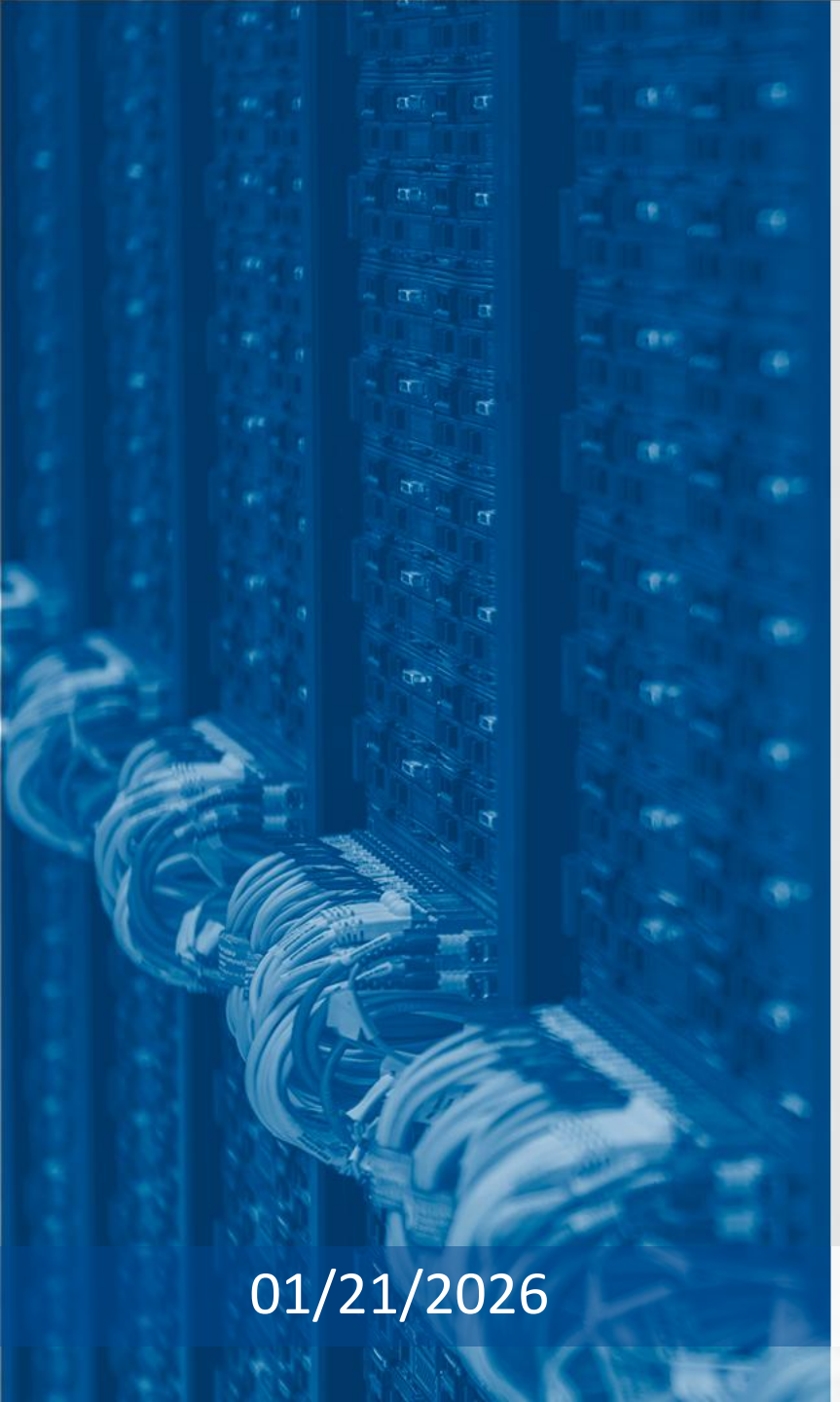


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Towards high-order low-dissipation strategies for fast industrial-scale simulations

Oriol Lehmkuhl*

*Large-scale Computational Fluid Dynamics Group Leader

oriol.lehmkuhl@bsc.es

01/21/2026

LS/CFD group at BSC

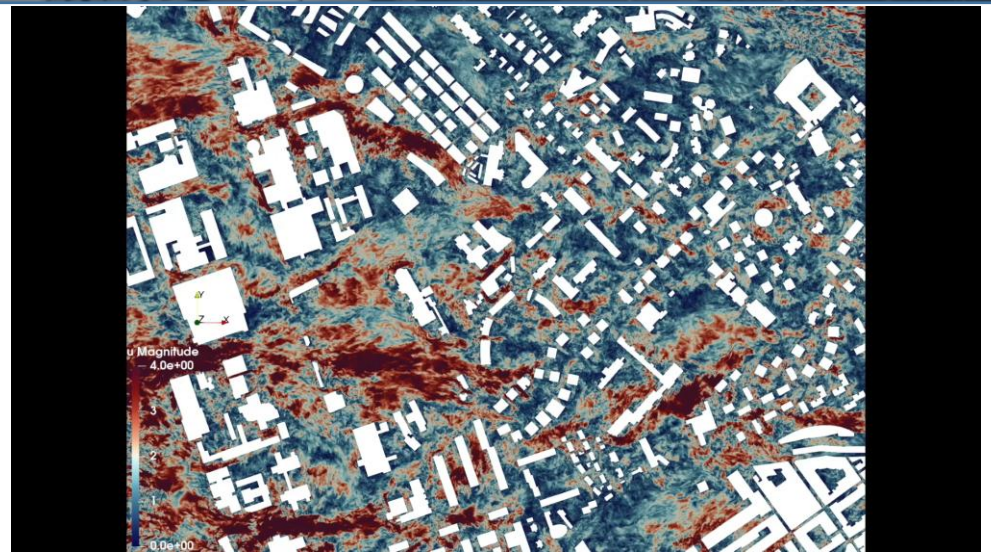
- Home
- Discover BSC
- Organisation
- Research Structure
  - High-Performance Embedded Systems (HPES)
  - Accelerators and Communications for HPC (AccelCom)
  - Atmospheric Composition
  - Best Practices for Performance and Programmability
  - Climate Variability and Change
  - Comparative Genomics

Large-scale Computational Fluid Dynamics

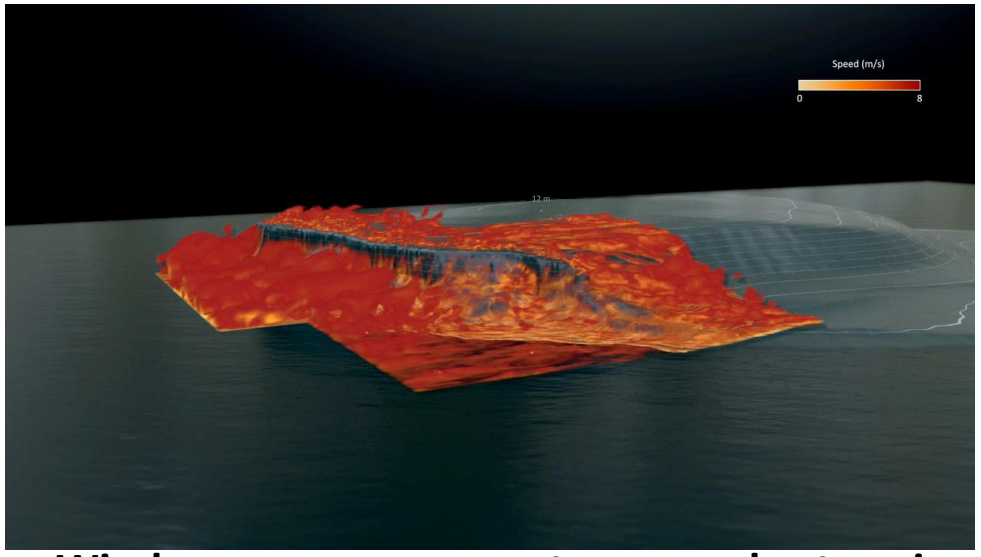
Overview | People | Research lines | Publications | Software

There is a need to increase the capabilities of current Computational Fluid Dynamics tools for engineering design by re-engineering them for extreme-scale parallel computing platforms. The backbone of **the Large-scale Computational Fluid Dynamics (LS/CFD) team** is centred on the fact that, today, **the capabilities of leading-edge emerging HPC architectures are not fully exploited by industrial simulation tools. Current state-of-the-art industrial solvers do not take sufficient advantage of the immense capabilities of new hardware architectures**, such as streaming processors or many-core platforms. A combined research effort focusing on novel numerical methods, more accurate physical models, algorithms and HPC application is the only way to make possible to develop and advance simulation tools to meet the needs of the European industry. The LS/CFD team will focus on the development of numerical tools, turbulence models, multi-physics algorithms, data driven methodologies and large-scale industrial simulations.

