is obtained with

$$\left\{ \begin{array}{l} \text{impose the solid temperature } T_S \text{ to the fluid solver} \\ \text{impose the fluid heat flux } \Phi_F \text{ to the solid} \end{array} \right.$$

Giles 1997

- For steady state problem, it is common to use

$$\left\{ \begin{array}{l} \text{impose the solid temperature } T_S \text{ to the fluid solver} \\ \text{impose the fluid heat flux } \Phi_F \text{ to the solid with } \Phi_S = h_c (T_c - T_s) \end{array} \right.$$

It may be difficult to find adequate values of hc and Tc

- Thus, we propose to adopt a mixed formulation:

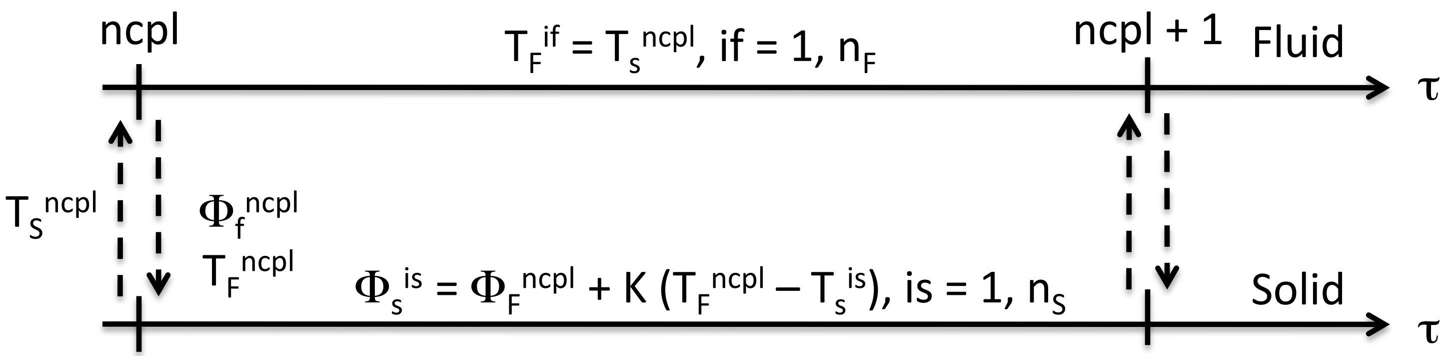
$$\left\{ \begin{array}{l} T_F = T_S \\ \Phi_F + K\, T_F = \Phi_S + K\, T_s \end{array} \right. \quad \Rightarrow \quad \left\{ \begin{array}{l} T_F = T_S \\ \Phi_S = \Phi_F + K\,(T_F - T_s) \end{array} \right.$$

F. Wlassow, F. Duchaine, G. Leroy and N. Gourdain. 3D Simulation of Coupled Fluid Flow and Solid Heat Conduction for the Calculation of Blade Wall Temperature in a Turbine Stage. GT2010-22513. In ASME Turbo Expo 2010, Glasgow, Scotland UK

Two sum up, the method relies on two parameters α and K

	Stability	*Efficiency*
1) $\alpha = \alpha_F = \alpha_S$	?	?
2) $\begin{cases} T_f = T_s \\ \Phi_s = \Phi_f + K\,(T_f - T_s) \end{cases}$	?	?

Lets quantify the effect of these parameters



$n_F = \alpha\tau_F / \Delta t_F$

$n_S = \alpha\tau_S / \Delta t_S$

Stability: 1D pure diffusion problem

=> $T_2 = T_1$

$\Phi_1 = \Phi_2 + k\,(T_2 - T_1)$ <=

Heat equation:

$$\rho_i C_i \frac{\partial T_i(x)}{\partial t} = \lambda_i \frac{\partial^2 T_i(x)}{\partial x^2}, \quad i = 1,2$$

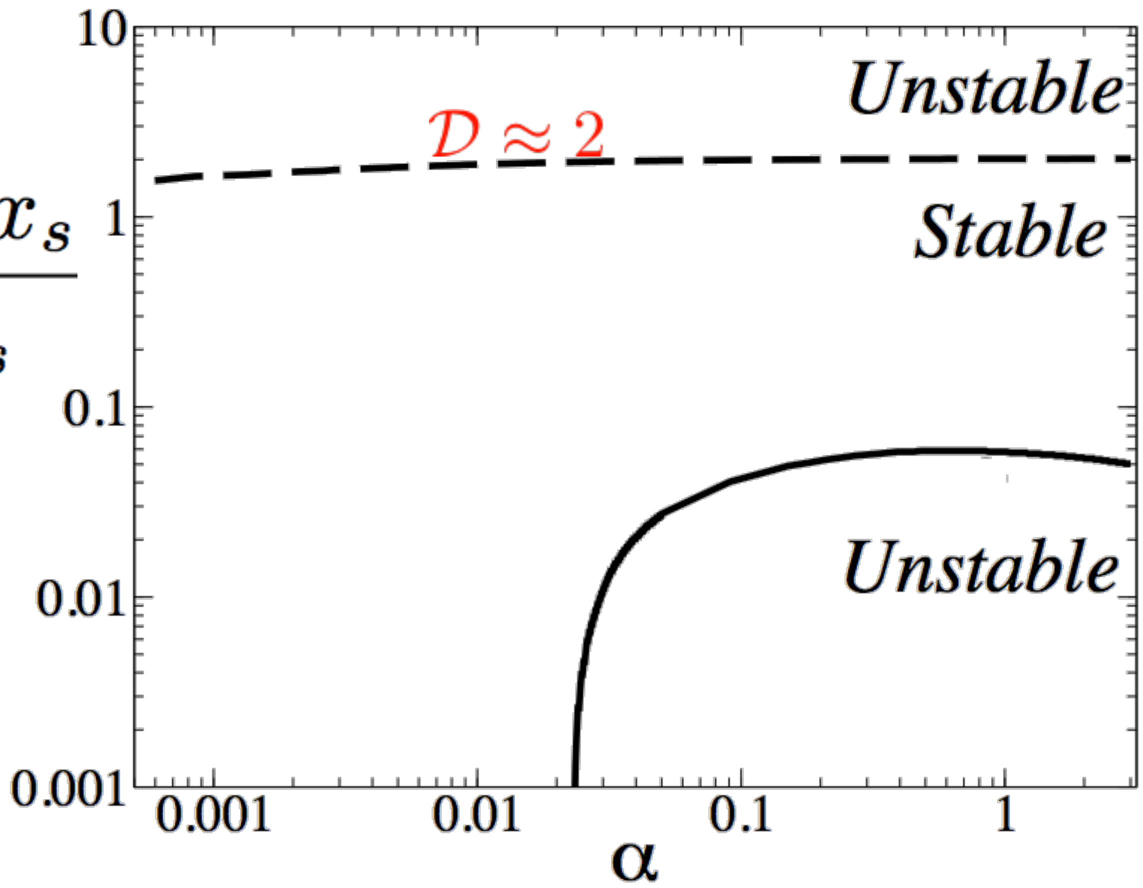
Discretization:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \mathscr{F}_i (T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}), \quad i = 1,2$$

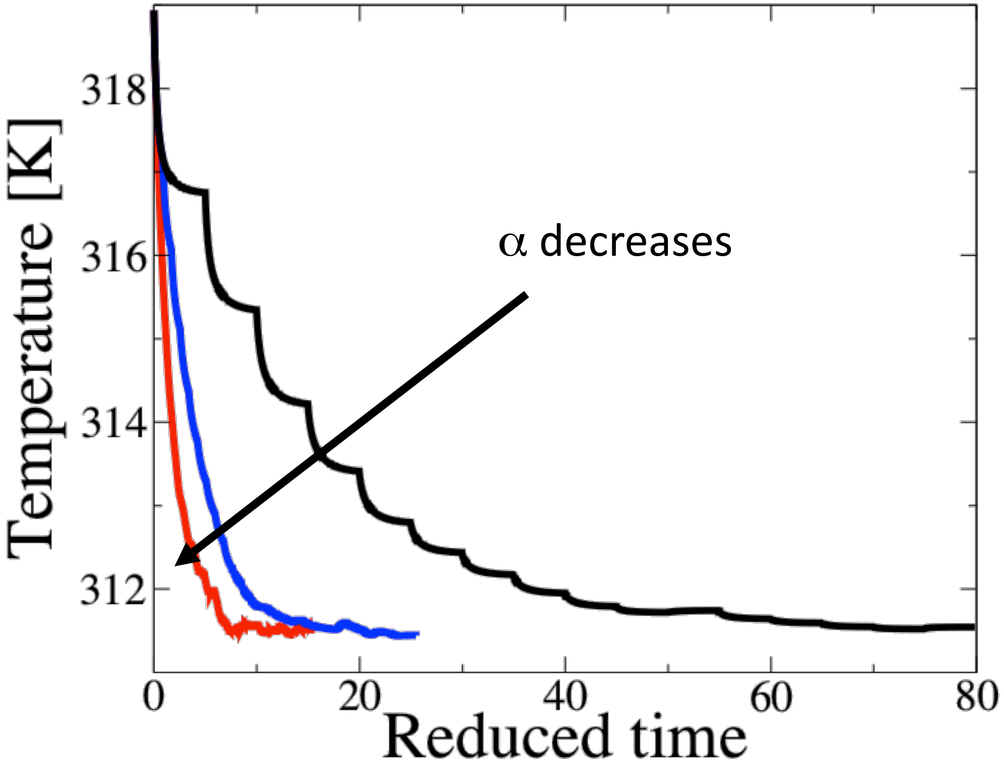
$$\mathscr{F}_i = \frac{\lambda_i \Delta t_i}{\rho_i C_i \Delta x_i^2}$$

