Multigrid Strategies Coupled with Anisotropic Mesh Adaptation

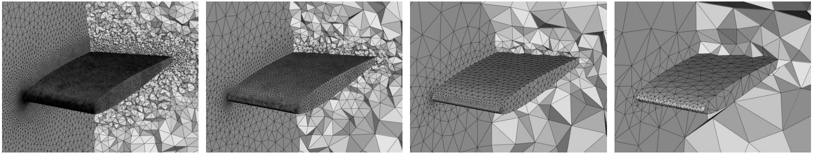
Victorien Menier[1], **Adrien Loseille**[2] and **Frédéric Alauzet**[2]

[1]PhD student, Gamma3 team, Inria Paris-Rocquencourt, recipient of a PHD grant from the Airbus Group Foundation

[2]Researchers, Gamma3 team, Inria Paris-Rocquencourt

- We want to couple **multigrid strategies** with anisotropic mesh adaptation.
  - Widely used for algebraic problems since the 80's
  - Efficiently solve problems from PDE

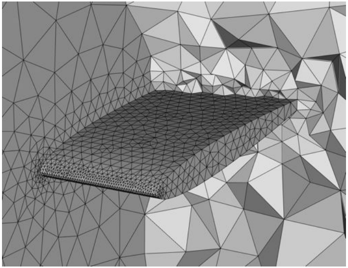
Main idea : Accelerate the convergence of the numerical solution by computing corrections on coarser meshes.

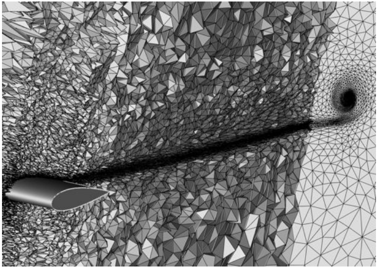


Scope of this presentation

- We want to couple multigrid strategies with **anisotropic mesh adaptation**.
  - Improves the accuracy of the solution
  - Reduces the computational time

Main idea : Modify the discretization of the computational domain to control the accuracy of the solution





Why coupling the two methods?

- At each stage in the mesh adaptation loop, meshes are generated with an increasing complexity and a single-grid computation is performed.
- Reasons to couple it with multigrid methods:
  1. The adaptive process is well suitable for multigrid computations.
     - A sequence of meshes is already generated
     - Anisotropic coarsening is handled automatically
  2. The full multigrid theory shows interesting properties regarding the convergence of the solution.

Outline

1. Implementation of an implicit multigrid procedure
2. Validation study in the non-adapted case
3. Coupling multigrid with mesh adaptation
4. Numerical Results

- **Objective:** Accelerate the convergence of a CFD simulation, thanks to a correction of the solution computed on coarser meshes.
  - Implicit time integration
  - A linear system is solved at each solver iteration

Accelerating the Newton method for solving the linear system can dramatically improve the total wall clock time of the simulation.

- A linearized system is solved **at each solver iteration**

Modeling equations: Euler equations

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) & = & 0\,, \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p & = & 0\,, \\ \dfrac{\partial (\rho e)}{\partial t} + \nabla \cdot ((\rho e + p)\mathbf{u}) & = & 0\,, \end{cases}$$

Vectorial form:

$$W_t + F_1(W)_x + F_2(W)_y + F_3(W)_z = 0 \quad (1)$$

- Spatial discretization:
  - Vertex-centered finite volume formulation
  - HLLC upwind schemes
  - 2nd order space accuracy: MUSCL procedure

- A linearized system is solved **at each solver iteration**

Linearized time discretization of (1) :

$$\left(\frac{|C_i|}{\delta t_i^n} I_d - \frac{\partial R_i}{\partial W}(W^n)\right)\left(W_i^{n+1} - W_i^n\right) \;=\; R_i(W^n)$$

written as :

$$\mathbf{A}^n\,\delta\mathbf{W}^n = \mathbf{R}^n \tag{2}$$

The Newton method used for solving (2) is a symmetric Gauss-Seidel (SGS) relaxation

Solving the linear system using a SGS relaxation

$\mathbf{A}^n$ is decomposed:

$$\mathbf{A}^n = \mathbf{L} + \mathbf{D} + \mathbf{U}$$

The linear system (2) is then approximated by:

$$(\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U})\,\delta\mathbf{W}^n = \mathbf{R}^n\,,$$

One SGS iteration consists in a forward and a backward sweep:

$$\begin{aligned}(\mathbf{D} + \mathbf{L})\,\delta\mathbf{W}^{k+1/2} &= \mathbf{R} - \mathbf{U}\,\mathbf{W}^k \\ (\mathbf{D} + \mathbf{U})\,\delta\mathbf{W}^{k+1} &= \mathbf{R} - \mathbf{L}\,\mathbf{W}^{k+1/2}\,.\end{aligned}$$

At each solver iteration, k_{max} SGS sub-iterations are performed in order to decrease the residual r of the system by a desired order of magnitude.

Multigrid for accelerating the Newton method

Objective: At a given solver iteration, accelerate the convergence of the Newton method by computing corrections on coarser meshes.

$$\mathbf{A}^n \, \delta\mathbf{W}^n = \mathbf{R}^n$$

- Single-grid: k_{max} SGS sub-iterations :

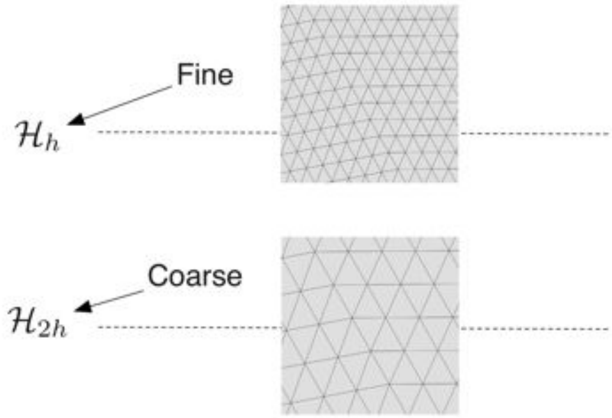
$$\delta\mathbf{W}_0^n \xrightarrow{SGS} \delta\mathbf{W}_1^n \xrightarrow{SGS} \delta\mathbf{W}_2^n \xrightarrow{SGS} \delta\mathbf{W}_3^n \xrightarrow{SGS} \dots$$

- Multigrid : n_{max} SGS sub-iterations with correction ($\mathbf{C}_i$) computations :

$$\delta\mathbf{W}_0^n \xrightarrow{SGS} \delta\mathbf{W}_1^n \leftarrow +\mathbf{C}_1 \xrightarrow{SGS} \delta\mathbf{W}_2^n \leftarrow +\mathbf{C}_2 \xrightarrow{SGS} \delta\mathbf{W}_3^n \leftarrow +\mathbf{C}_3 \dots$$

The corrections to the solution $\mathbf{C}_i$ are computed using multigrid cycles. The simplest one is the two-grid V-cycle.