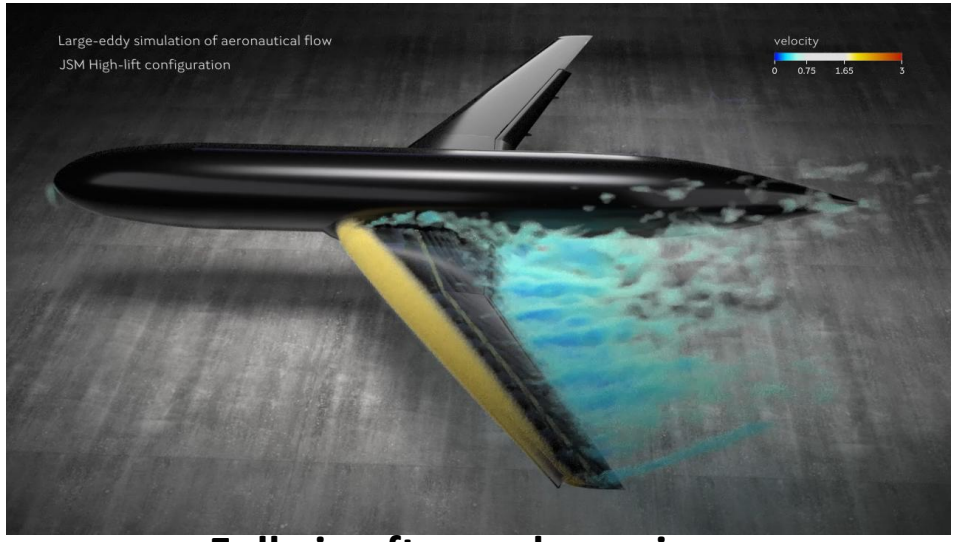
LS/CFD group research focus



Urban flows



Wind resource assesment on complex terrain

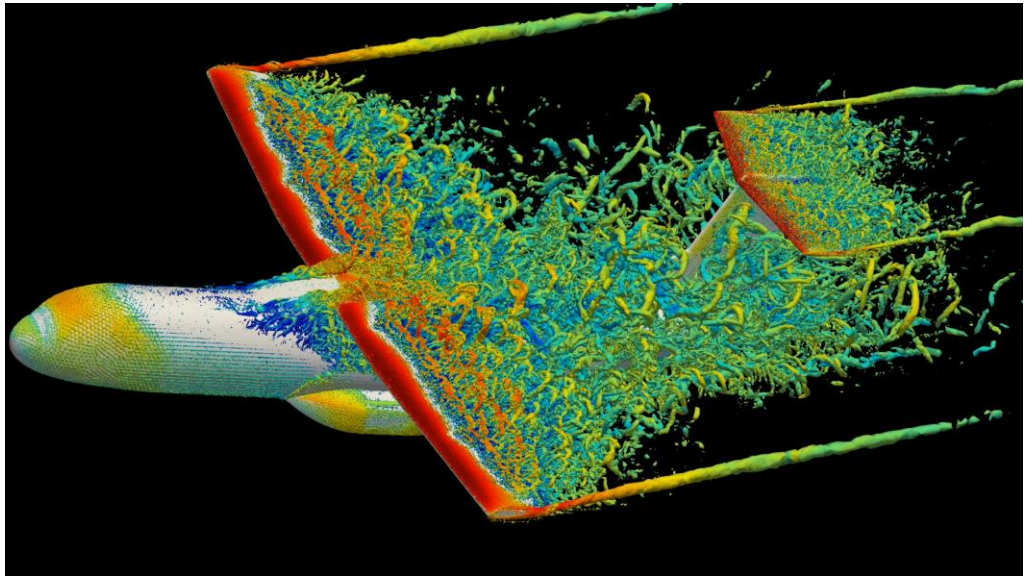


Full aircaft aerodynamics

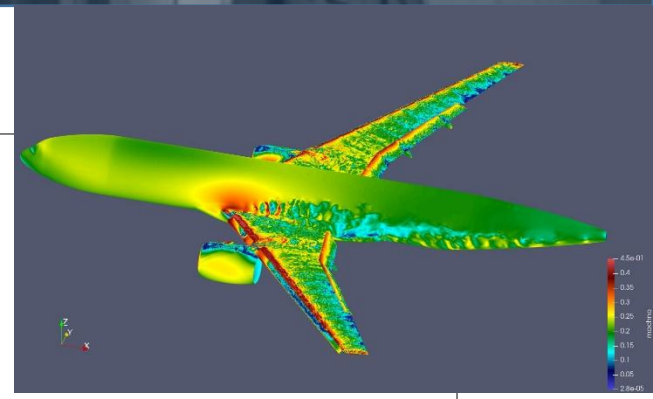
External aerodynamics of road vehicles

SOD2D: Spectral high-Order coDe 2 solve partial Differential equations

- SOD2D: **S**pectral high-**O**rder co**D**e **2** solve partial **D**ifferential equations
  - https://gitlab.com/bsc_sod2d/sod2d_gitlab
- Based on Spectral Finite Elements Method
- Simulations of turbulent compressible and incompressible flows over complex geometries
- Fully accelerated using OpenACC
- Used in aeronautical and wind energy applications
- Developed at BSC as an Open-Source in different EuroHPC projects (NextSim, CEEC, EcoE3, Excellerat 2, WinDTwin and TRANSDIFFUSE)
- 1 NVIDA H100 behaves like 560 Intel Sapphire Rapid cores
- We can run a full Aircraft in 12h using 100 H100 GPUs in the previous MN4 we needed the 70% of the machine and several days.



wmLES simulation of a new concept Aircaft from Airbus, using 32 H100 and 200M DoF.
Simulations carried out in the CDTI PTA 2023 (CETACEO) under the collaboration agreement between BSC and Airbus



Scalability of SOD2D in MN5 for the simulation of the NASA HL-CRM with 4.89Billion nodes and the compressible solver

L. Gasparino, F. Spiga, O. Lehmkuhl, SOD2D: A GPU-enabled Spectral Finite Elements Method for compressible scale-resolving simulations, Computer Physics Communications, Volume 297,
2024, 109067, ISSN 0010-4655

Large eddy simulation models: challenges and bottlenecks

By spatially filtering the NS equations:

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} - \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} + \rho^{-1} \frac{\partial \overline{p}}{\partial x_i} - F_i = -\frac{\partial \mathcal{T}_{ij}}{\partial x_j}$$

$$\mathcal{T}_{ij} - \frac{1}{3}\mathcal{T}_{kk}\delta_{ij} = -2\nu_{sgs}\overline{\mathcal{S}}_{ij}$$

- Smagorinsky
- Dynamic Smagorinsky
- Wall-Adapting Local Eddy-Viscosity (WALE) Model
- **Vreman:**

$$\nu_t = c\sqrt{\frac{B_\beta}{\alpha_{ij}\alpha_{ij}}}$$

$$\alpha_{ij} = \boldsymbol{S} = S_{ij}$$

$$\beta_{ij} = \Delta_m^2 \alpha_{mi}\alpha_{mj}$$

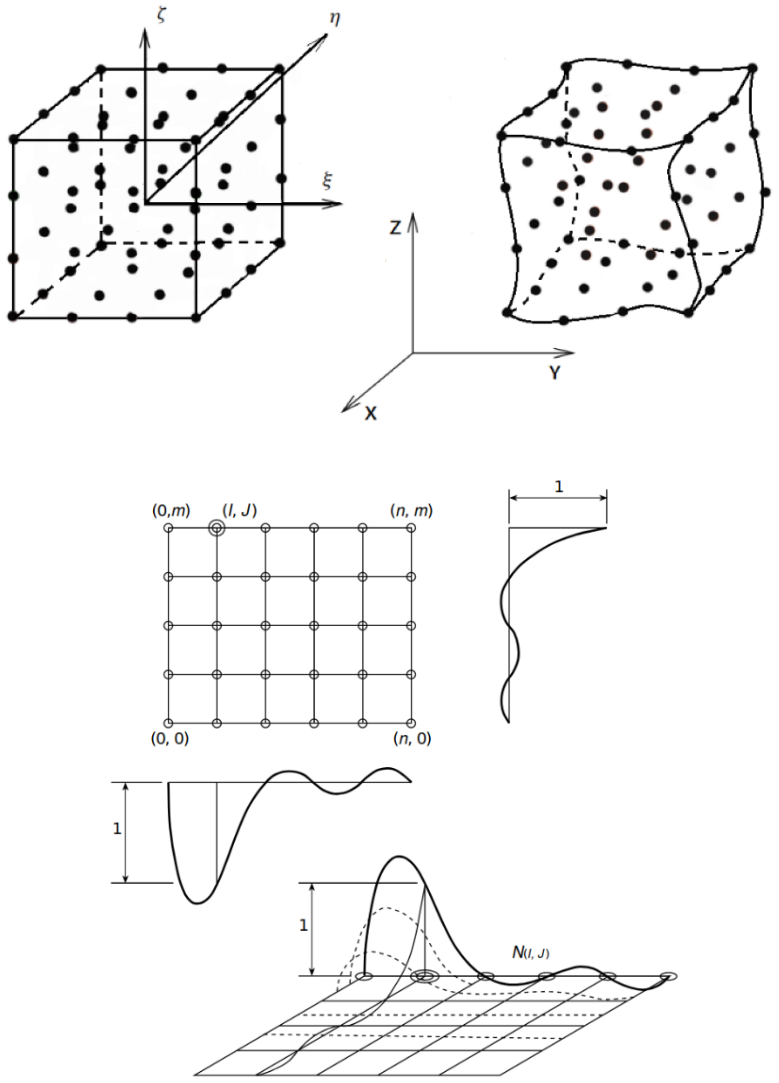
$$B_\beta = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2$$

Specific challenges:

- **Numerics interact with the LES model**
- **Usually the mesh is the filter**
- **Scales at the wall are case dependent**
- **More sensible to geometry and boundaries**

Continuous Galerkin Finite Elements model (SEM)

- **Spectral formulation** of the **Continuous Galerkin Finite Elements model (SEM)** applied to the spatial terms in the Navier-Stokes equations.
- The **Lobatto-Gauss-Legendre** (LGL) quadrature is used in the developed algorithm. (nodes are **non-equispaced**, **avoiding** the **Runge effect** on high-order interpolations)
  - The quadrature points coincide with the element nodes (***closed rule integration***) → This can lead to **aliasing effects** due to the reduced order integration of closed rule quadrature.
- Very high arithmetic intensity → **well fitted for GPUs**



Why Split Forms? (Aliasing in Collocated SEM)

- For SEM on collocated LGL nodes, convective terms suffer from aliasing errors due to under integration of nonlinear fluxes.
- Split forms (Kennedy-Gruber, Pirozzoli, Shima, Chandrashekar, etc.) reduce aliasing-driven instabilities by rewriting convective terms in alternative but mathematically equivalent forms.

$$\partial_x(ab) \approx \frac{1}{2}\partial_x(ab) + \frac{1}{2}\left(a\,\partial_x b + b\,\partial_x a\right).$$

- For collocated SEM on LGL nodes, the mass matrix $\mathcal{M} = \omega_i \delta_{ij}$ and derivative matrix $\mathcal{D} = l_j'(\xi_i)$ provide a summation by parts (SBP) operator $\mathcal{Q} = \mathcal{M}\mathcal{D}$:

$$\mathcal{Q} + \mathcal{Q}^T = \mathcal{B}, \quad \mathcal{B} = \mathrm{diag}(-1, 0, \ldots, 0, 1).$$

- For SBP operators, the application of $\mathcal{D}$ can be writen in a subcell finite volume like flux differencing form. With freedom to choose a symmetric and consistent numerical two-point flux $F^{\#}$.