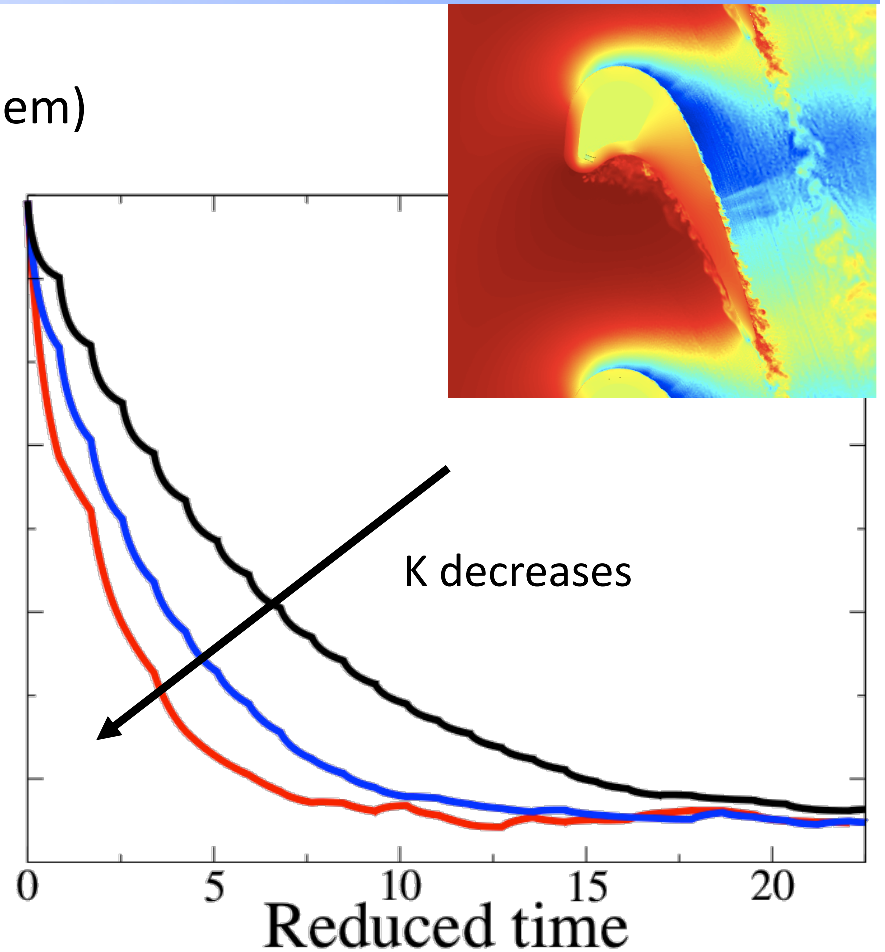
Stability: 1D pure diffusion problem

$$\mathcal{D} = \frac{k \Delta x_s}{\lambda_s}$$



Influence of α and K on efficiency (real problem)





α = 5 ; k = 100	α = 0.85 ; k = 200
α = 0.85 ; k = 100	α = 0.85 ; k = 100
α = 0.1 ; k = 100	α = 0.85 ; k = 50

Two sum up, the method relies on two parameters α and K

	Stability	*Efficiency*
1) $\alpha = \alpha_F = \alpha_S$	small	small
2) $\begin{cases} T_f = T_s \\ \Phi_s = \Phi_f + k\,(T_f - T_s) \end{cases}$	In a given range depending on α	small

From this study, it seams that small values of α and K are good candidates for stability and efficiency.

Nevertheless, in this study efficiency is gauged with convergence speed and not with restitution times !

Small values of α imply a high frequency of data exchange not compatible with efficiency on HPC

=> Trade off between communications / computation => α => K> 0

1 Numerical developments in CFD for HPC
- CFD
- Flow solver examples
- Speed-up and Mesh-partitioning
- Communication, Impact on numerical solutions
- Applications to aeronautic challenges

2 Code coupling
- Why multi-physics simulations
- Physical and numerical issues
- **HPC issues**
- Applications to aeronautic challenges

3 Conclusion and perspectives

HPC issues for code coupling with mesh partitioning

- data transfer between massively parallel codes
- the data can be light compared to the model (surface exchanges) or heavy (volume exchanges)
- interpolations from one model to the other(s)

Constraints

⇒ efficiency (execution time and memory consumption) of the processing strategy

⇒ scalability when the model and/or the number of core increase

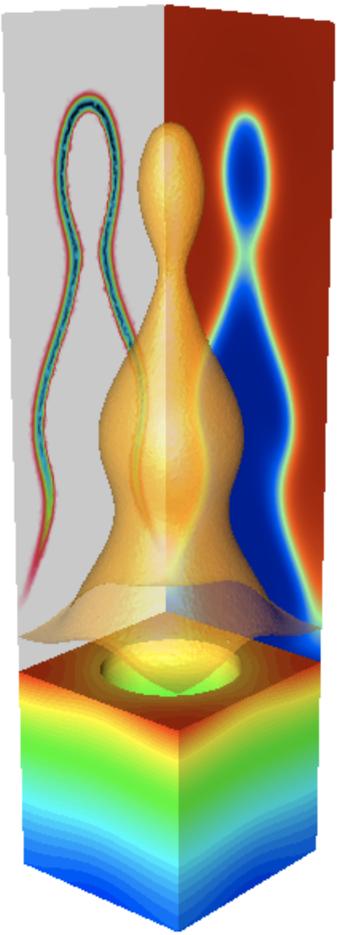
⇒ precision of the interpolation results

⇒ easy to use for different types of discretizations

Two main challenges:

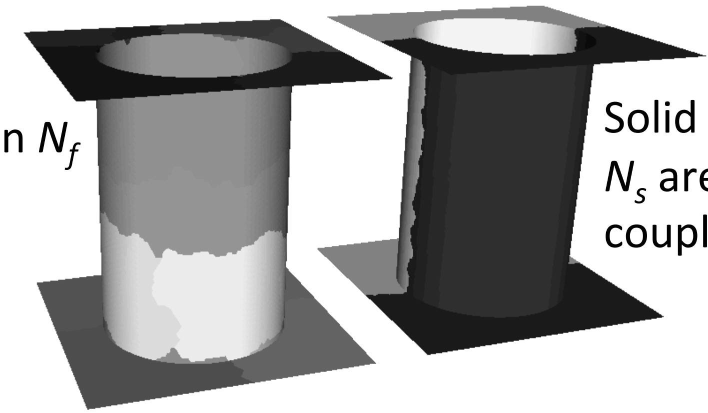
- information routing between parallel codes
- interpolation in parallel

A coupled simulation step by step



- Launch the parallel codes
- Domain decomposition in the codes
- (*) Sharing of the parts to couple: communication graph between the codes