Basic two-grid V-cycle

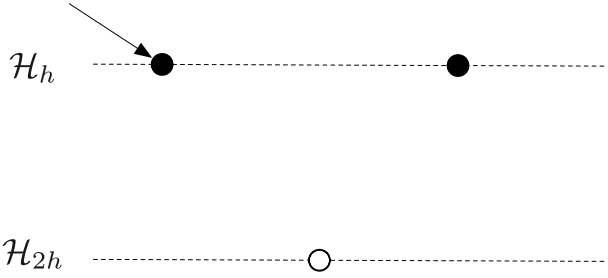


A_h and A_{2h} are built on $\mathcal{H}_h$ and $\mathcal{H}_{2h}$, resp.

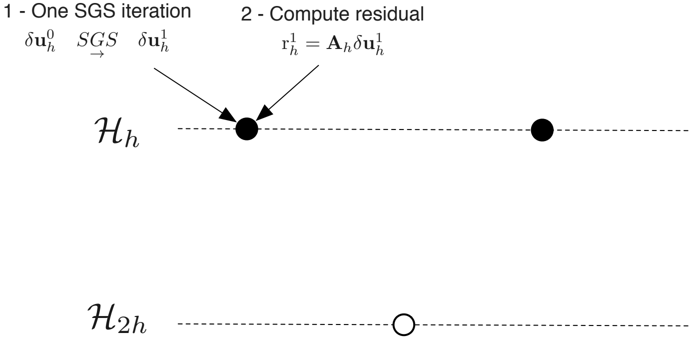
Basic two-grid V-cycle

1 - One SGS iteration

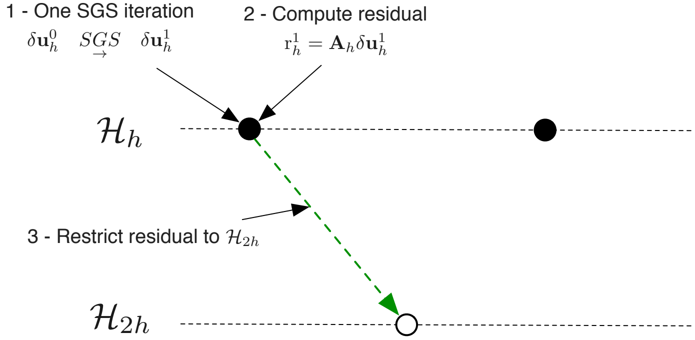
$\delta \mathbf{u}_h^0 \quad \underset{\rightarrow}{SGS} \quad \delta \mathbf{u}_h^1$



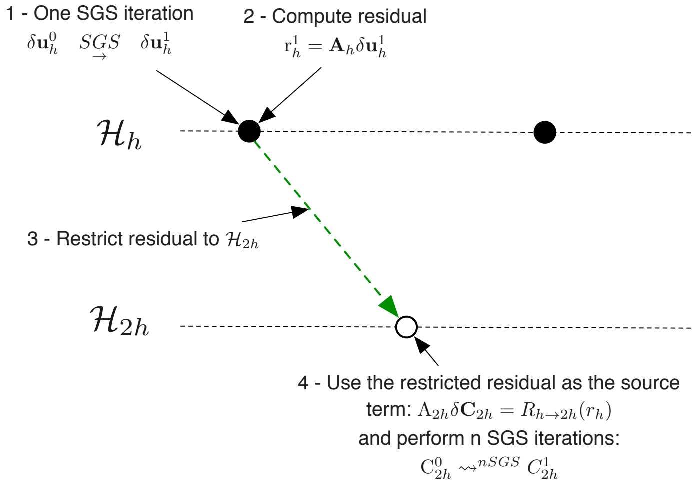
Basic two-grid V-cycle



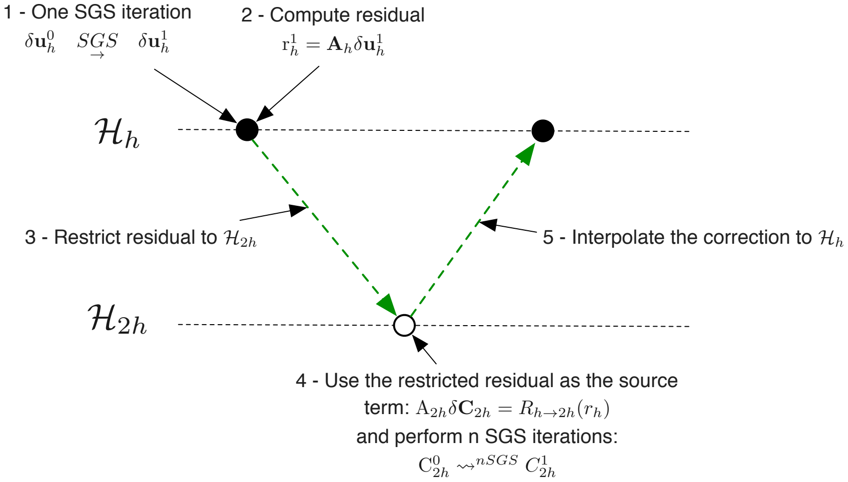
Basic two-grid V-cycle



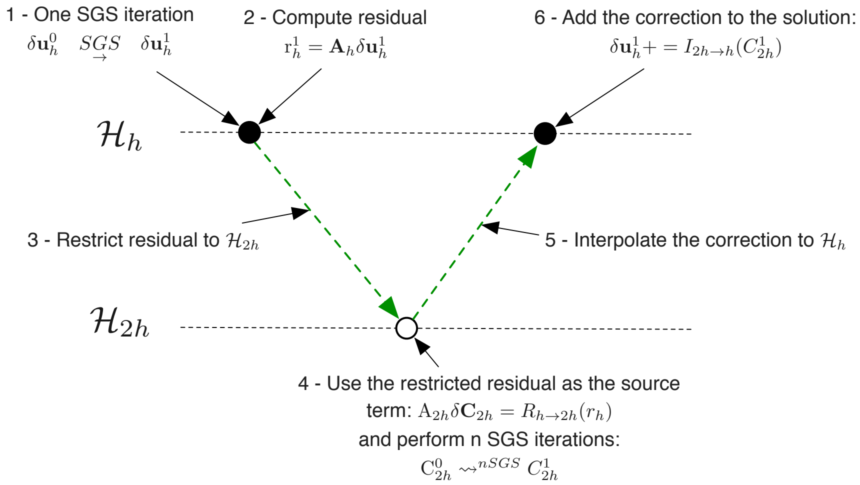
Basic two-grid V-cycle



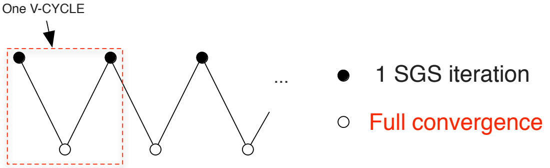
Basic two-grid V-cycle



Basic two-grid V-cycle



Ideal Bigrid V-cycle

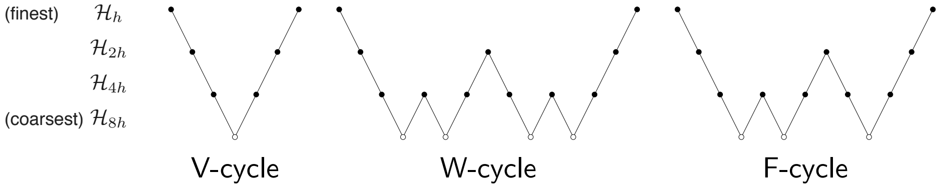


- An ideal bigrid V-cycle consists in fully converging the correction on the coarse mesh $\mathcal{H}_{2h}$ instead of performing n iterations.