Key Result: Numerical Volume Flux F # ↔ Split Forms

You can reproduce classical split formulations exactly at the discrete level using two point fluxes, $F^{\#}$. ([1])

Discrete split-form identities

Let $\{\!\{a\}\!\}_{im} = \frac{1}{2}(a_i + a_m)$ be the arithmetic mean between nodes i and m. Then the following identities hold:

$$2\sum_{m=0}^{N} D_{im}\ \{\!\{a\}\!\}_{im} \;=\; (a_x)_i$$

$$2\sum_{m=0}^{N} D_{im}\ \{\!\{a\}\!\}_{im}\ \{\!\{b\}\!\}_{im} \;=\; \frac{1}{2}\Big((ab)_x + a\,b_x + b\,a_x\Big)_i$$

$$2\sum_{m=0}^{N} D_{im}\ \{\!\{a\}\!\}_{im}\ \{\!\{b\}\!\}_{im}\ \{\!\{c\}\!\}_{im} \;=\; \frac{1}{4}\Big((abc)_x + a\,(bc)_x + b\,(ac)_x + c\,(ab)_x + bc\,a_x + ac\,b_x + ab\,c_x\Big)_i$$

- Arithmetic mean ⇒ conservative form
- Product of means ⇒ quadratic or cubic split form

Two-Point Fluxes for Compressible Flows in SOD2D

- In sod2d multiple two-point fluxes are implemented, providing kinetic energy preserving and entropy stable options.
- For example the split form of Pirozzoli (PI) ([6]) is given by:

$$F^{\#}_{\mathrm{PI}} = \begin{bmatrix} \{\!\{\rho\}\!\}\,\{\!\{u\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{u\}\!\}^2 + \{\!\{p\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{u\}\!\}\,\{\!\{v\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{u\}\!\}\,\{\!\{w\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{u\}\!\}\,\{\!\{h\}\!\} \end{bmatrix} \quad G^{\#}_{\mathrm{PI}} = \begin{bmatrix} \{\!\{\rho\}\!\}\,\{\!\{v\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{u\}\!\}\,\{\!\{v\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{v\}\!\}^2 + \{\!\{p\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{v\}\!\}\,\{\!\{w\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{v\}\!\}\,\{\!\{h\}\!\} \end{bmatrix} \quad H^{\#}_{\mathrm{PI}} = \begin{bmatrix} \{\!\{\rho\}\!\}\,\{\!\{w\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{u\}\!\}\,\{\!\{w\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{v\}\!\}\,\{\!\{w\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{w\}\!\}^2 + \{\!\{p\}\!\} \\ \{\!\{\rho\}\!\}\,\{\!\{w\}\!\}\,\{\!\{h\}\!\} \end{bmatrix}$$

- On curved meshes, the computation also involves averaging of the metric terms. For example for the ξ-direction of the divergence in reference space:

$$2\sum_{m=0}^{N} \mathcal{D}_{im}\left[\mathbf{F}^{\#}_{(i,m)jk}\,\{\!\{J\vec{a}^{1}_{1}\}\!\}_{(i,m)jk} + \mathbf{G}^{\#}_{(i,m)jk}\,\{\!\{J\vec{a}^{1}_{2}\}\!\}_{(i,m)jk} + \mathbf{H}^{\#}_{(i,m)jk}\,\{\!\{J\vec{a}^{1}_{3}\}\!\}_{(i,m)jk}\right].$$

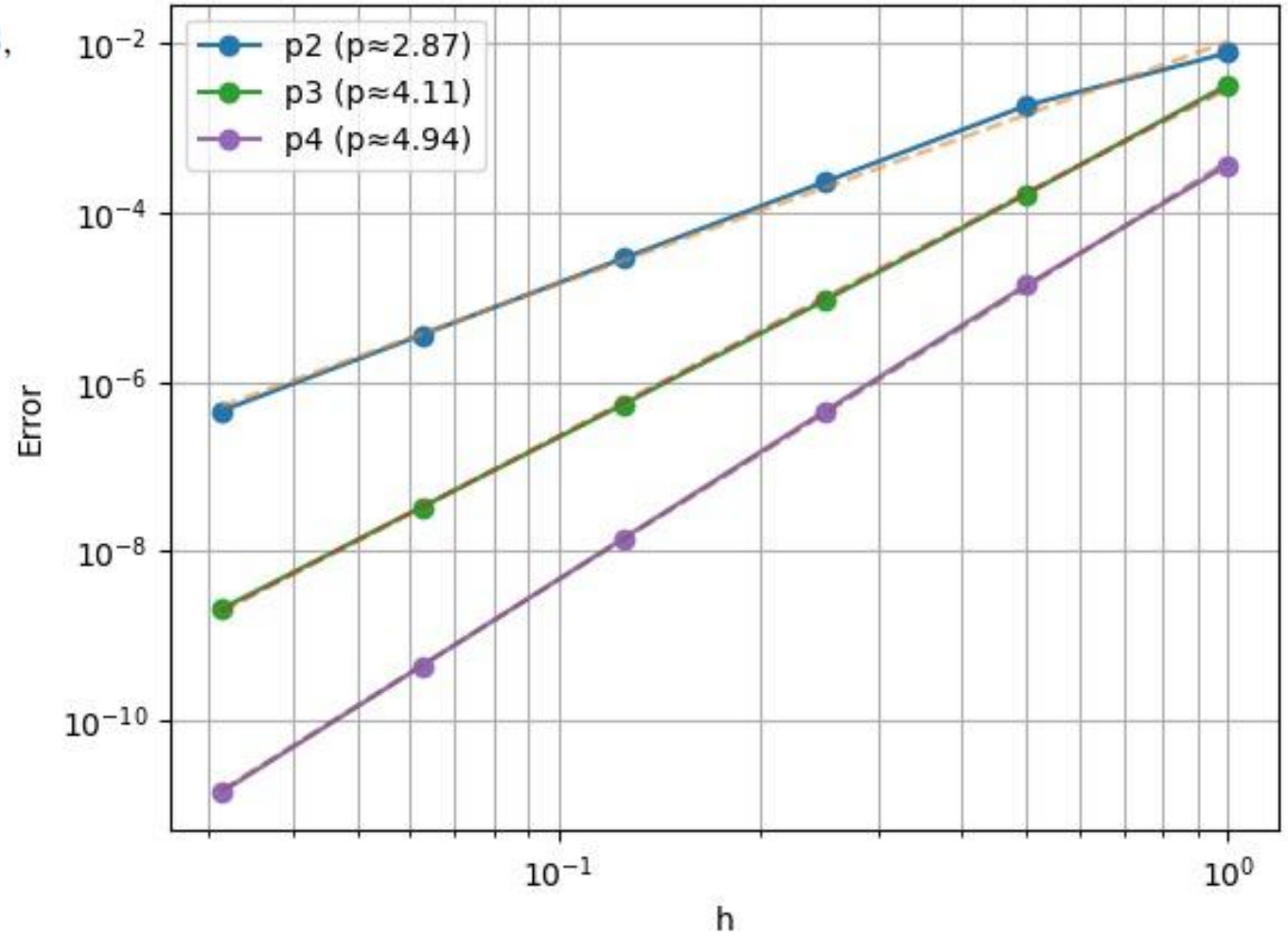
- To guarantee free-stream preservation, the metric terms are computed with the invariant form of Kopriva([5])

$$J\vec{a}^{i}_{n} = \frac{1}{2}\,\hat{x}_i \cdot \nabla_\xi \times \left(x_m \nabla_\xi x_l - x_l \nabla_\xi x_m\right), \quad (n, m, l) \text{ cyclic.} \tag{1}$$

Two-Point Fluxes for Compressible Flows in SOD2D

$$\rho\left(\vec{x},t\right) = 2 + 0.1\sin\left(\pi\left(x_1 + x_2 + x_3 - 0.6t\right)\right),$$
$$\rho u_\iota\left(\vec{x},t\right) = 2 + 0.1\sin\left(\pi\left(x_1 + x_2 + x_3 - 0.6t\right)\right), \qquad \iota = 1,2,3,$$
$$E\left(\vec{x},t\right) = \left[2 + 0.1\sin\left(\pi\left(x_1 + x_2 + x_3 - 0.6t\right)\right)\right]^2,$$

- MMS of a compressible flow
- The theoretical order of the convective operator is recovered
- Meshes:
  - 2x2x2
  - 4x4x4
  - 8x8x8
  - 16x16x16
  - 32x32x32
  - 64x64x64



Two-point flux with symmetric Lax Friedrichs blending

- To deal with discontinuities the numerical flux is written as a blend of a two point conservative flux and a symmetric Lax Friedrichs type dissipation:

$$\mathbf{f} = \mathbf{f}_{\text{cons}} + \sigma_e \, \mathbf{f}_{\text{diss}}.$$

Symmetric dissipation term.
Metrics are averaged and normalized:

$$\boldsymbol{m} = \tfrac{1}{2}\left(\{\!\{ J\vec{a}_1^1 \}\!\}_L + \{\!\{ J\vec{a}_1^1 \}\!\}_R \right), \qquad \|\boldsymbol{m}\| = \sqrt{\{\!\{ J\vec{a}_1^1 \}\!\} \cdot \{\!\{ J\vec{a}_1^1 \}\!\}}, \qquad \hat{\boldsymbol{n}} = \boldsymbol{m}/\|\boldsymbol{m}\|.$$

With $v_n = \boldsymbol{u} \cdot \hat{\boldsymbol{n}}$ and $\Delta(\cdot) = (\cdot)_R - (\cdot)_L$, the dissipative contribution is

$$\mathbf{f}_{\text{diss}} = -\alpha \begin{bmatrix} \Delta\rho \\ \bar{\rho}\,\Delta v_n\,\hat{n}_1 \\ \bar{\rho}\,\Delta v_n\,\hat{n}_2 \\ \bar{\rho}\,\Delta v_n\,\hat{n}_3 \\ \Delta p \end{bmatrix} \|\boldsymbol{m}\|, \qquad \bar{\rho} = \max\!\left(\tfrac{1}{2}(\rho_L + \rho_R), \rho_{\min}\right).$$

The dissipation strength is

$$\alpha = \tfrac{1}{2}\, s_{\max}\, \operatorname{signl}\theta, \qquad s_{\max} = \max(\bar{c}, |\bar{v}_n|) + c_{\text{fix}} \max(c_L, c_R), \qquad \bar{c} = \tfrac{1}{2}(c_L + c_R).$$

Here $\theta \in [0, 1]$ is a local symmetric entropy based limiter already embedded in iso parametric axis.