Fluid solver: n_f cores on N_f are concerned by the coupled interface

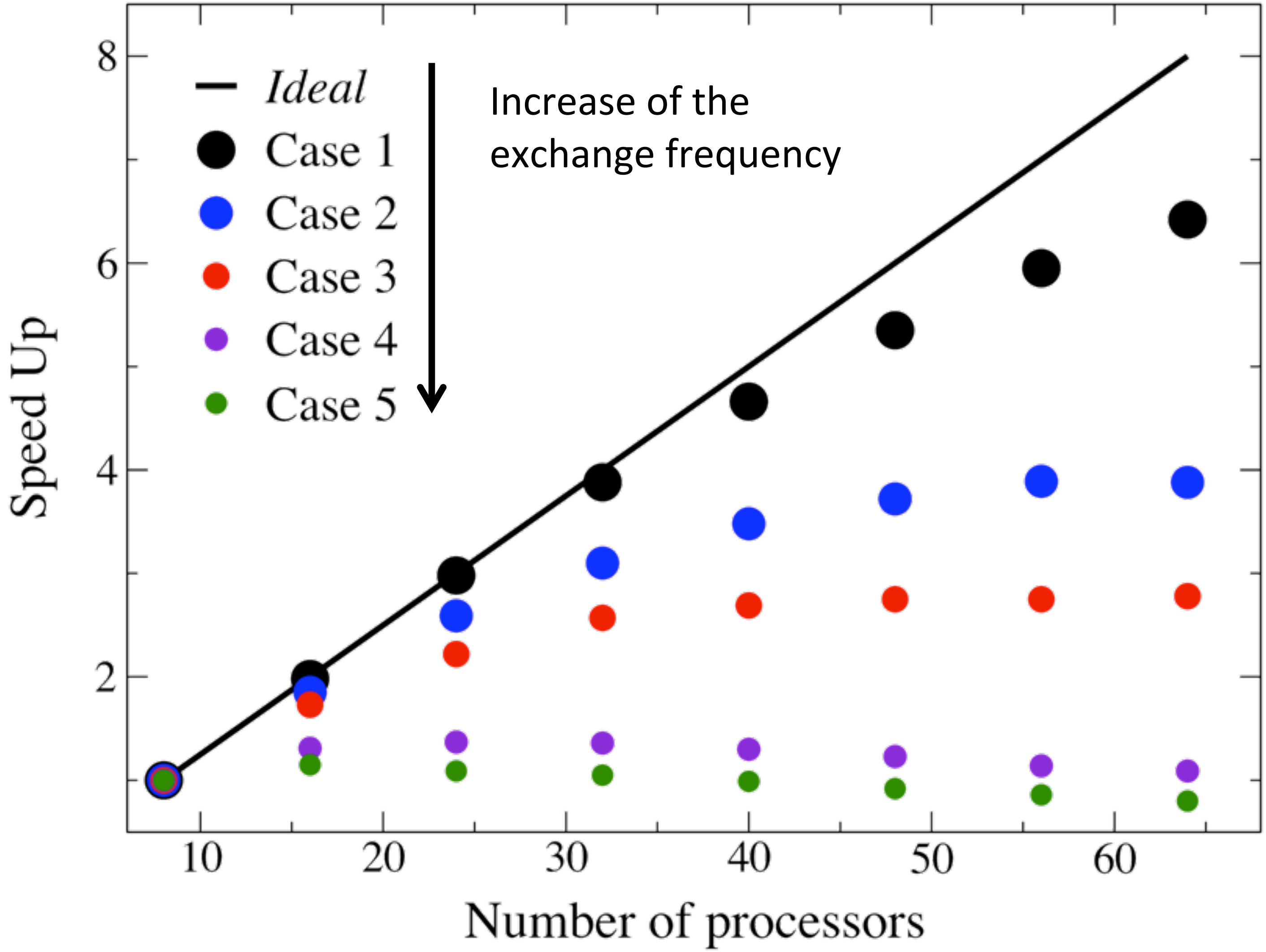


Solid solver: n_s cores on N_s are concerned by the coupled interface

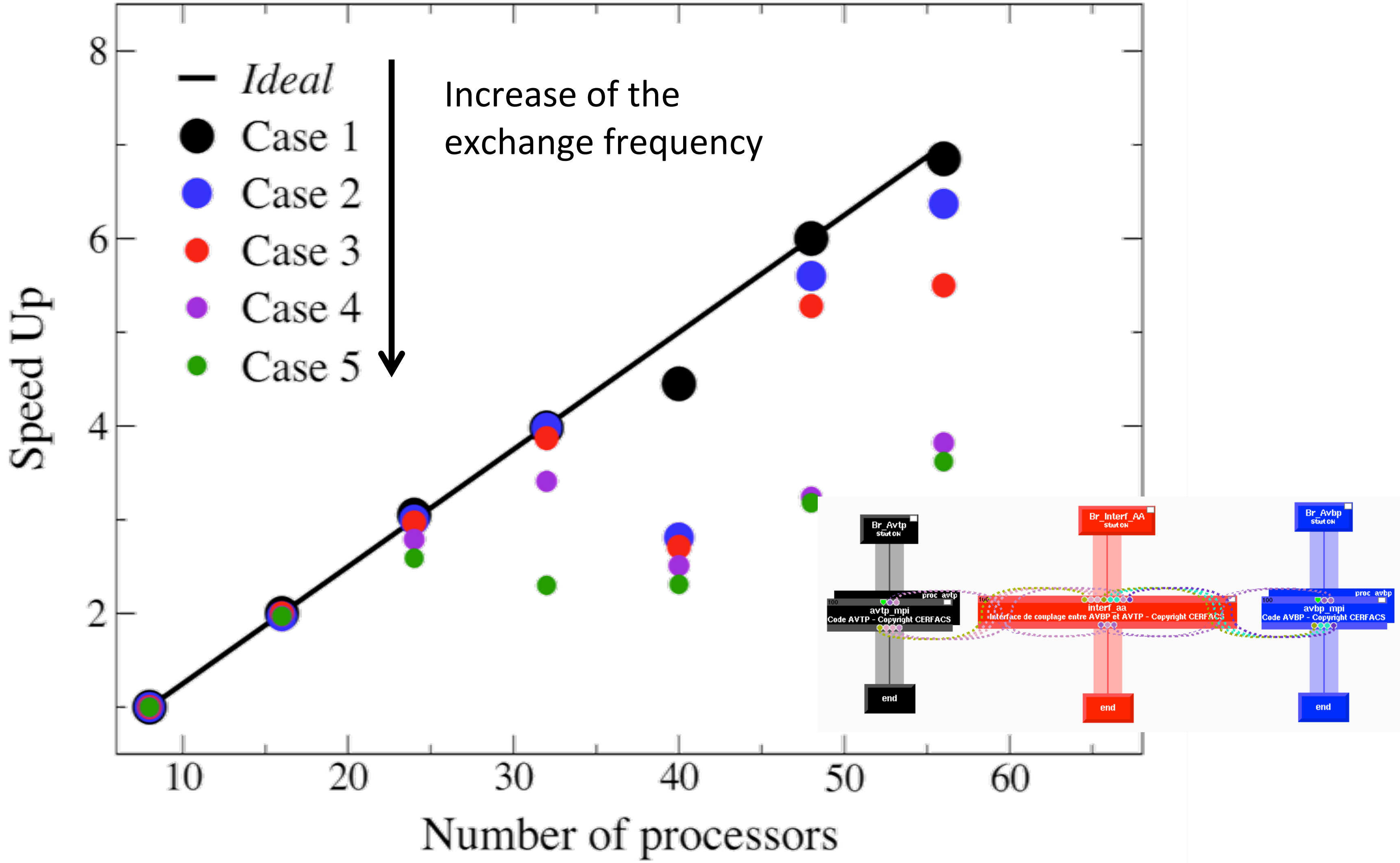
- (*) Temporal loop in the solvers: exchange of physical quantities

(*) can either be done by concentrating information on one process or in a distributed way

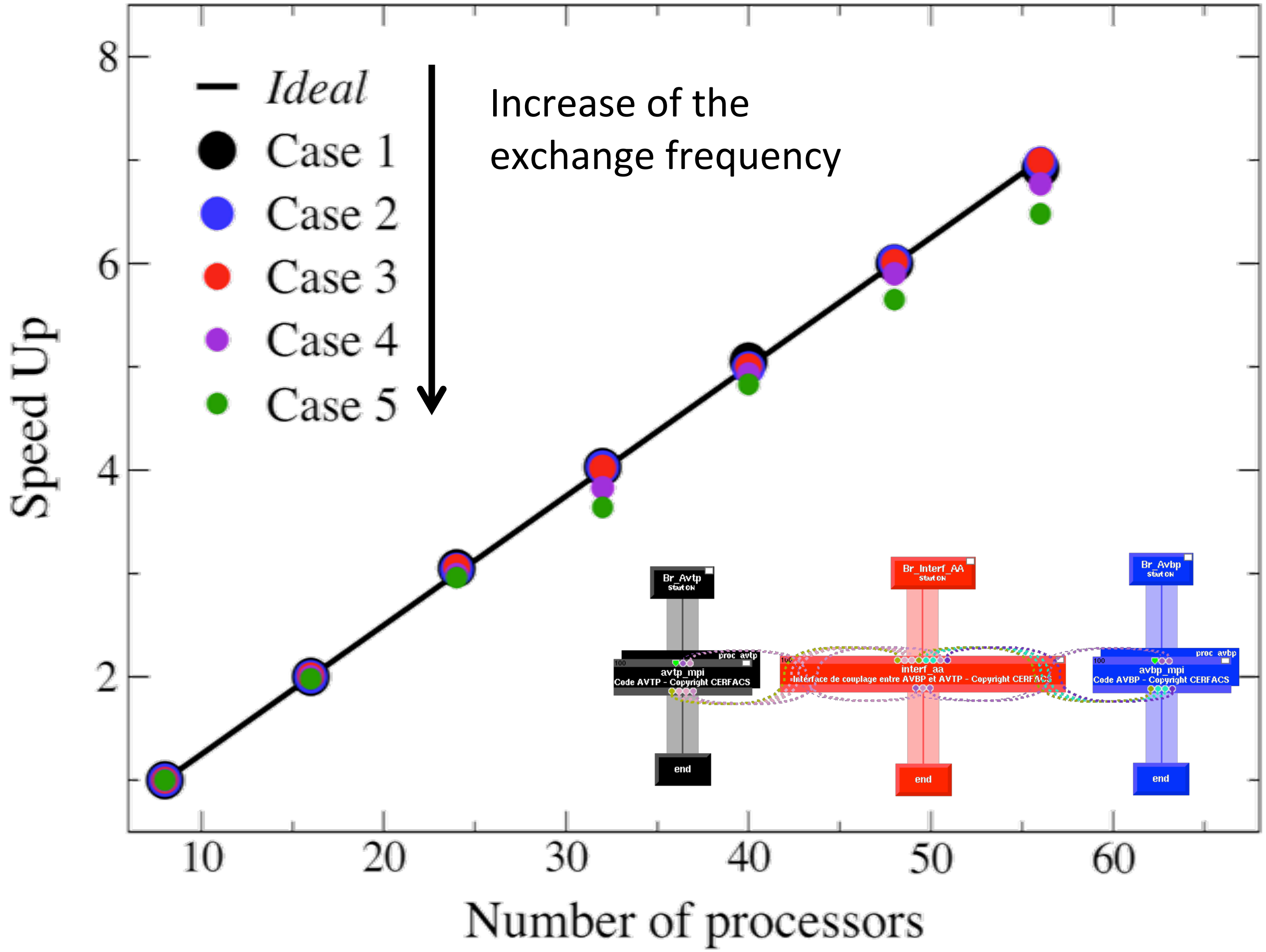
Code sequencing + file exchange: example of a conjugate heat transfer application (LES + conduction)