- **Why?** It is CPU consuming, but it provides an idea of the best residual convergence in terms of the number of iterations that can be reached using multigrid.

- A "good" multigrid cycle aims at requiring as few iterations as the ideal bigrid cycle to decrease the residual on $\mathcal{H}_h$, while being less costly in terms of CPU thanks to the use of more coarser mesh levels.

Examples of multigrid cycles



• : 1 SGS iteration, ∘ : *N* SGS iterations

- Depending on the problem, different numbers of coarse levels or different cycle types might be used.
- In the presented examples, no post-smoothing SGS iterations.
- Some stiff problems may require several iterations of post- and/or pre-smoothing.

Outline

Multigrid validation : Two comparisons

- **Objective**: Compare single-grid simulations to a various set of multigrid cycles (V, W, F)

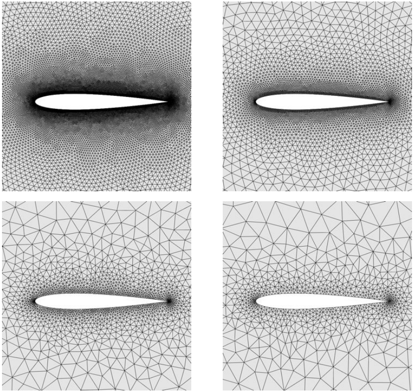
- Two different comparisons:

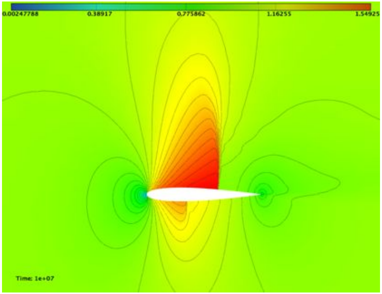
  1. At a given solver iteration, compare the residual convergence of the Newton method.
  2. Compare the impact on the whole simulation (residual convergence in terms of the number of solver iterations)

NACA 0012 (1/2) : At a given solver iteration

Comparison of the convergence of the system at solver iteration N

- Validation: 2D transonic NACA 0012 (Mach 0.8, $\alpha = 1.25$)
- 120 solver iterations have been performed $\rightarrow$ solution is 'almost' converged
- CFL highered from 10 to 1000.

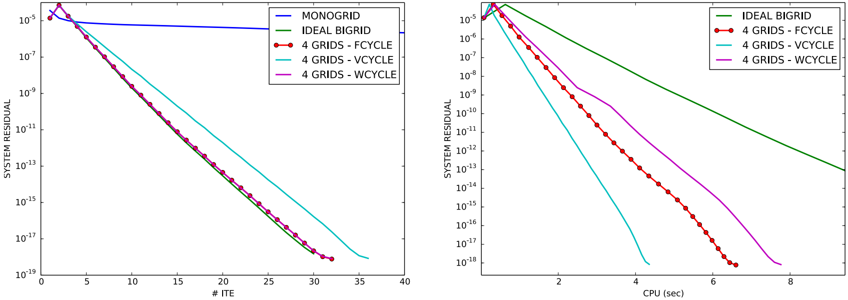




Close-up views of the four meshes used and solution.

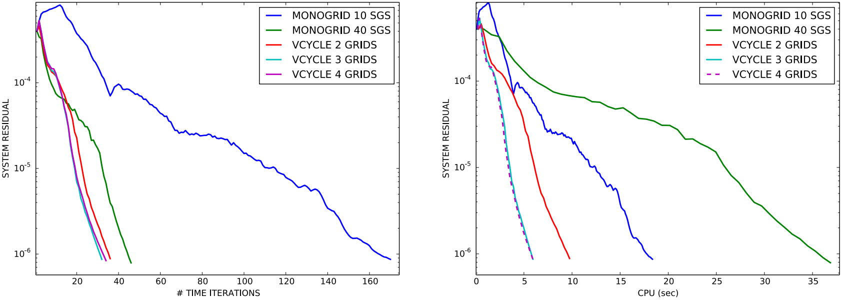
Comparison of the convergence of the system at solver iteration N

- Validation: 2D transonic NACA 0012 (Mach 0.8, $\alpha = 1.25$)
- 120 solver iterations have been performed $\rightarrow$ solution is 'almost' converged
- CFL highered from 10 to 1000.



Comparison of the convergence of the Newton method (solver iteration 120).

Impact of multigrid on the whole simulation.

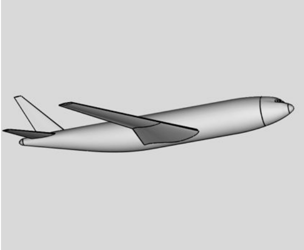


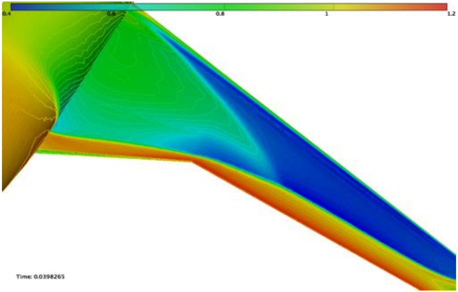
Comparison of the convergence of the whole simulation.

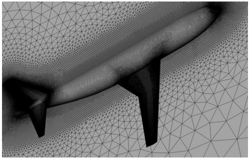
- Single-grid: No less than 10 SGS sub-iterations are required to converge
- Multigrid: 3 V-cycles are enough.
- Wall clock time drops from 20 sec to 7 sec.