Local symmetric entropy-based limiter

The dissipation amplitude is locally modulated by a symmetric entropy based limiter

$$\theta \in [0, 1],$$

constructed from jumps of the discrete entropy flux across the iso parametric line.
We define the physical entropy and entropy flux as

$$s = \log\left(\frac{p}{\rho^\gamma}\right), \qquad \eta = -\rho s\, u_n,$$

with $u_n = \boldsymbol{u} \cdot \boldsymbol{n}$.
The limiter is obtained from the entropy flux jump

$$\Delta\eta = |\eta_R - \eta_L|,$$

and normalized as

$$\delta_\eta = \frac{\max(0,\, \Delta\eta - \varepsilon_\eta)}{|\eta_R| + |\eta_L| + \varepsilon},$$

leading to the final limiter

$$\theta = \min\left(1,\, \delta_\eta^\beta\right), \qquad \beta > 1.$$

Global elemental blending sensor

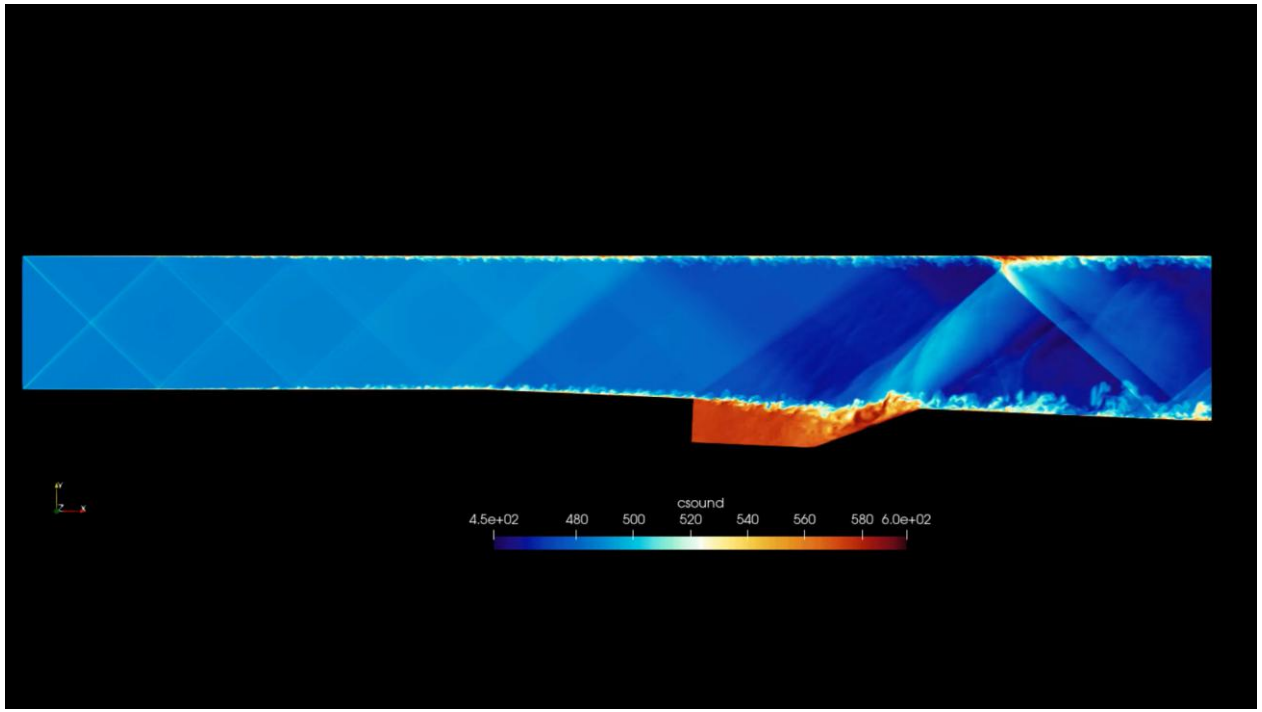
Element constant blending sensor. (Ducros based)
The blending coefficient σ_e is constant in each element and is obtained from a compression indicator built from the local divergence and vorticity:

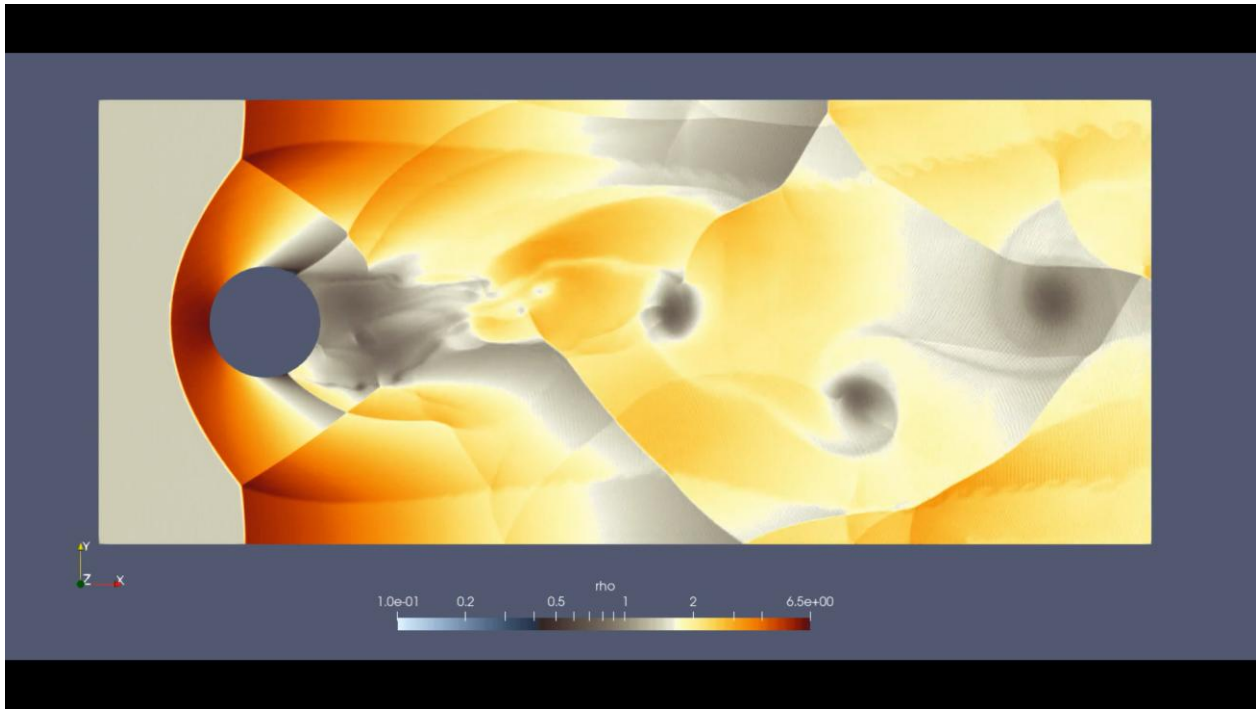
$$\phi = \min\left(\frac{\frac{4}{3}(\nabla\cdot\boldsymbol{u})^2}{(\nabla\cdot\boldsymbol{u})^2 + \|\boldsymbol{\omega}\|^2 + \varepsilon}, 1 \right), \qquad \phi_2 = \frac{(\nabla\cdot\boldsymbol{u})^2}{(\nabla\cdot\boldsymbol{u})^2 + \left(\frac{|\boldsymbol{u}|}{h}\right)^2 + \varepsilon}.$$

A nodal quantity $\phi\,\phi_2$ is accumulated and then averaged with quadrature weights inside each element:

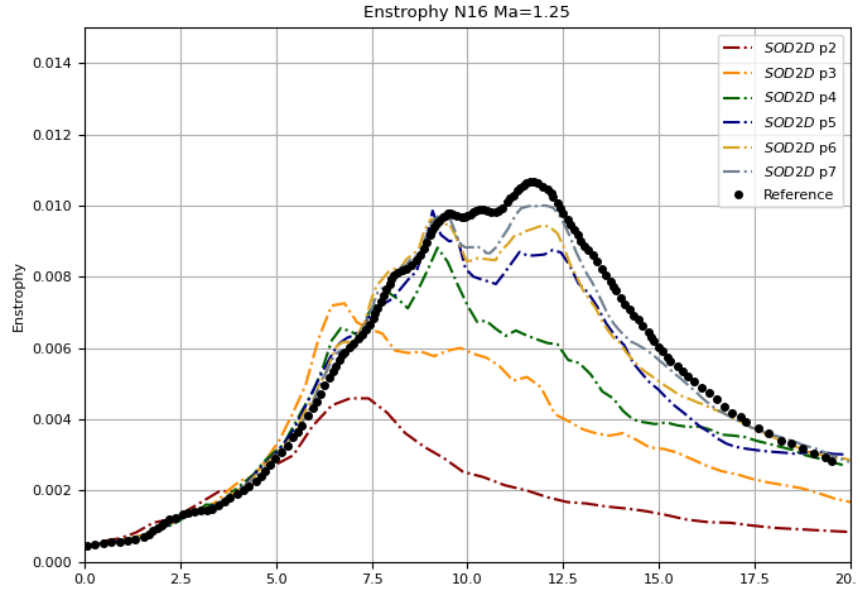
$$\sigma_e = \frac{\sum_{i\in e} w_i\,(\phi\,\phi_2)_i}{\sum_{i\in e} w_i}.$$

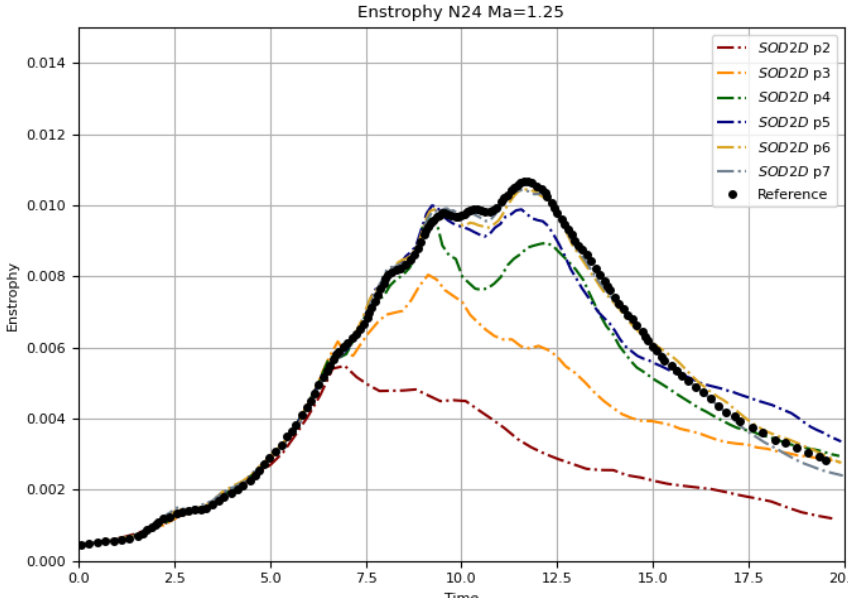
All togather :