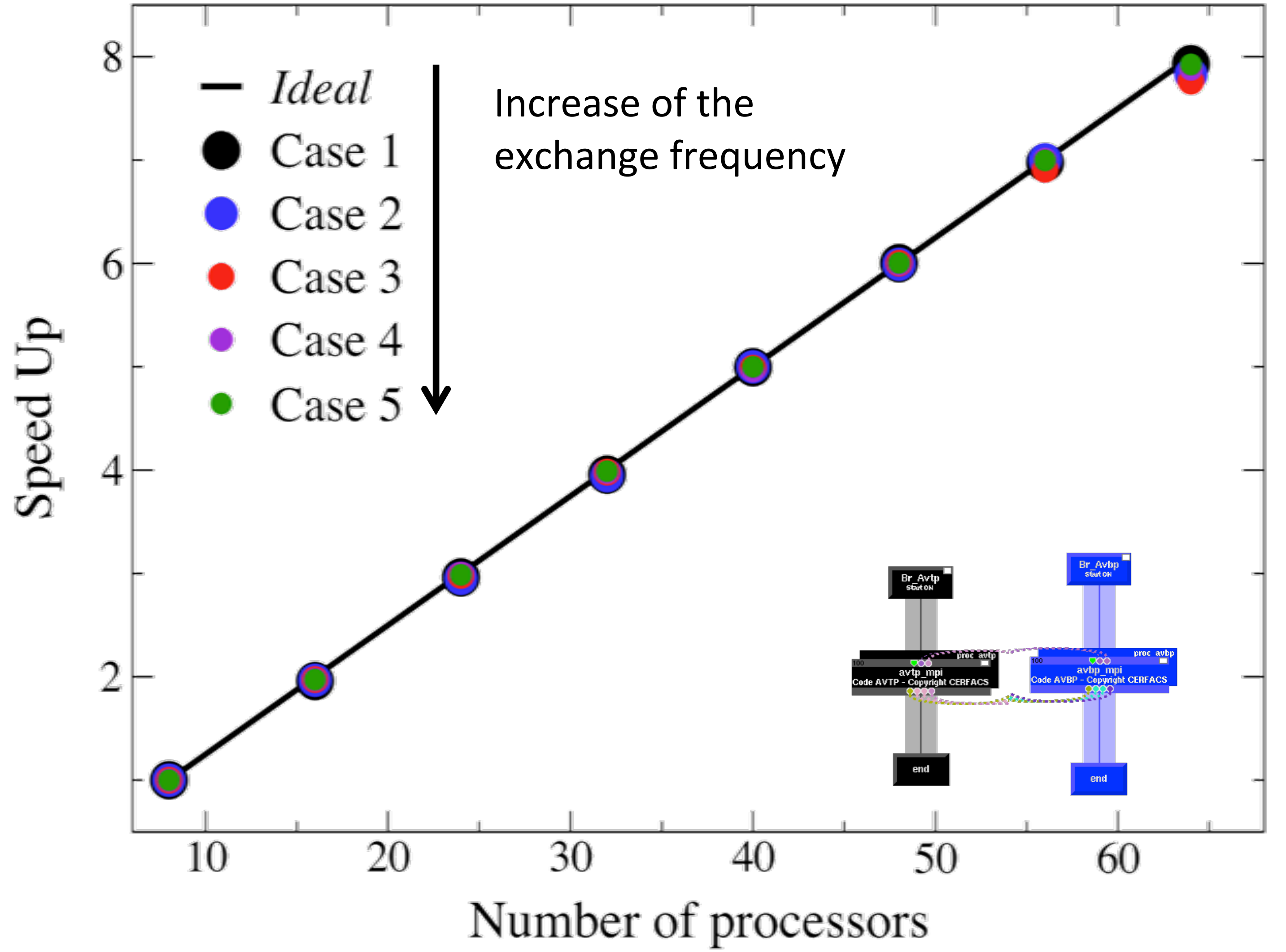
Coupler + merge of the information: example of a conjugate heat transfer application (LES + conduction)



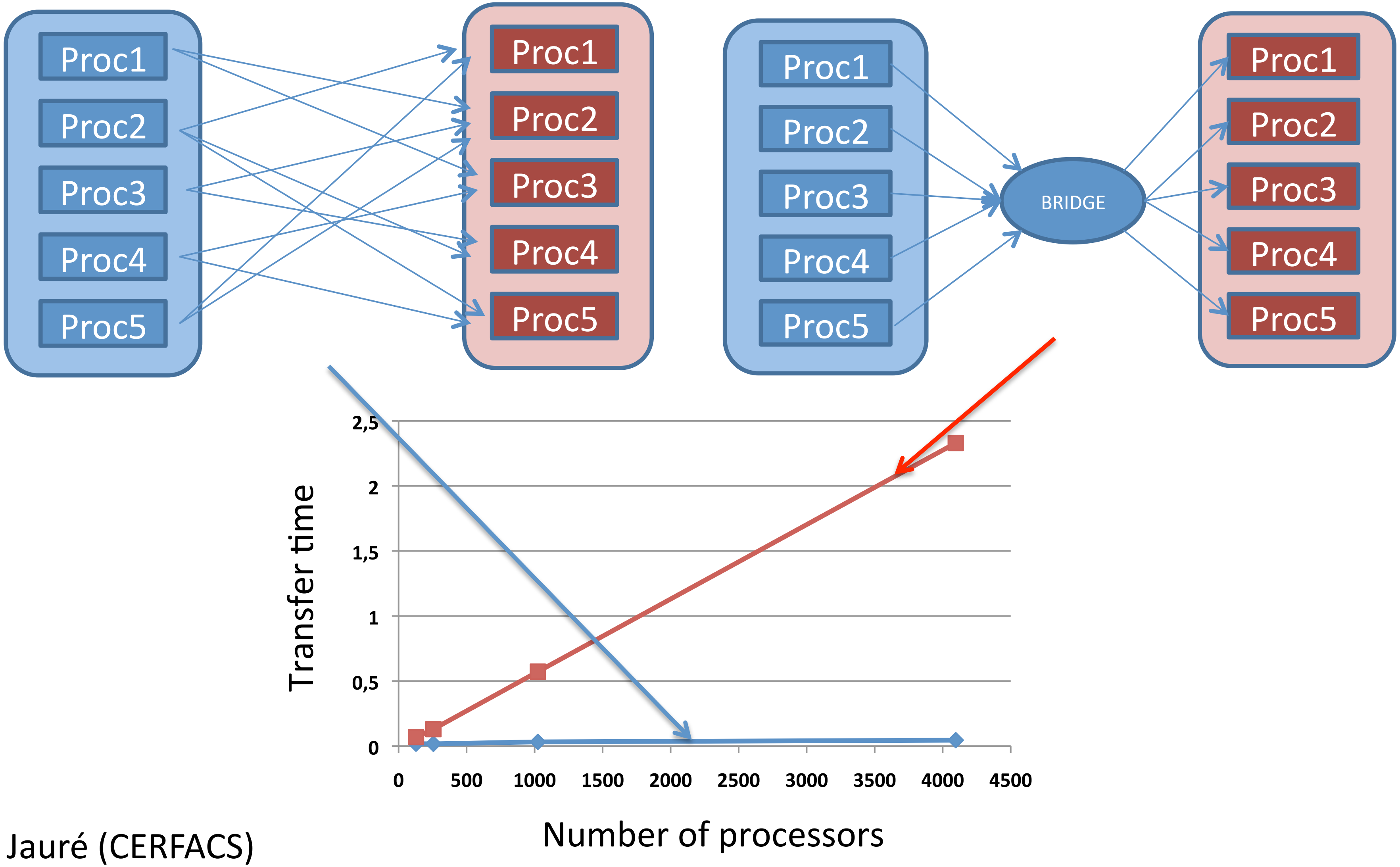
Coupler + merge of the information + direct communications : example of a conjugate heat transfer application (LES + conduction)



Coupler + fully distributed direct communications : example of a conjugate heat transfer application (LES + conduction)



Direct communications are mandatory for HPC



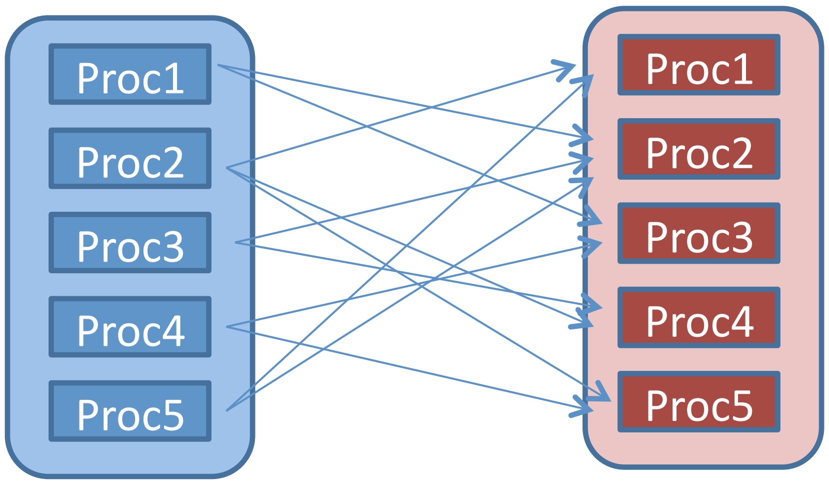
S. Jauré (CERFACS)

To establish efficiently the routing between the cores, it is necessary to consider:

- CPU time of the solutions (weak limit)
- number of communications (weak limit)
- memory allocation (strong limit)

⇒ avoid a master / slaves scheme with information merging

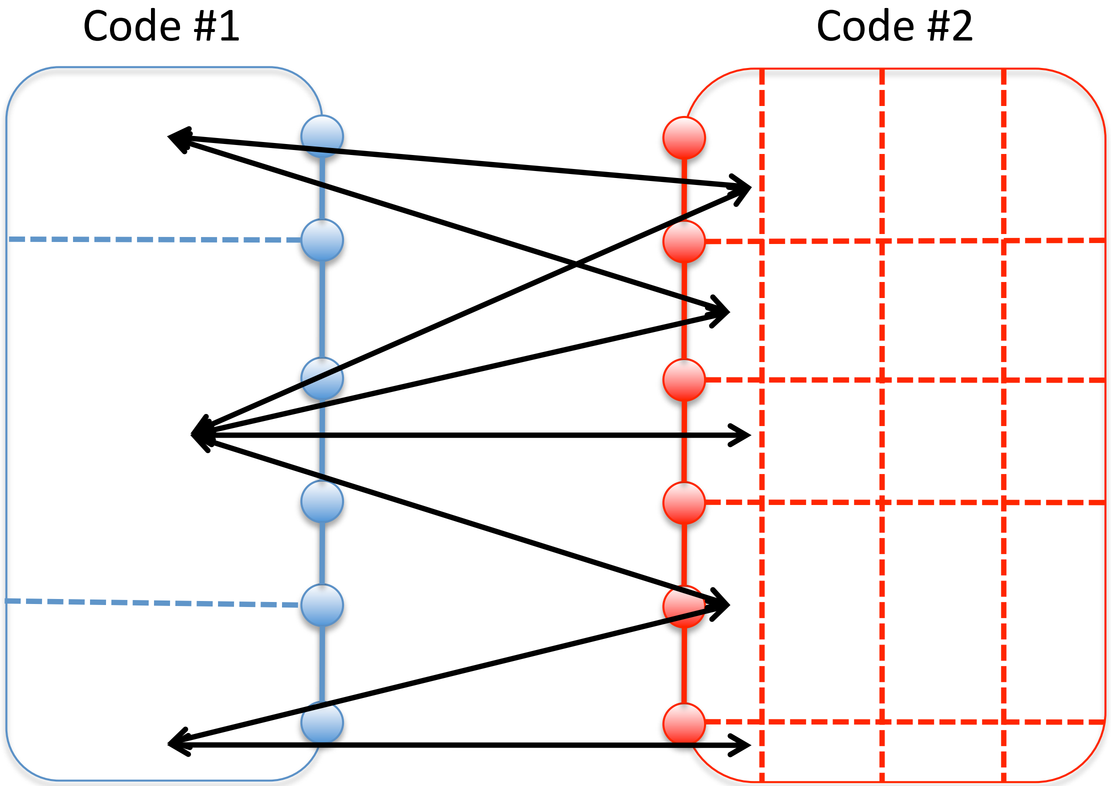
⇒ investigate solutions with different servers, until *Peer 2 Peer*