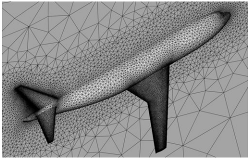
3D transonic WBT configuration (1/2)

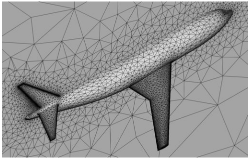
- 3D wing body tails (WBT) configuration
- Mach 0.8, $\alpha = 1.\,\mathrm{deg}$





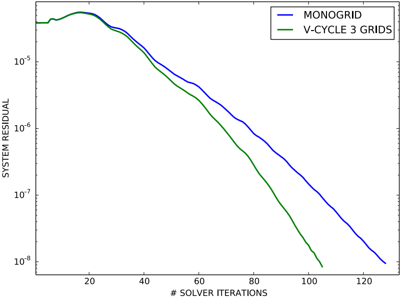




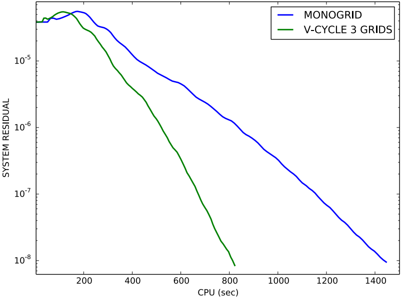


Impact of multigrid on the whole simulation

Solver iterations



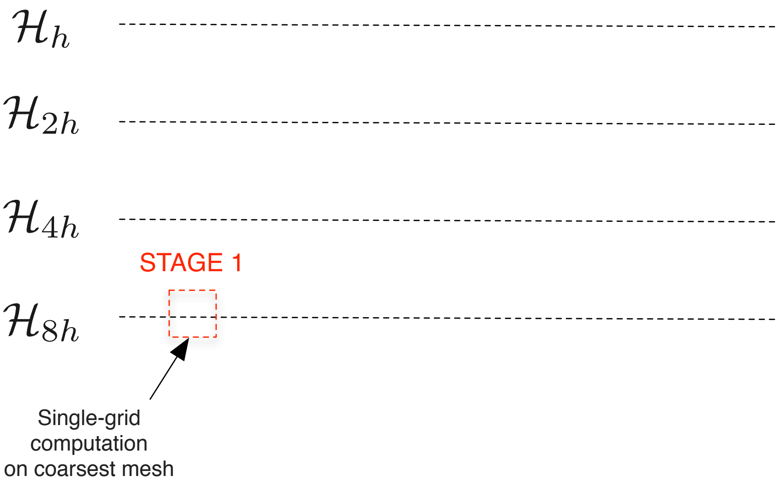
Wall clock time



Comparison of the convergence of the whole simulation.

- 3 V-cycles (multigrid) vs 10 SGS sub-iterations (single-grid).
- Total wall clock time drops from 24m9s to 13m30s thanks to multigrid.

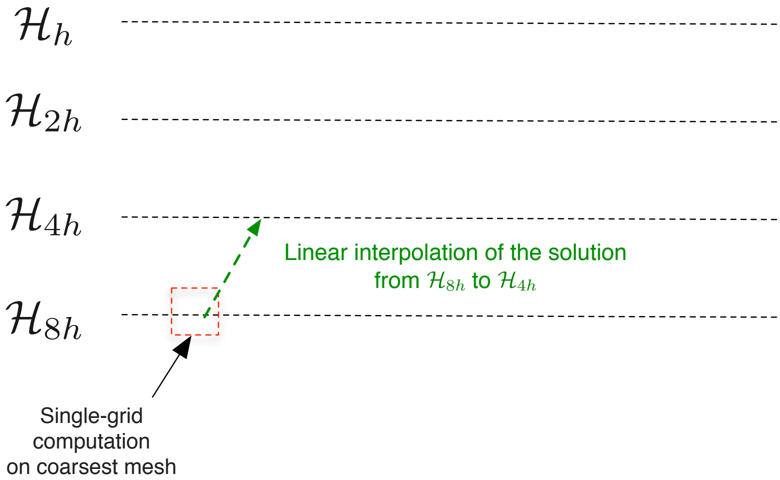
- Description of the FMG algorithm



Full convergence

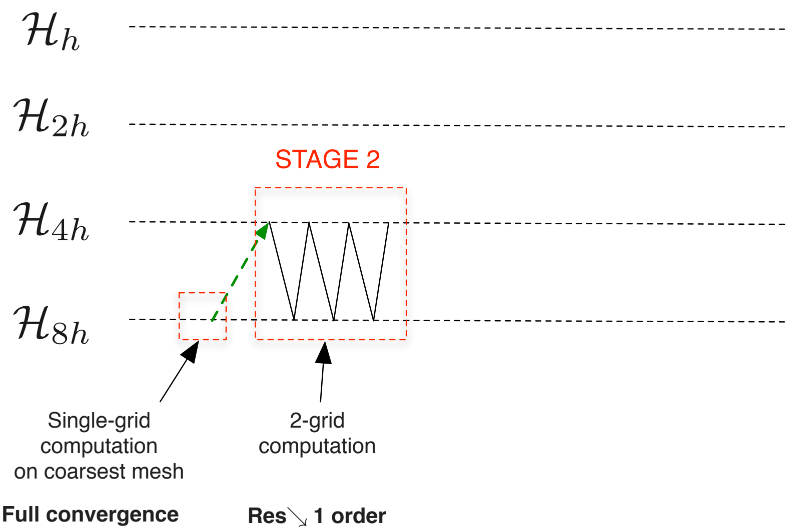
Full multigrid (FMG) (1/2)