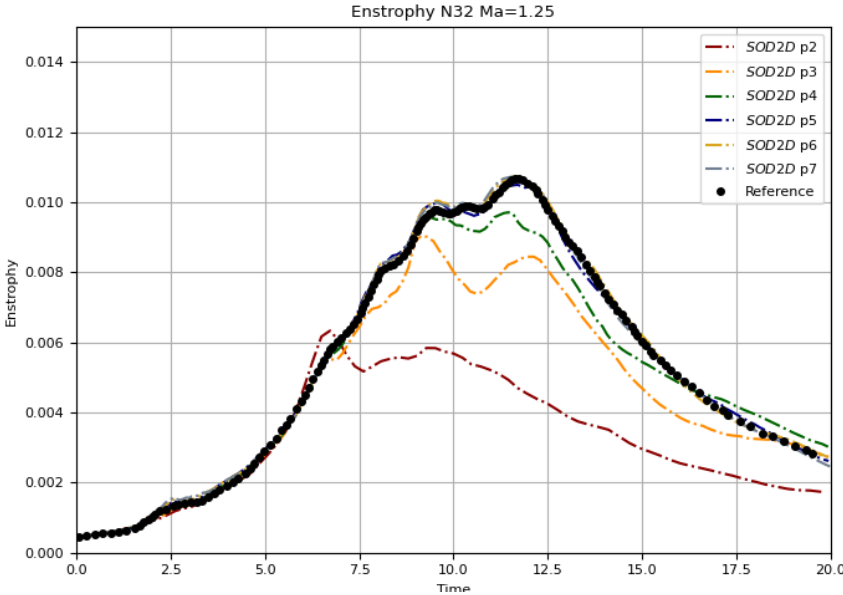


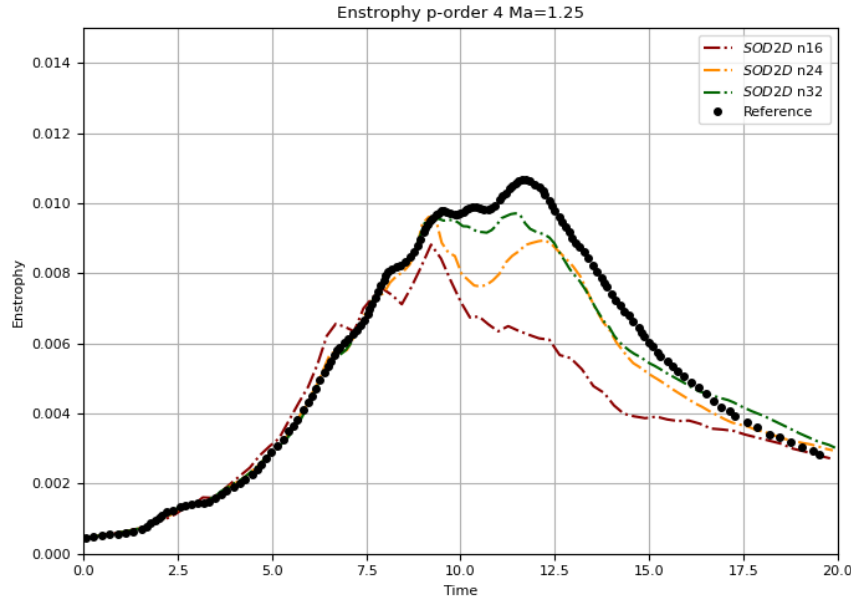


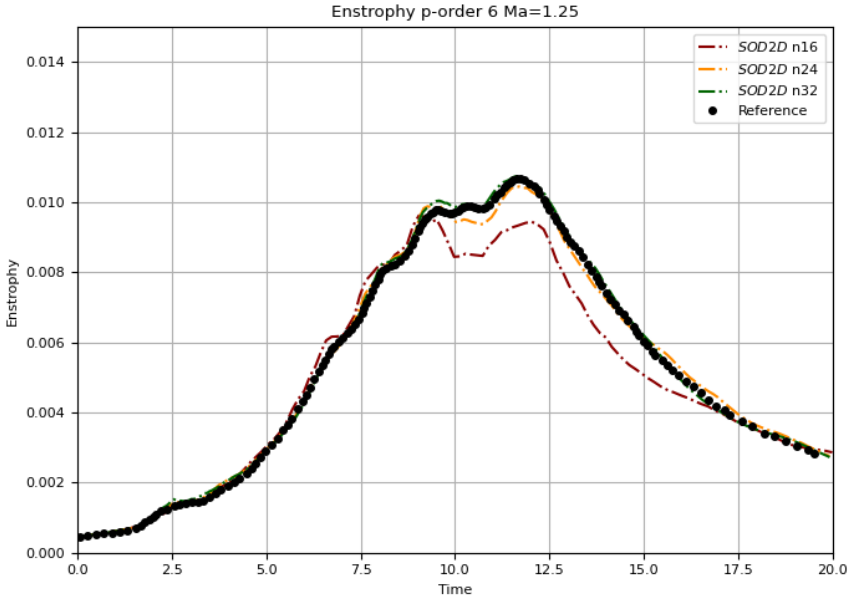
M 1.5 TGV

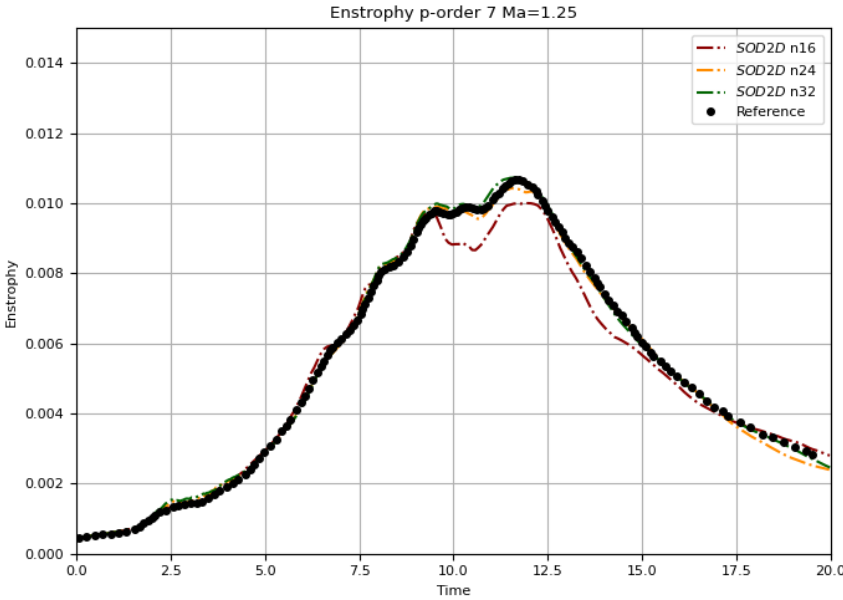

Enstrophy N16 Ma=1.25
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference
Enstrophy


Enstrophy N24 Ma=1.25
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference
Enstrophy
Time

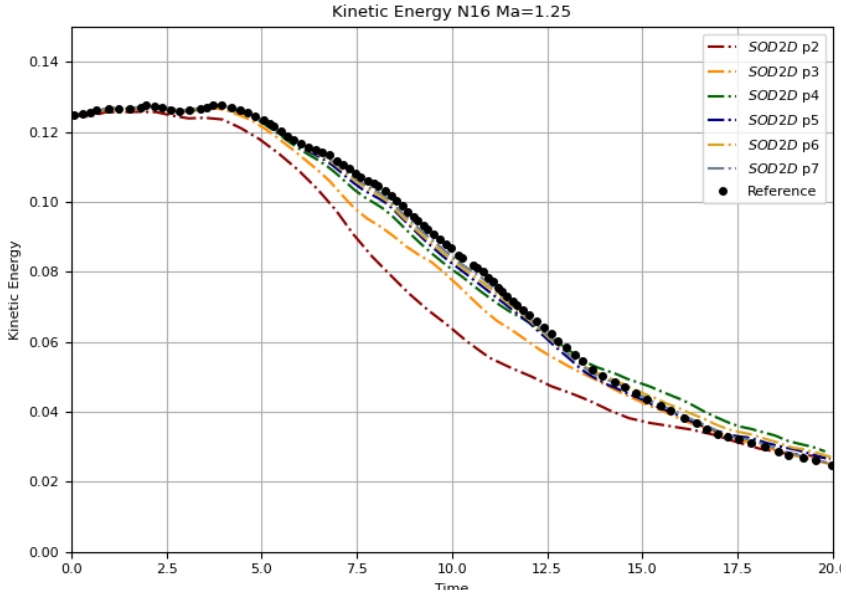

Enstrophy N32 Ma=1.25
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference
Enstrophy
Time

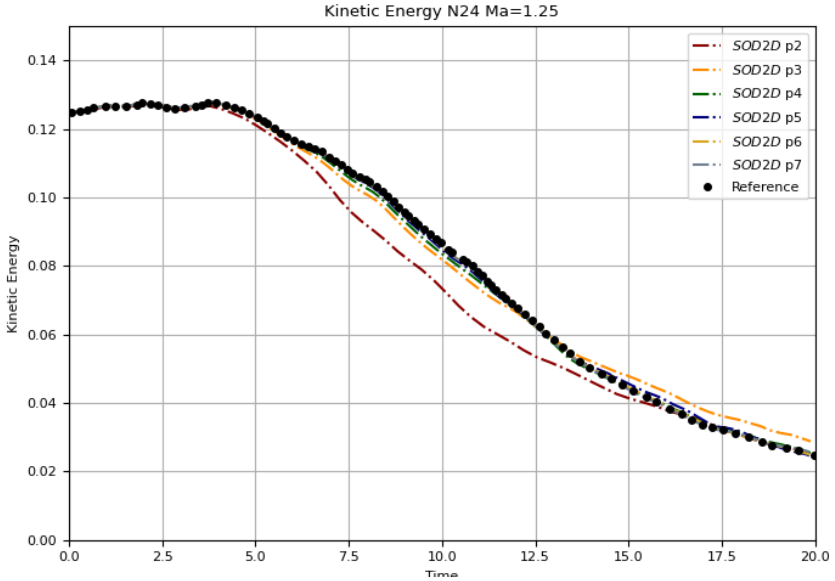

Enstrophy p-order 4 Ma=1.25
SOD2D n16
SOD2D n24
SOD2D n32
Reference
Enstrophy
Time