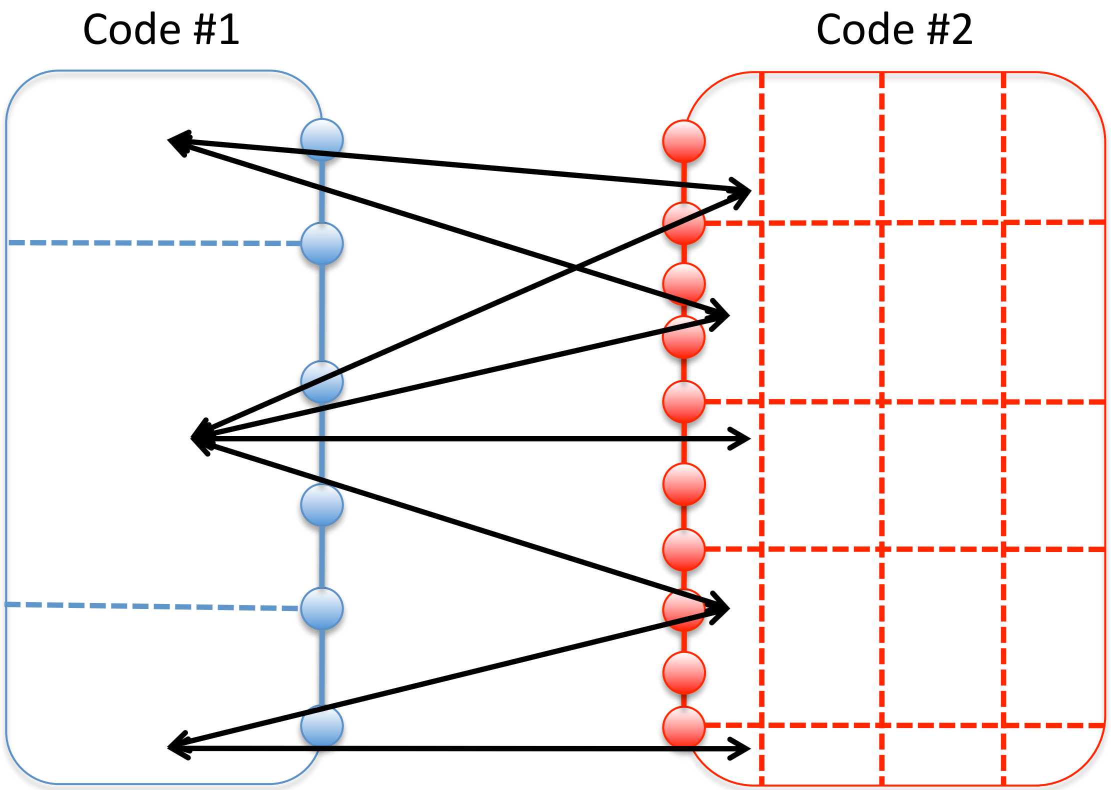
The problem of routing consists in determining

<mode I’m a core of a code> with which cores of the other(s) code(s) I need to receive and send information <end mode>

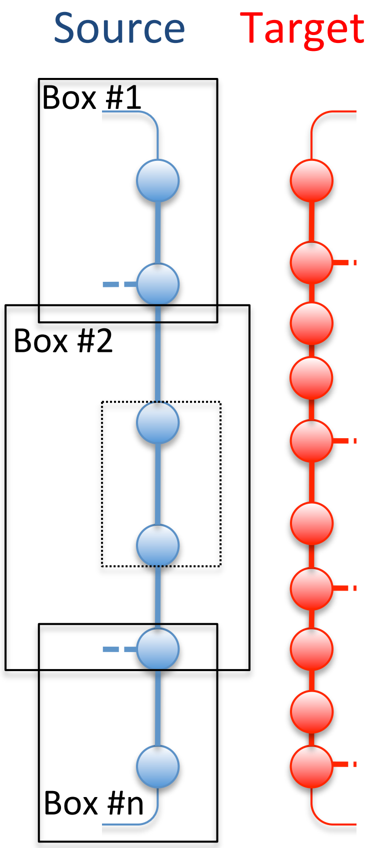
=> Geometrical localization



If the position of the nodes are not coincident: geometrical localization in the vicinity (+interpolation process)

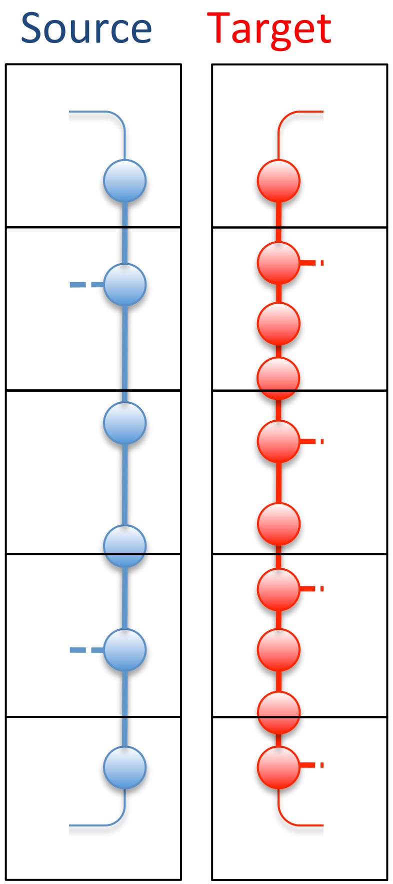


Geometrical localization in the vicinity: example #1 (Errera *et al.* 2010)



- Step 1: definition of the source mesh (nodes & connectivity), and of the target mesh (nodes & connectivity)
- Step 2: each process of the the source code defines a surrounding box of its partition
- Step 3: each process of the source code checks for geometrical intersections with target nodes => determination of a reduce number of target nodes and a first communication graph
- Step 4: for each process classification of the target nodes in an octree structure
- Step 5: for each process, definition of a sub-box per element of the source mesh
- Step 6: intersection between each sub-box and the corresponding target nodes thanks to the octree => refinement of the number of target nodes
- Step 7: for each target node, identification of the closest element of the source code and definition of the final communication graph

Geometrical localization in the vicinity: example #2 (Jauré *et al.* 2011)



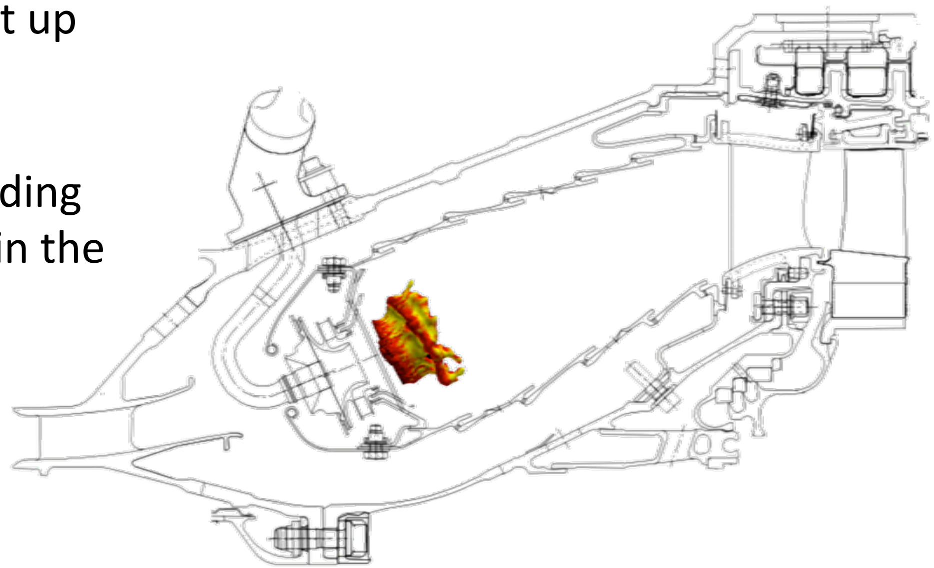
- Step 1: definition of a coarse uniform grid (i,j,k) that includes the interface meshes
- Step 2: mapping of the nodes (x,y,z) of the source and target codes to the cells (i,j,k)
- Step 3: construction of a hash table (HT) based on the cell decomposition (association between cells and processors)
- Step 4: on the source code, distribution of the HT on number of master processors such that 1 < mp < nb proc of the solvers !
- Step 5: the target code interrogates the distributed HT to define a first communication graph
- Step 6: for each target node, identification of the closest node of the source code (KD tree) and definition of the final communication graph

Profiling of a conjugate heat transfer computation for the determination of wall combustor temperature

Solver	# cells	# nodes	# interface nodes	# iterations
Fluid (AVBP)	4 511 357	846 245	74 644	150
Solid (AVTP)	2 806 834	596 205	237 524	200