- Description of the FMG algorithm

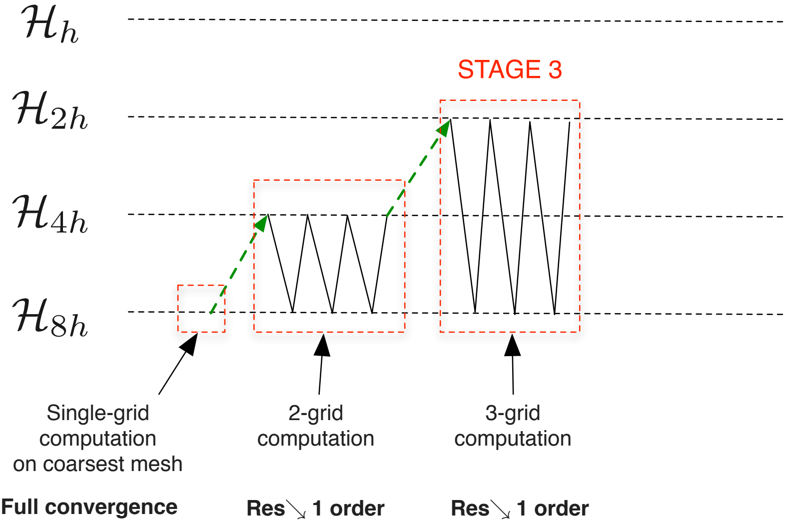


Full convergence

- Description of the FMG algorithm

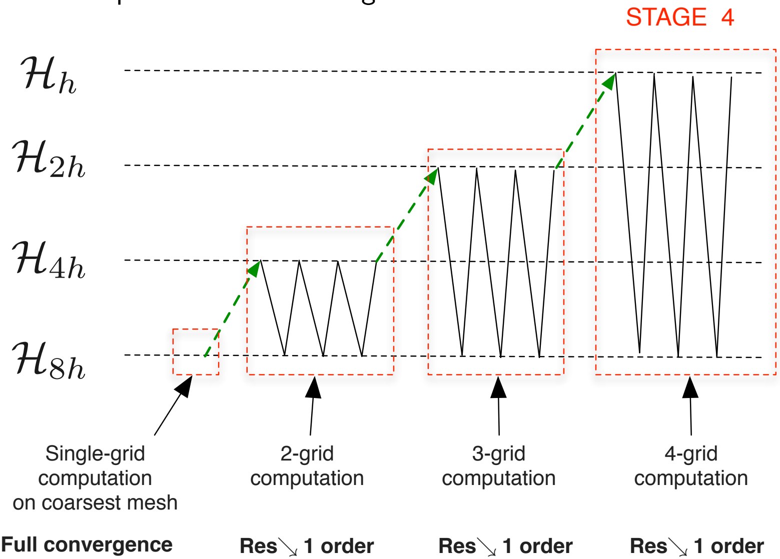


- Description of the FMG algorithm



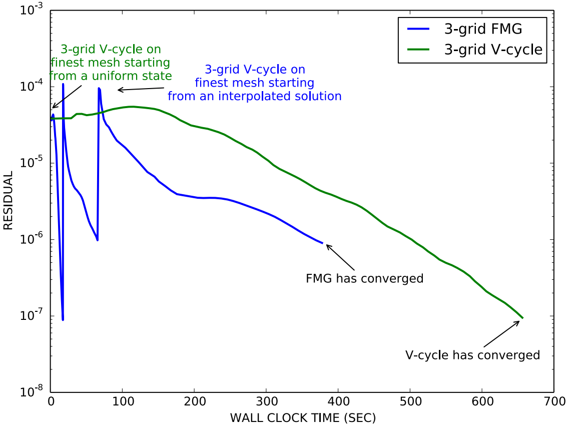
Full multigrid (FMG) (1/2)

- Description of the FMG algorithm

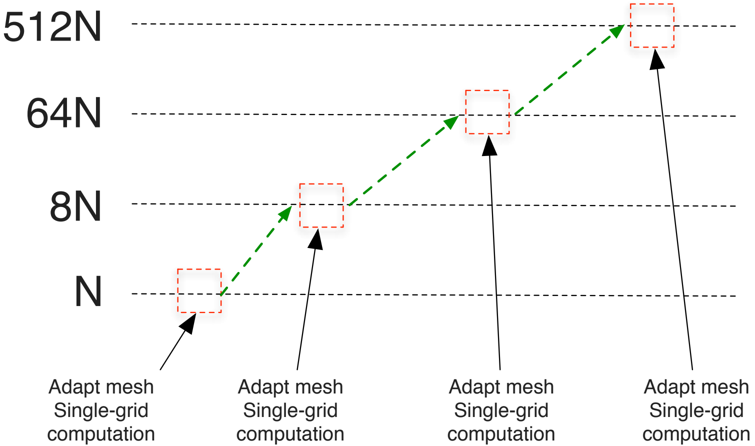


Full multigrid (FMG) (2/2)

- **Validation :** transonic WBT configuration
  - Comparison between a 3-grid V-cycle starting from a uniform state, and the FMG algorithm
  - A faster convergence is observed using the FMG method

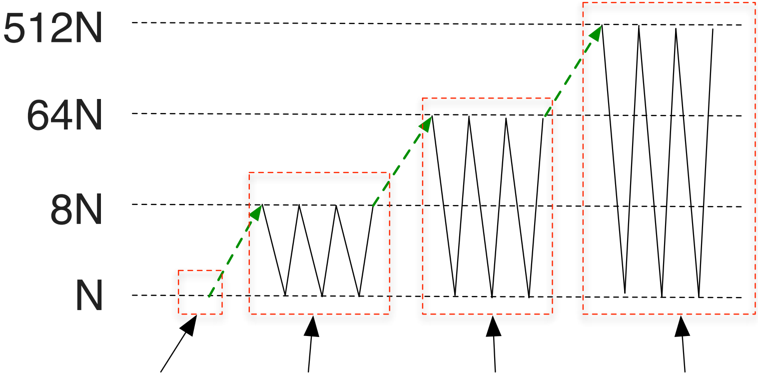


Mesh Adaptation Algorithm



Full convergence **Full convergence** **Full convergence** **Full convergence**

Coupling (1/2)



Adapt mesh Single-grid computation	Adapt mesh ~~Single-grid~~ **2-grid** computation	Adapt mesh ~~Single-grid~~ **3-grid** computation	Adapt mesh ~~Single-grid~~ **4-grid** computation
Full convergence	~~**Full convergence**~~ **Res ↘ 2 orders**	~~**Full convergence**~~ **Res ↘ 2 orders**	~~**Full convergence**~~ **Res ↘ 2 orders**

Coupling the two methods consists in performing multigrid simulations in the classical mesh adaptation loop, instead of single-grid simulations.

- **Why is it interesting?**

1. Adapted meshes are well-suitable for multigrid computations: anisotropic coarsening automatically handled.
2. FMG theory [Carre, 1999] $\Longrightarrow$ converging the solution by two orders of magnitude at each stage is enough to ensure the global convergence.
   - No need to fully converge at each stage, as it is usually done
   - Improves the total wall clock tine of the simulation

Outline

1. Implementation of an implicit multigrid procedure
2. Validation study in the non-adapted case
3. Coupling multigrid with mesh adaptation
4. Numerical Results

3D subsonic NACA 0012 : Mach 0.6, $\alpha = 2\,\text{deg}$.

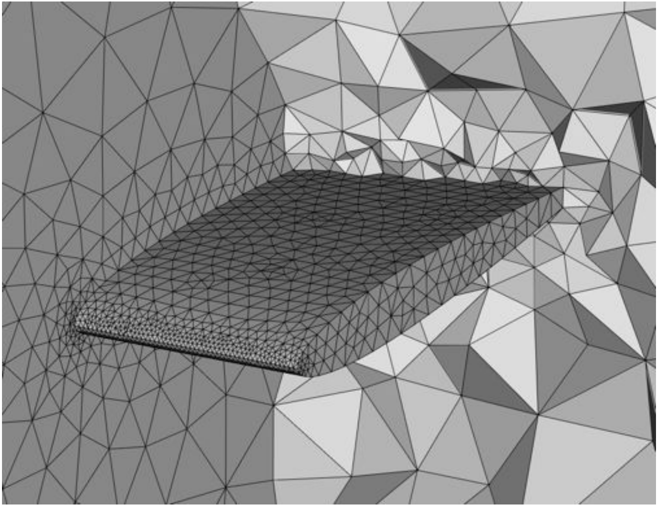
- **Two simulations** were performed:
  1. A classical mesh adaptation
     - Full convergence at each stage
  2. An adaptive FMG algorithm
     - Full convergence at stage 1, two orders of magnitude at stages 2, 3, etc.