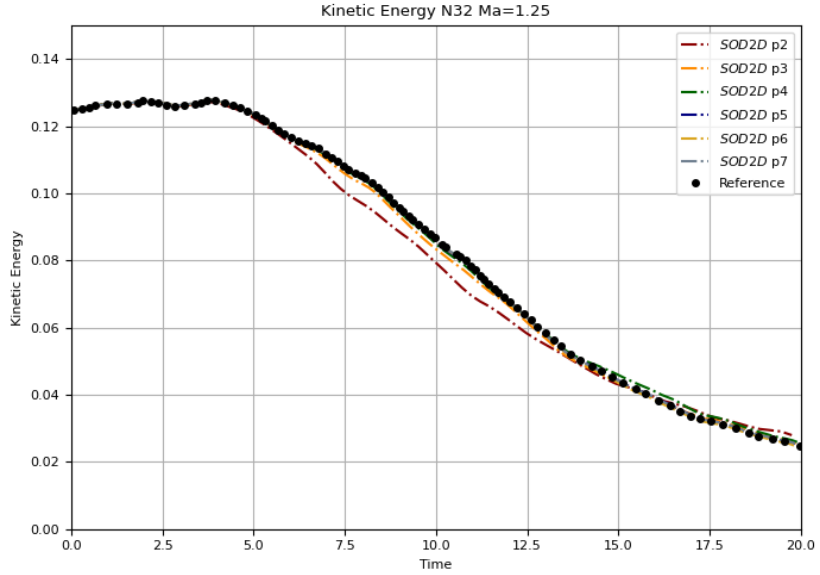

Enstrophy p-order 6 Ma=1.25
SOD2D n16
SOD2D n24
SOD2D n32
Reference
Enstrophy
Time

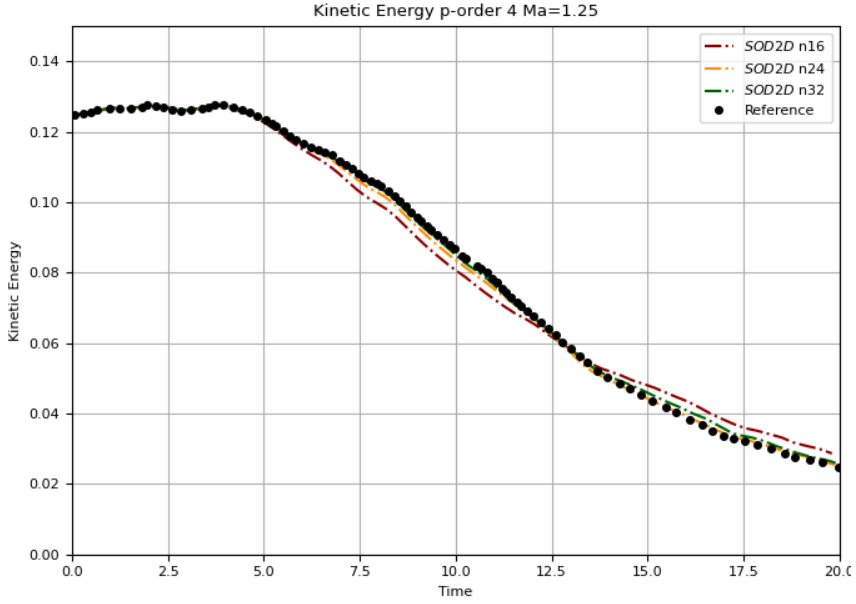

Enstrophy p-order 7 Ma=1.25
SOD2D n16
SOD2D n24
SOD2D n32
Reference
Enstrophy
Time

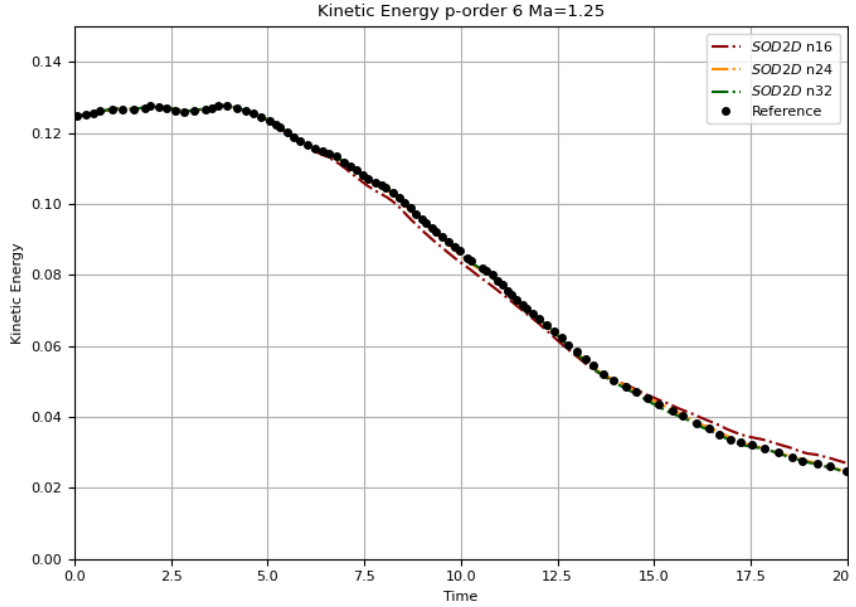
M 1.5 TGV

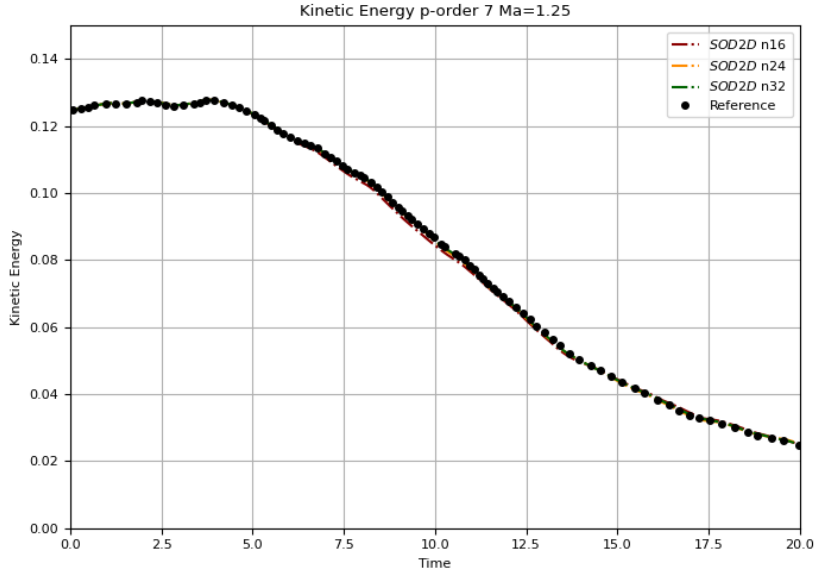

Kinetic Energy N16 Ma=1.25
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference
Kinetic Energy
Time


Kinetic Energy N24 Ma=1.25
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference
Kinetic Energy
Time

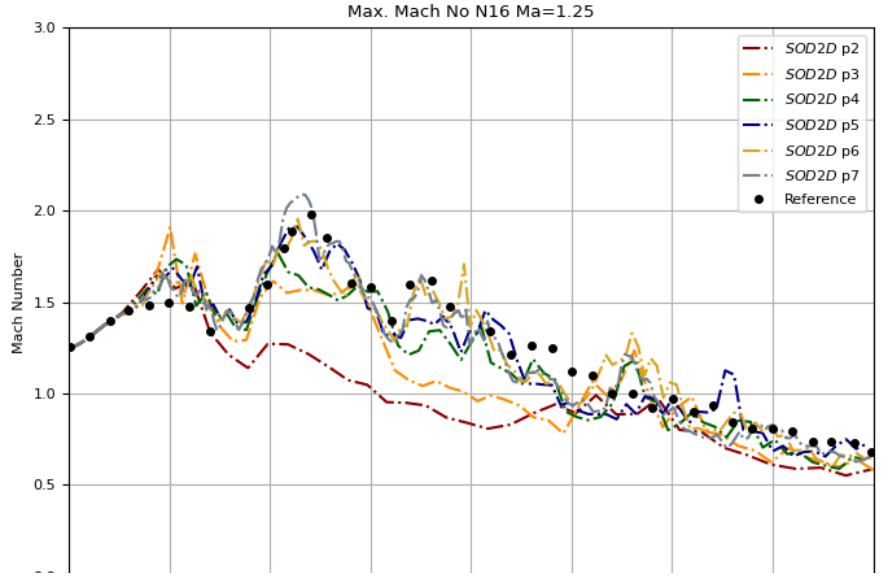

Kinetic Energy N32 Ma=1.25
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference
Kinetic Energy
Time

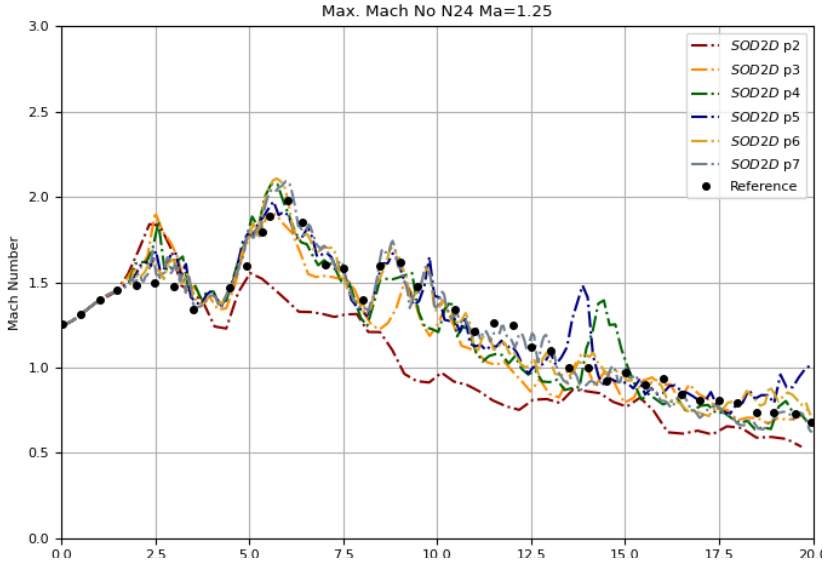

Kinetic Energy p-order 4 Ma=1.25
SOD2D n16
SOD2D n24
SOD2D n32
Reference
Kinetic Energy
Time