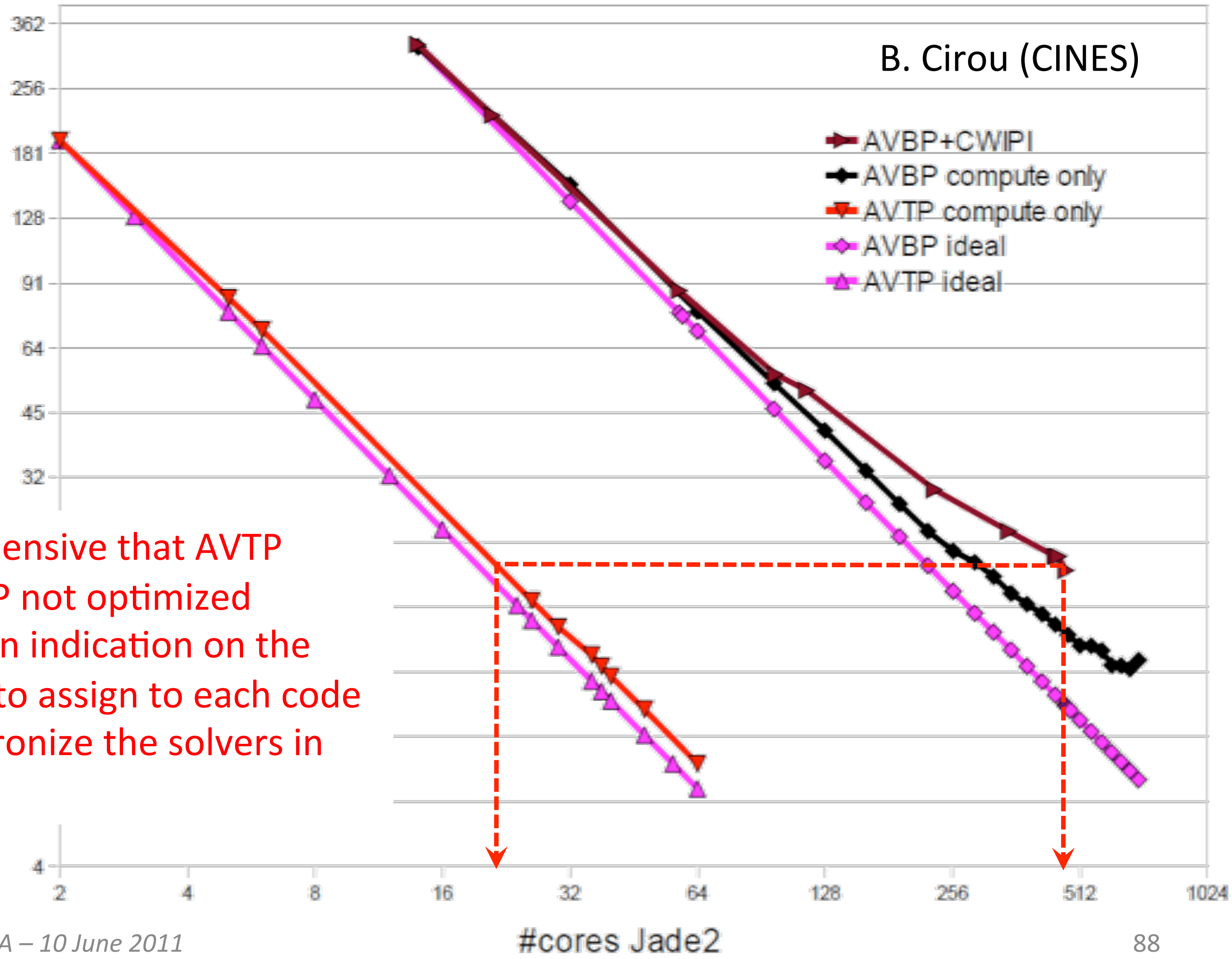
⇒ rather small configuration designed to set up the profiling methodology

⇒ very small cells in the combustor wall leading to an important number of interface nodes in the solid domain



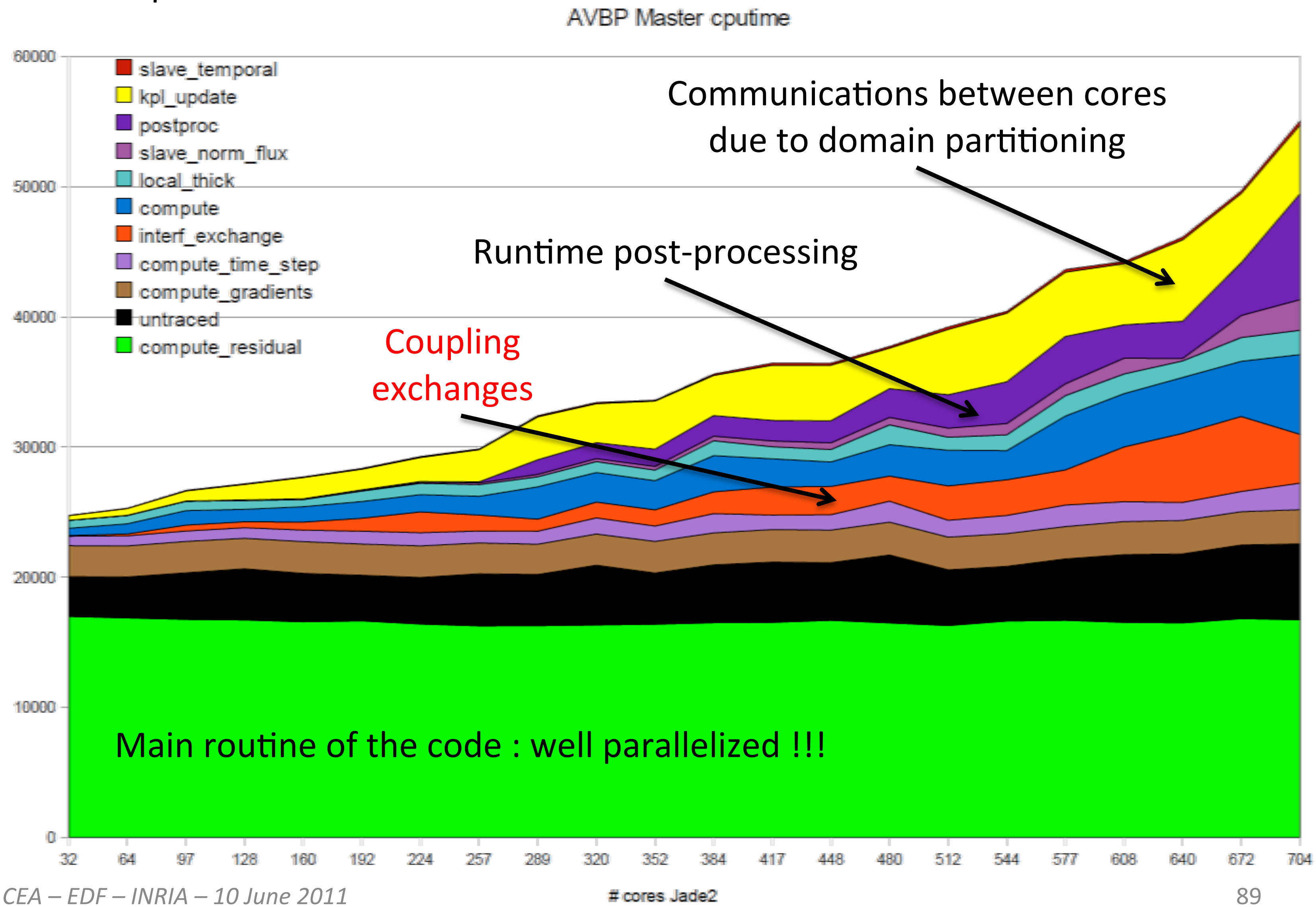
Profiling of a conjugate heat transfer computation for the determination of wall combustor temperature

# of seconds between 2 exchanges



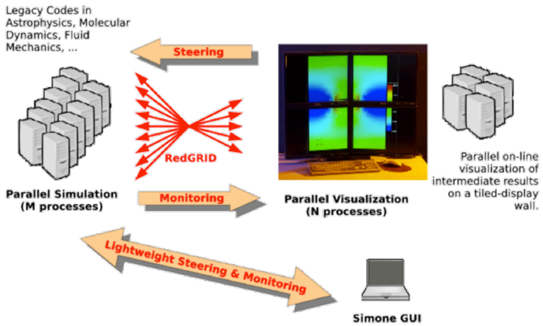
1) AVBP is more expensive that AVTP
2) Speed-up of AVBP not optimized
3) The graph gives an indication on the number of cores to assign to each code in order to synchronize the solvers in CPU time

Profiling of a conjugate heat transfer computation for the determination of wall combustor temperature

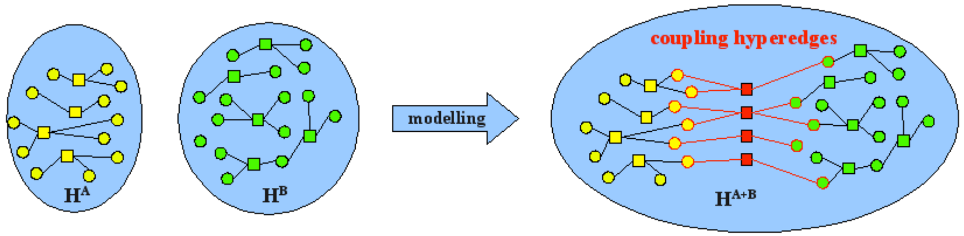


Open and interesting questions under investigation:

- Interpolation: high order / conservative interpolation in parallel
- Online visualization (EPSN – INRIA)



- Co-partitioning of the solvers to optimize the communication graph



- Charge equilibrium: avoid CPU lost when codes are waiting for the others

  ⇒ almost easy when the charge is constant during the simulation,
  ⇒ challenging when the charge evolves during the simulation