For both simulations, prescribed mesh complexities lead to the following number of vertices:

$$8k, 16k, 32k, 64k, 128k, 256k\,.$$

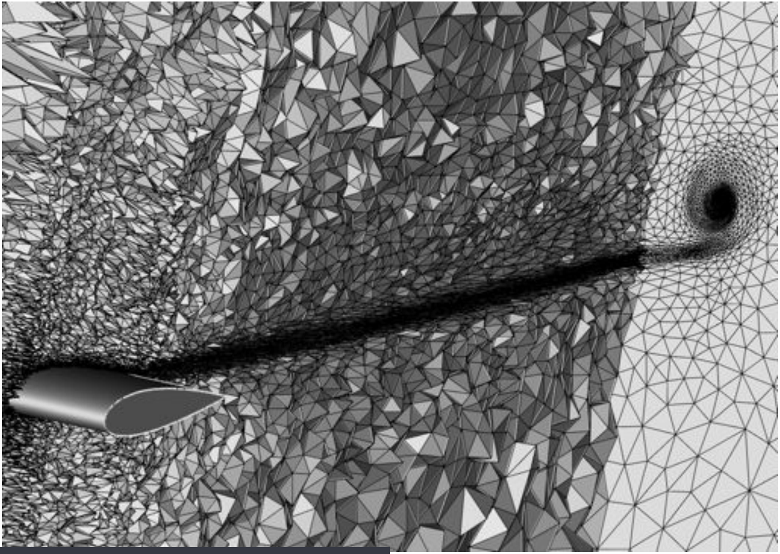
3 adaptive iterations were performed for each complexity.

- Adaptive FMG: Views of the meshes
  - Initial mesh

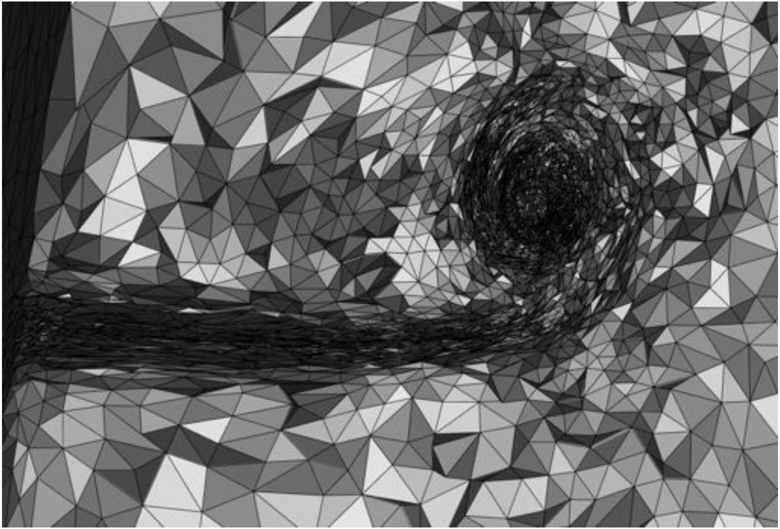


3D subsonic NACA 0012 (2/5)

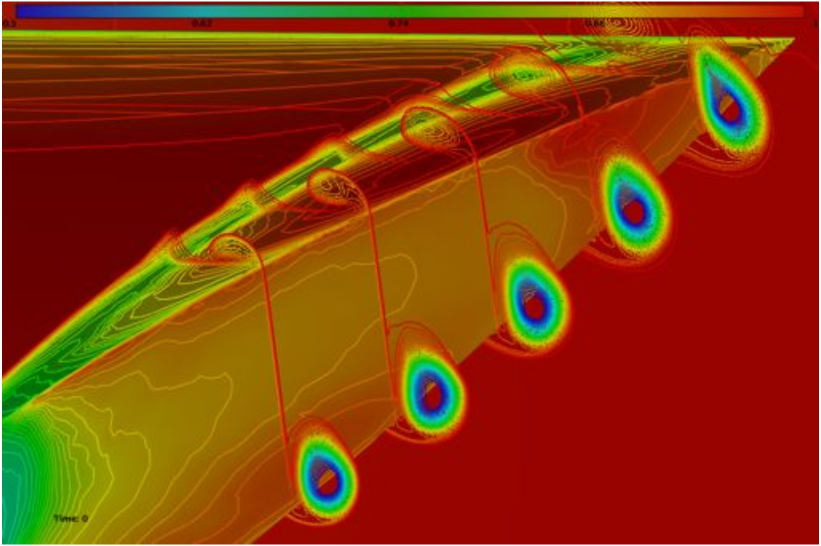
- Adaptive FMG: Views of the meshes
  - Final adapted mesh



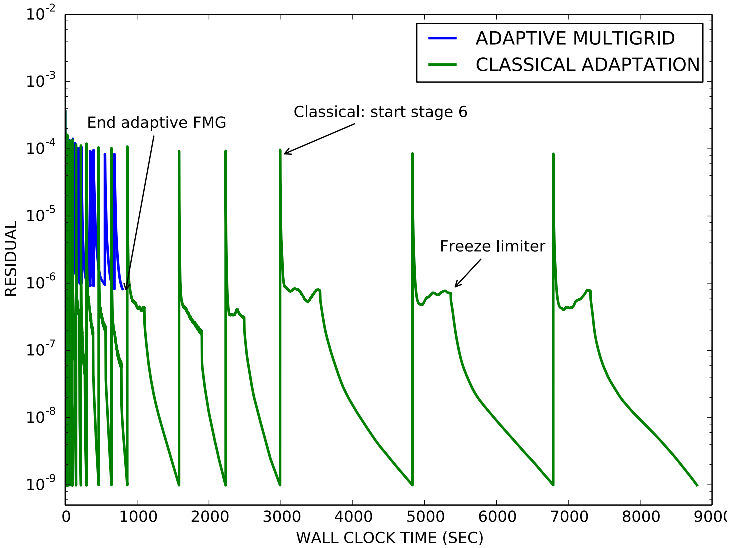
- Adaptive FMG: Views of the meshes
  - Final adapted mesh (trailing vortices region)