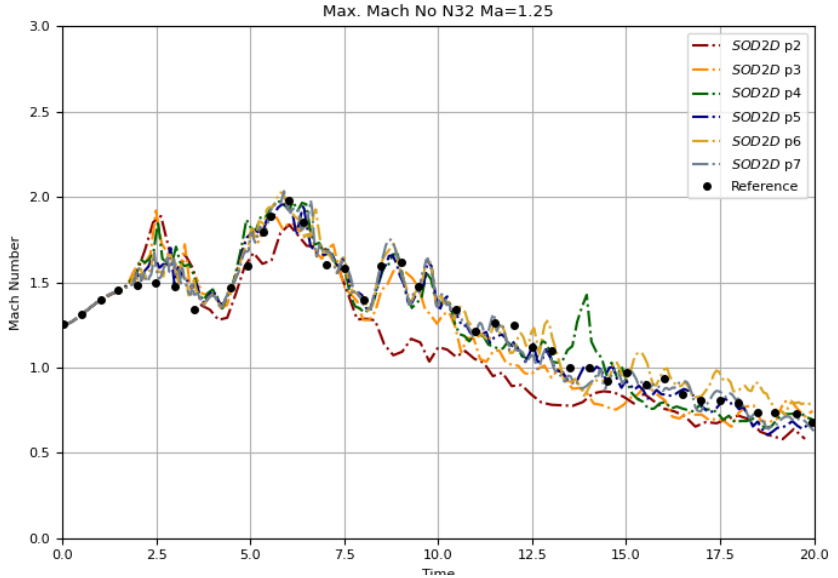

Kinetic Energy p-order 6 Ma=1.25
SOD2D n16
SOD2D n24
SOD2D n32
Reference
Kinetic Energy
Time

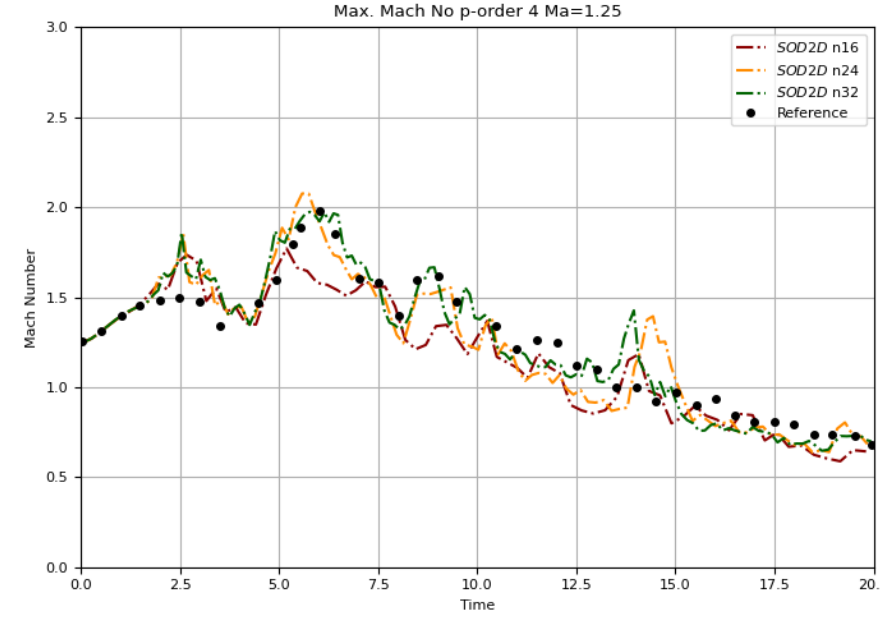

Kinetic Energy p-order 7 Ma=1.25
SOD2D n16
SOD2D n24
SOD2D n32
Reference
Kinetic Energy
Time

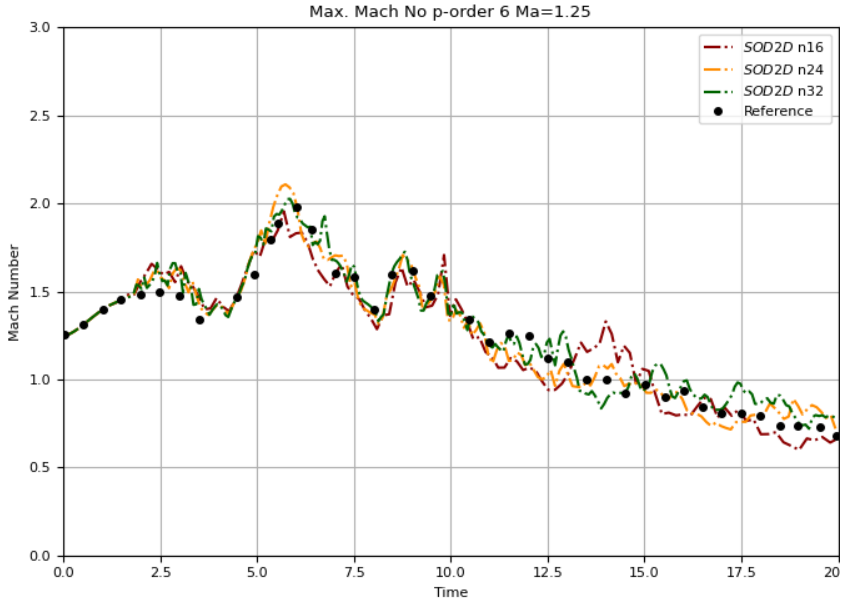
M 1.5 TGV

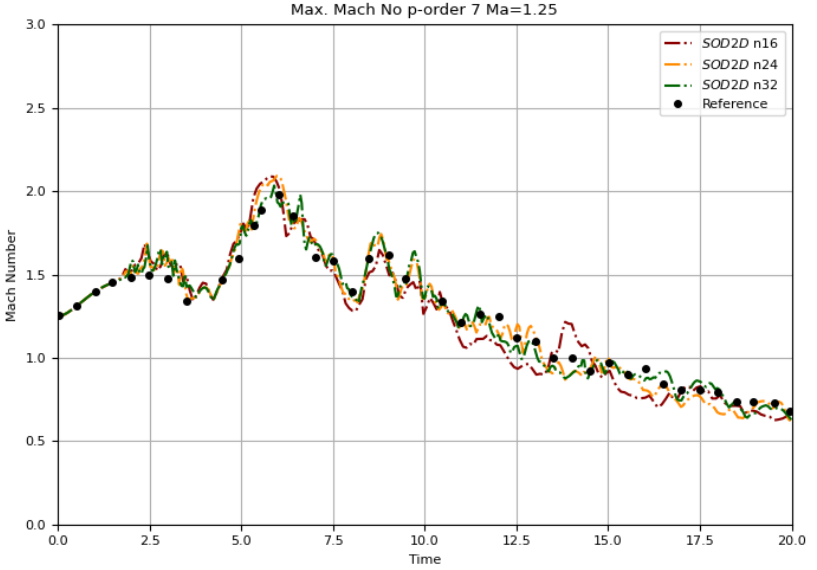

Max. Mach No N16 Ma=1.25
Mach Number
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference


Max. Mach No N24 Ma=1.25
Mach Number
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference

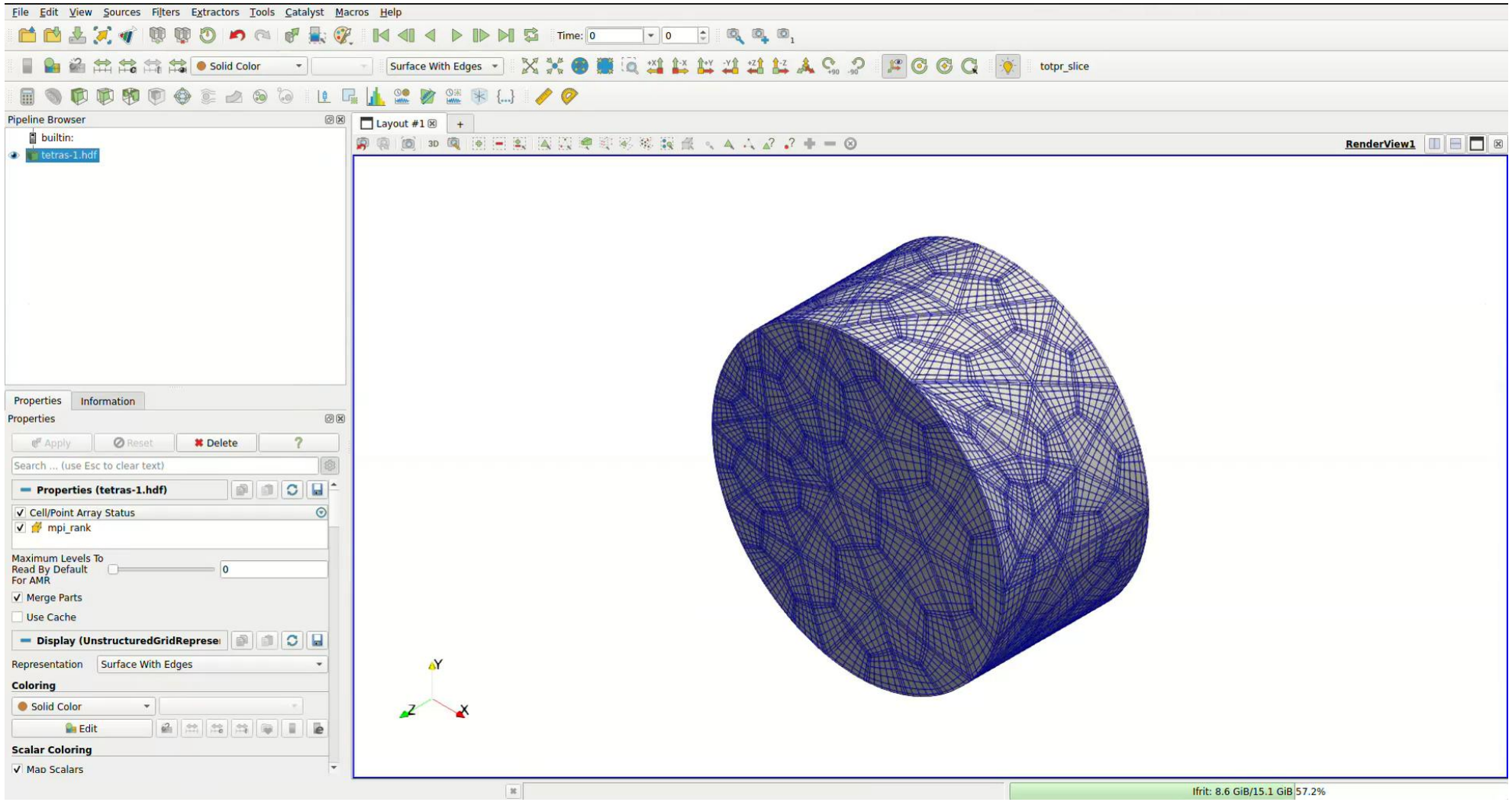

Max. Mach No N32 Ma=1.25
Mach Number
Time
SOD2D p2
SOD2D p3
SOD2D p4
SOD2D p5
SOD2D p6
SOD2D p7
Reference