- Today: one tool (=one effort) for developments in climatology and CFD

1 Numerical developments in CFD for HPC

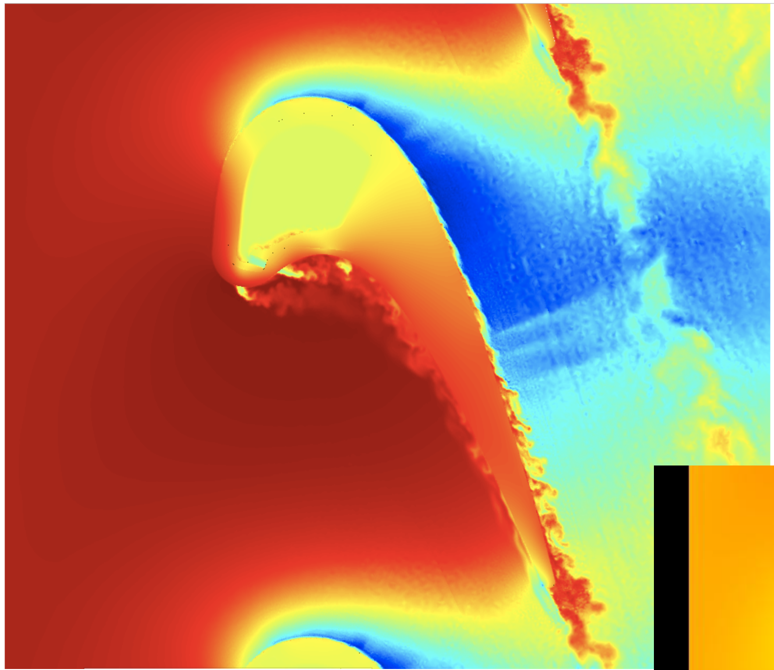
- CFD
- Flow solver examples
- Speed-up and Mesh-partitioning
- Communication, Impact on numerical solutions
- Applications to aeronautic challenges

2 Code coupling

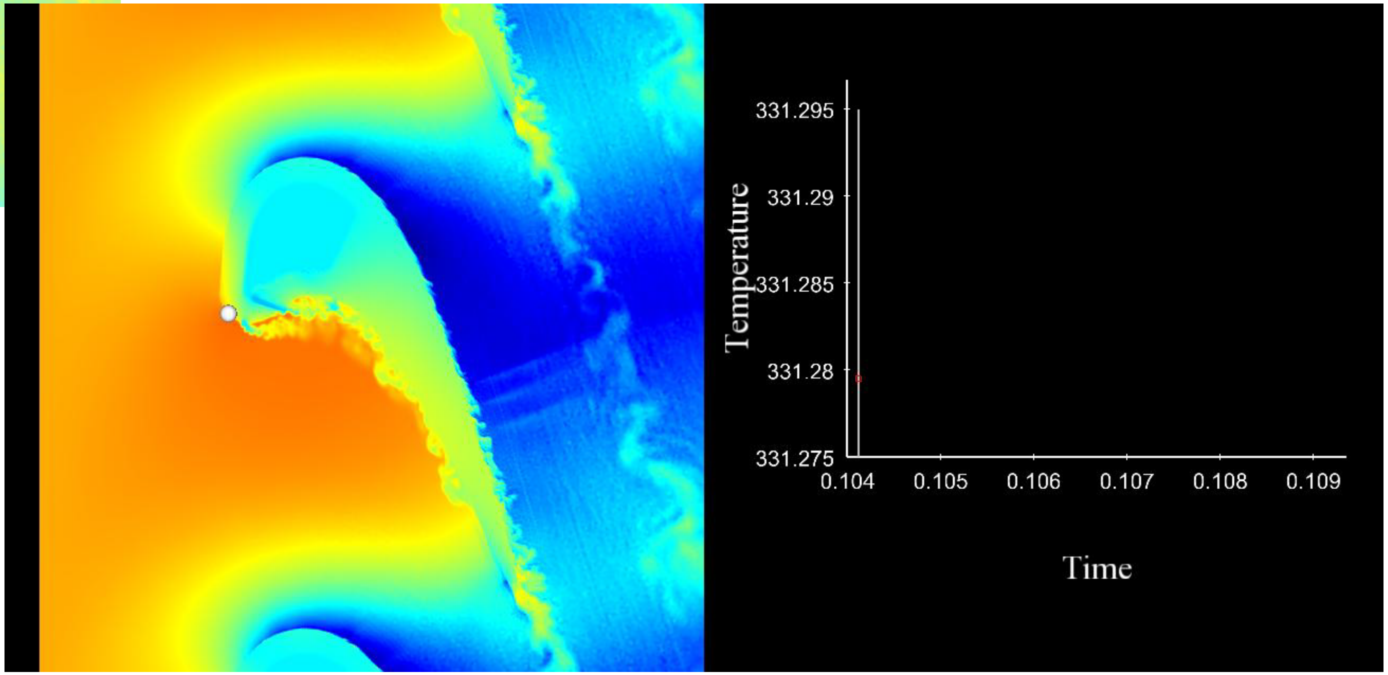
- Why multi-physics simulations
- Physical and numerical issues
- HPC issues
- **Applications to aeronautic challenges**

3 Conclusion and perspectives

Aerothermal applications at CERFACS: LES + conduction in solid

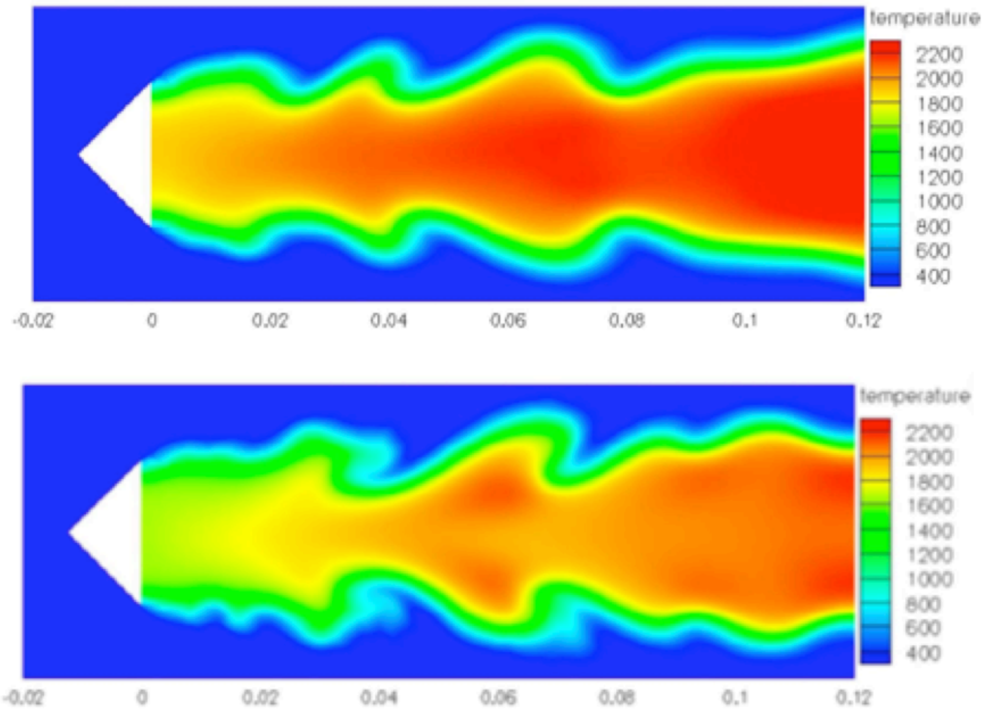


- Exchange of data on surfaces,
- Wall in the fluid part need an important resolution to capture heat transfer (=> very big meshes)
- Interpolation



F. Duchaine (CERFACS)

Aerothermal applications at CERFACS: LES + radiation



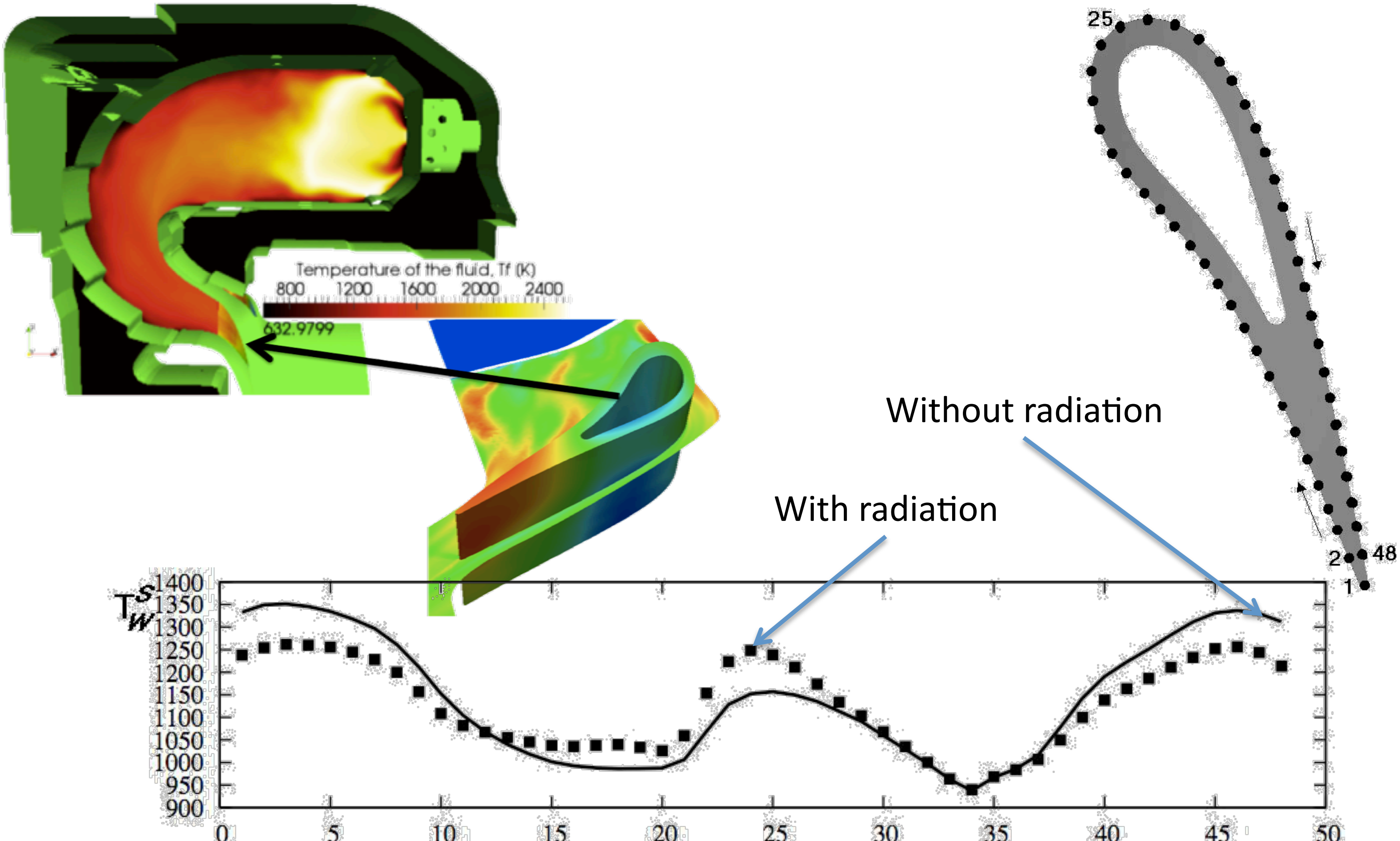
D. Poitou (CERFACS)