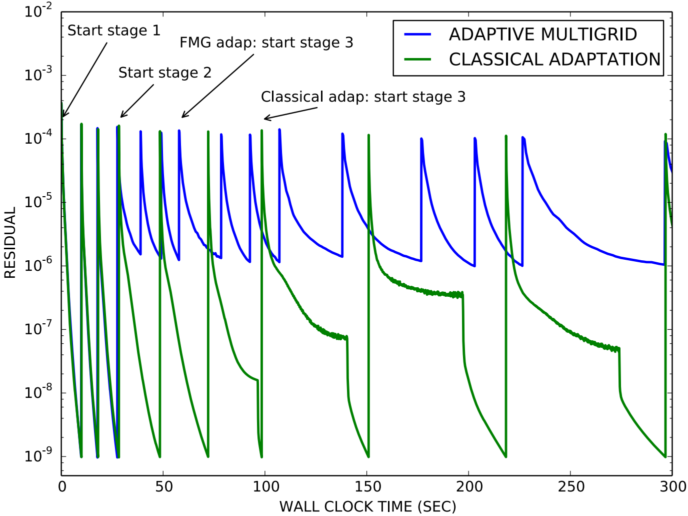
- Adaptive FMG: final solution (velocity isovalues)
    - Small flow details captured thanks to mesh adaptation



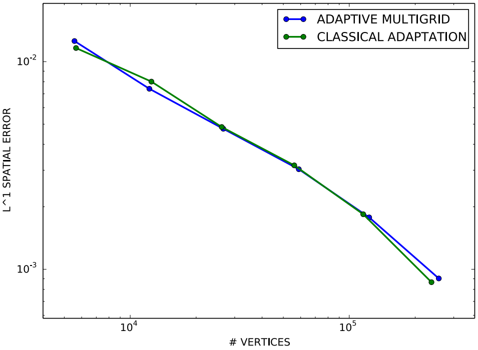
- Comparison of the residual convergence



- Comparison of the residual convergence
  - Close-up view of the first iterations



- Verification of the mesh convergence



NB: The reference couple mesh/solution used to compute the spatial error was generated using one more step in the classical mesh adaptation (with a higher mesh complexity).

3D transonic wing body tails (WBT) configuration
Mach 0.8, $\alpha = 1\,\text{deg}$.

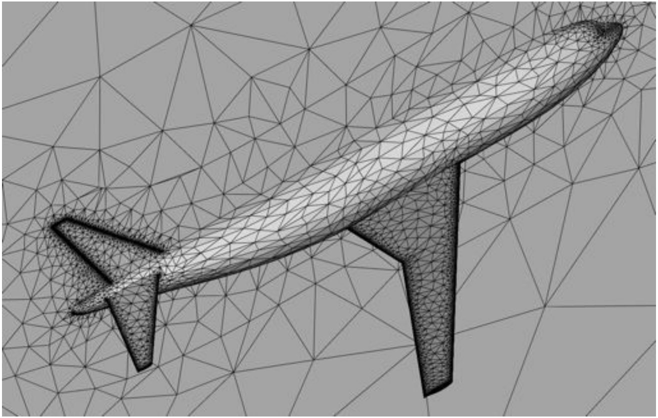
- **Two simulations** were performed:
  1. A classical mesh adaptation
     - Full convergence at each stage
  2. An adaptive FMG algorithm
     - Full convergence at stage 1, two orders of magnitude at stages 2, 3, etc.

For both simulations, prescribed mesh complexities lead to the following number of vertices:

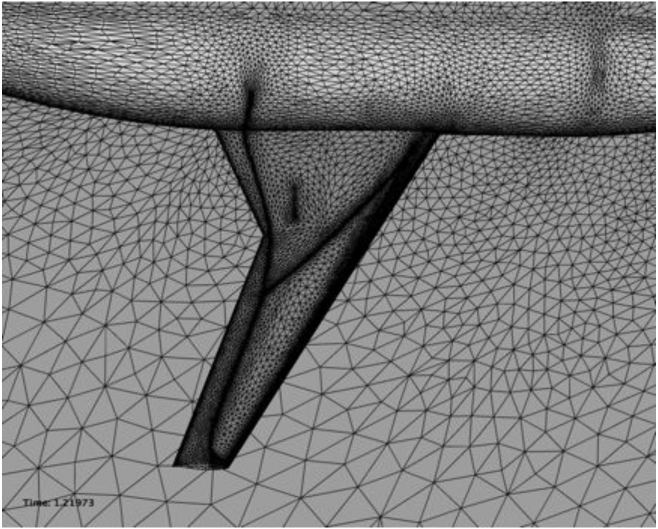
$$140k, 210k, 340k, 620k\,.$$

5 adaptive iterations were performed for each complexity.

- Adaptive FMG: Views of the meshes
  - Initial mesh (coarse, non-adapted)

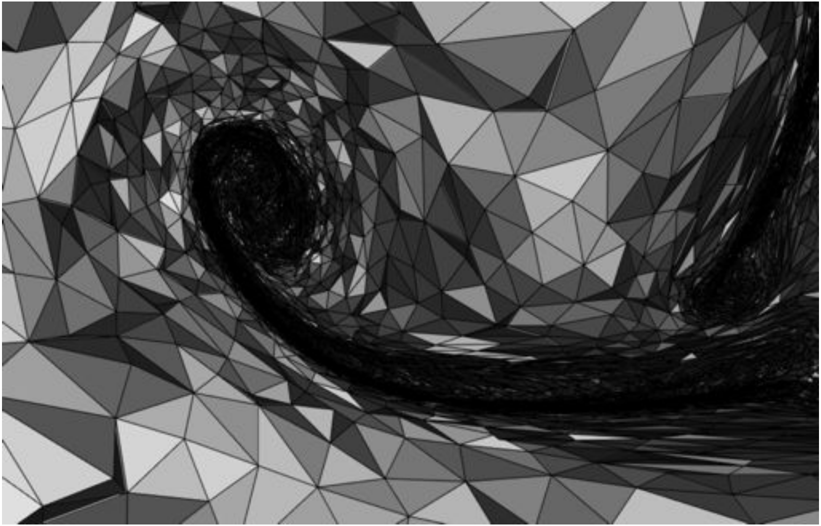


- Adaptive FMG: Views of the meshes
  - Final adapted mesh (anisotropy on the wing)



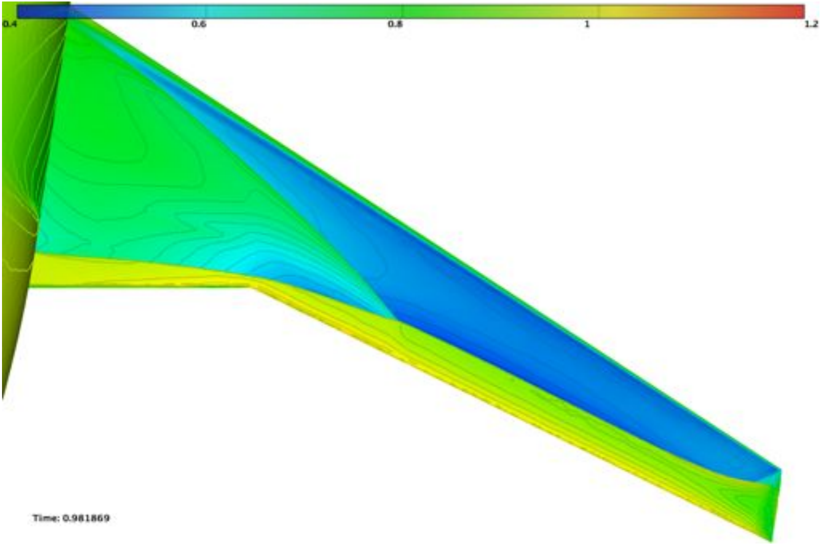
3D transonic WBT (2/5)

- Adaptive FMG: Views of the meshes
  - Final adapted mesh (trailing vortices region)

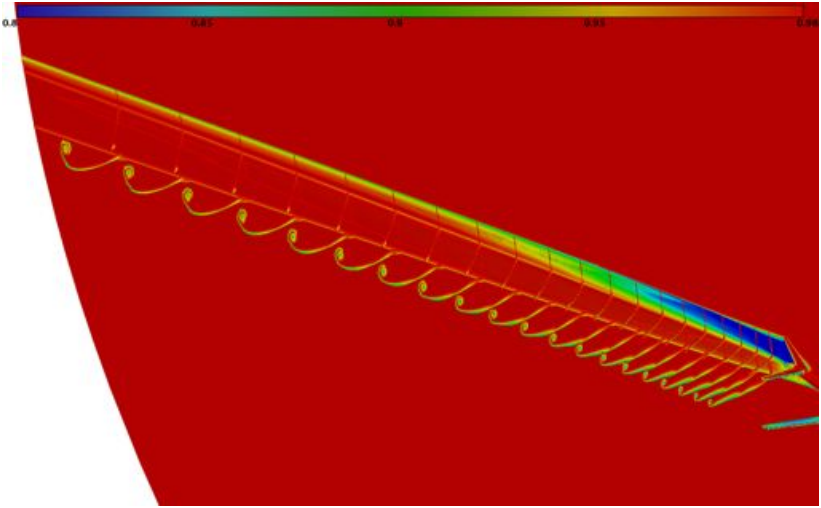


3D transonic WBT (3/5)

- Adaptive FMG: Solution (pressure on the wing)

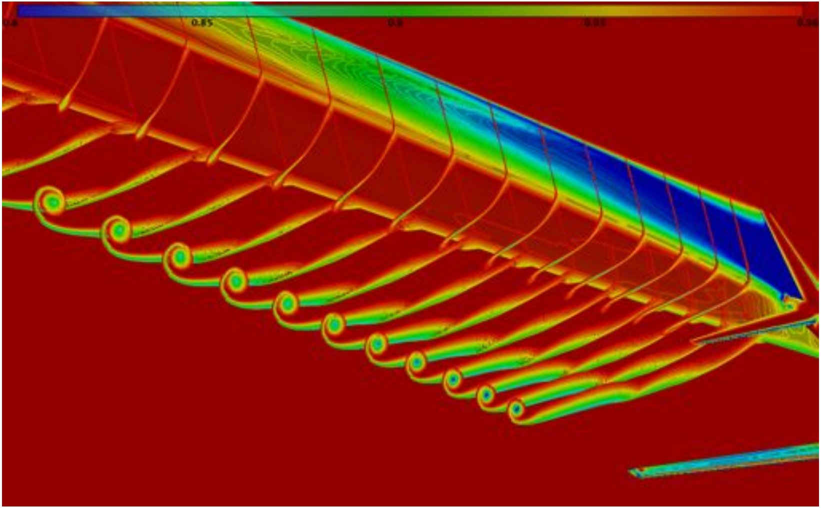


- Adaptive FMG: Solution (trailing vortices - large view)



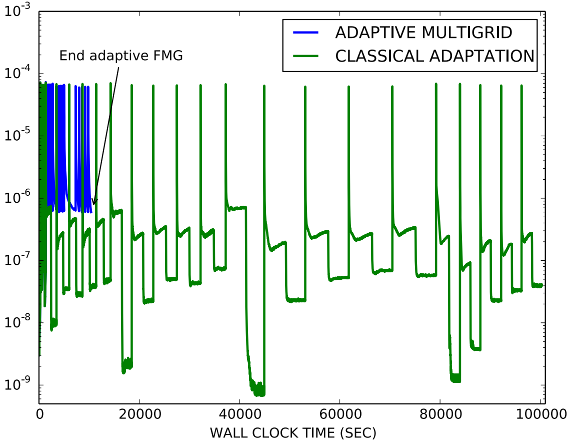
3D transonic WBT (3/5)

- Adaptive FMG: Solution (trailing vortices - close-up view)



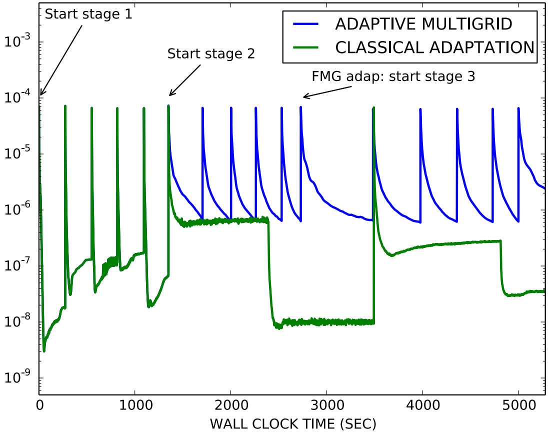
3D transonic WBT (4/5)

- Comparison of the residual convergence in terms of wall clock time
  - Wall clock time drops from 1d4h to 2h53m

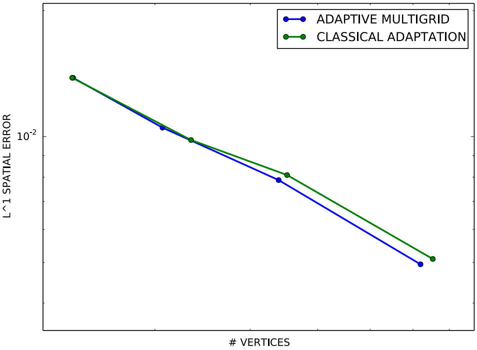


3D transonic WBT (4/5)

- Comparison of the residual convergence in terms of wall clock time (close-up view of the first iterations)
  - Wall clock time drops from 1d4h to 2h53m



- Verification of the mesh convergence



NB: The reference couple mesh/solution used to compute the spatial error was generated using one more step in the classical mesh adaptation (with a higher mesh complexity).

Conclusion/Perspectives

- Done :
  - Implementation of the full multigrid (FMG) algorithm in the non-adapted case
  - Validation study
  - Coupling with adaptivity

Results : Significant reduction of the total wall clock time of the simulation, thanks to the properties from the FMG theory.

- Perspectives :
  - RANS simulations
  - Specific error estimates for an ideal relaxation
  - Further investigation of the FMG theory in the case of adapted meshes

Thank you!