- Exchange of data on volumes,
- Very different grid resolution for the fluid and the radiation solvers

  ⇒ high resolution to capture turbulence and the flame front

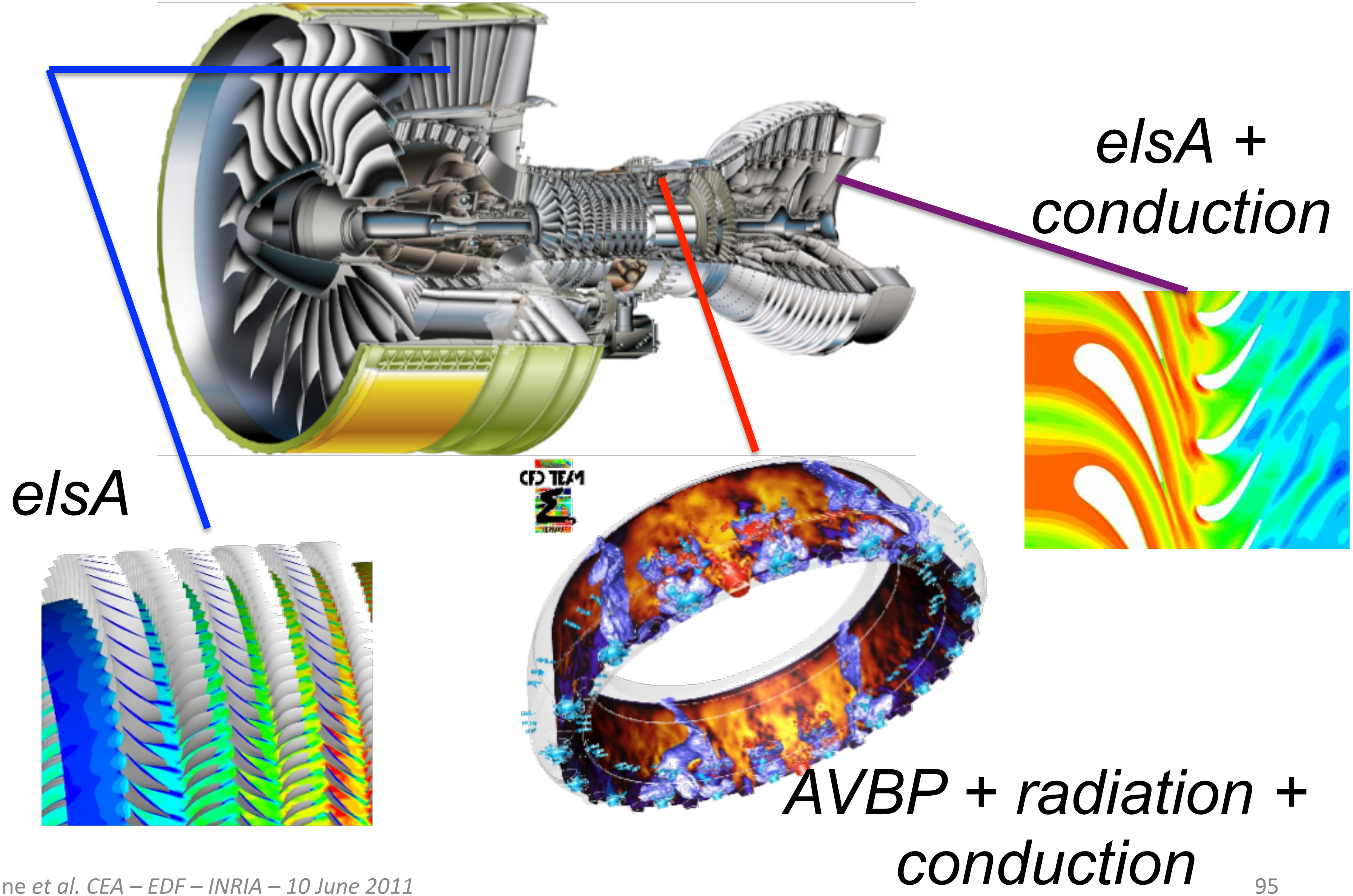
  ⇒ radiation is a long distant physic
- Interpolation

Aerothermal applications at CERFACS: LES + radiation + conduction



J. Amaya (CERFACS)

Future applications at CERFACS: multi-component and multi-physic simulations



- Examples of applications have been presented for aeronautic and propulsion domains,
- The estimation of the parallel efficiency is complex in industrial context:
  - ➢ the most relevant indicator is the time needed to obtain the solution
- High-fidelity simulations allowed by HPC improve the numerical solution reliability
  - ➢ clear impact on industrial application
  - ➢ clear impact for fundamental research
- For aeronautic industry, CFD is a key technology for design, time and cost developments,
- It is also a very effective tool for investigating complex flow phenomena,
  - ➢ need to go for fully unsteady flow simulations
- Multi-physics and multi-components simulations are mandatory to
  - ➢ improve the comprehension of complex phenomena
  - ➢ improve the predictions

➔ Effort have to be done to ensure the scalability of the codes as well as of the coupled applications on massively parallel architectures

Thanks for your attention

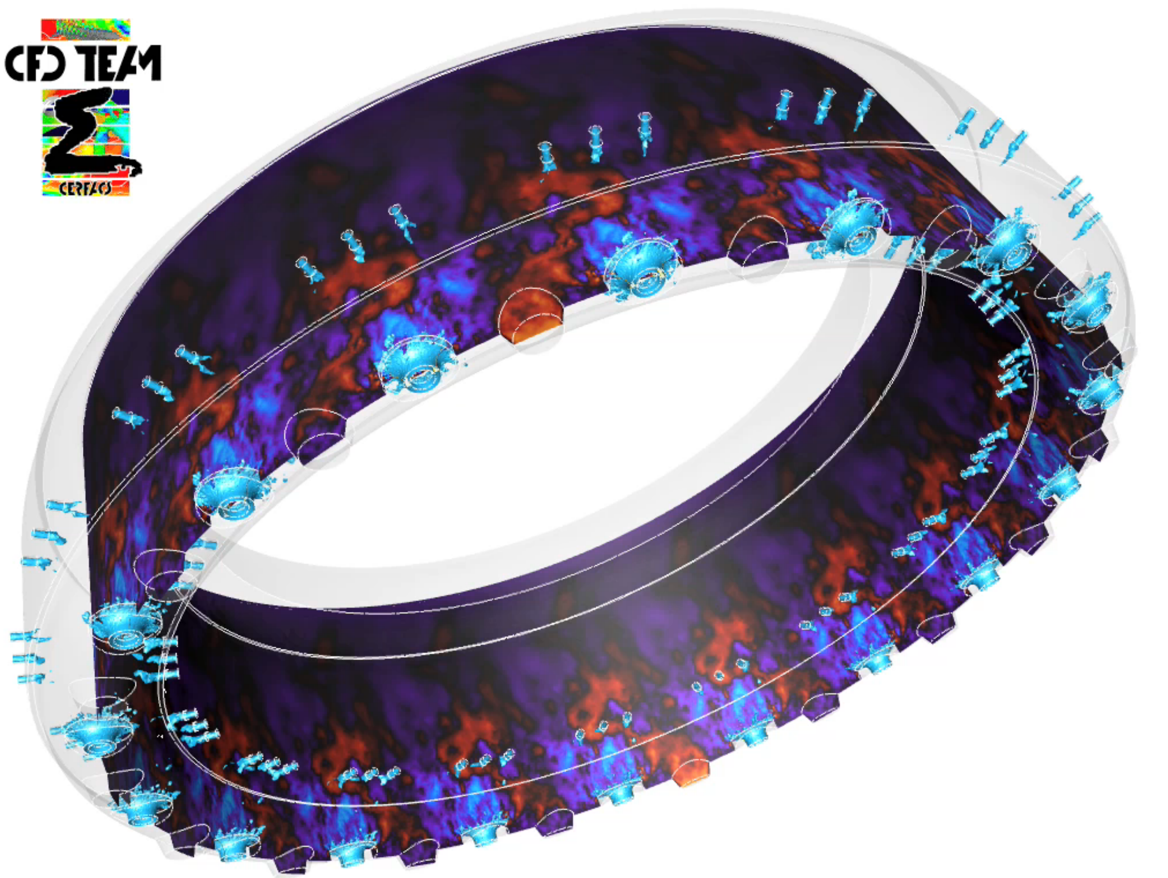
Thanks to the CERFACS GLOBC and CFD teams, and our partners!

http://www.cerfacs.fr

http://www.cerfacs.fr/~palm

Florent.Duchaine@cerfacs.fr

Gabriel.Staffelbach@cerfacs.fr



Ignition sequence of an annular burner – M. Boileau (CERFACS)