Max. Mach No p-order 4 Ma=1.25
Mach Number
Time
SOD2D n16
SOD2D n24
SOD2D n32
Reference

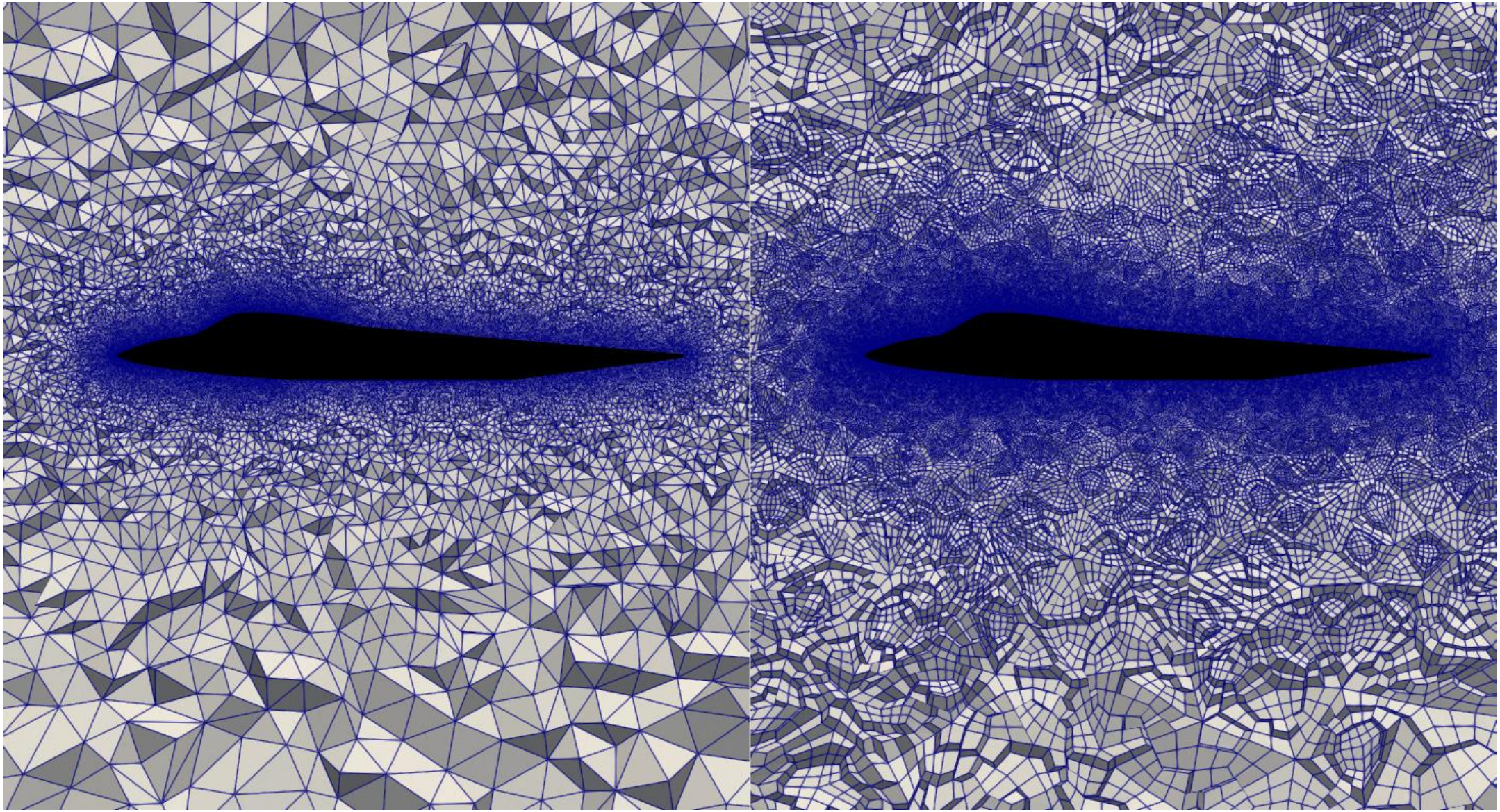

Max. Mach No p-order 6 Ma=1.25
Mach Number
Time
SOD2D n16
SOD2D n24
SOD2D n32
Reference


Max. Mach No p-order 7 Ma=1.25
Mach Number
Time
SOD2D n16
SOD2D n24
SOD2D n32
Reference

How we can go to full unstructured meshes



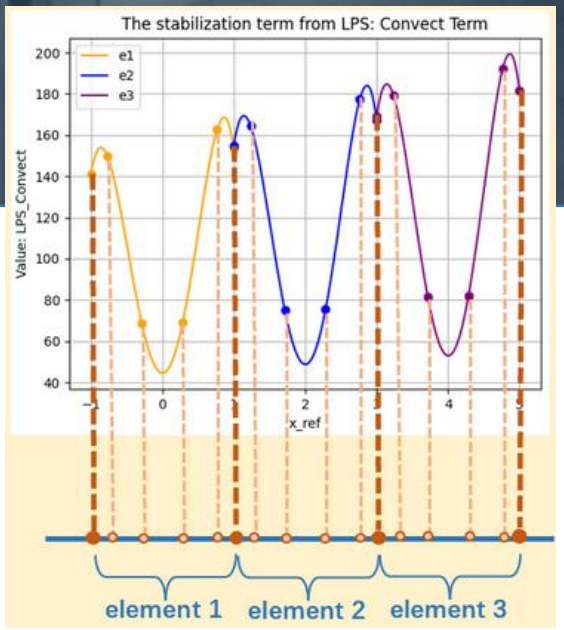
How we can go to full unstructured meshes



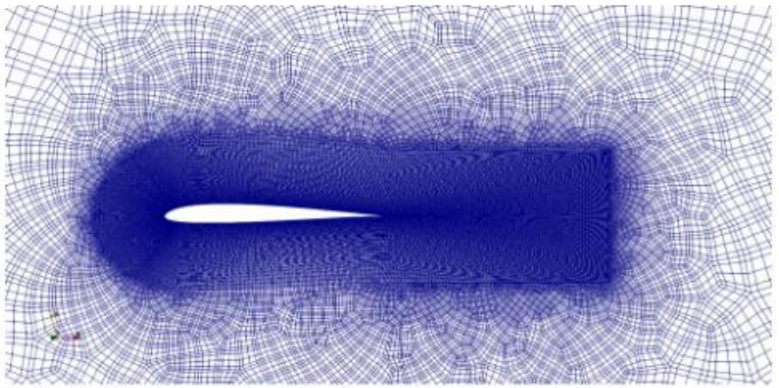
Local Projection Stabilisation

$$s_h^{e,\mathrm{LPS}} = c_s \nu^{e,\mathrm{FU}} \int_{K_e} \nabla w \cdot (g_h - \nabla u_h)\, d\mathbf{x}, \qquad \nu^{e,\mathrm{FU}} = \frac{h^e ||u||}{2p}$$

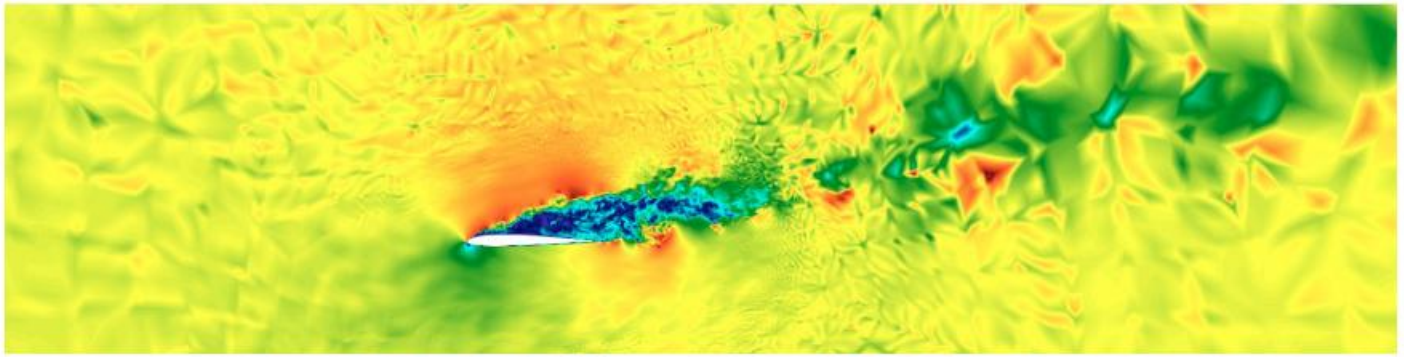
$c_s = 1$



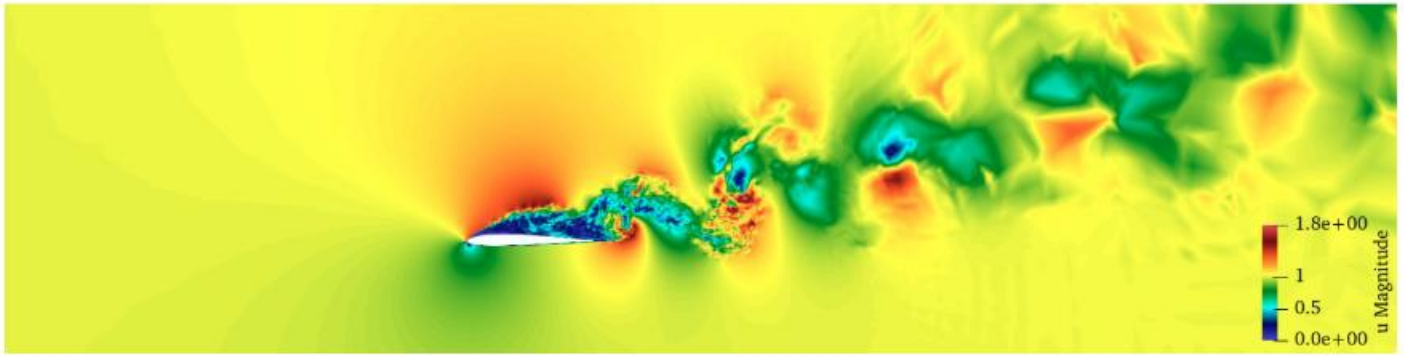
P4 spectral elements



Galerkin

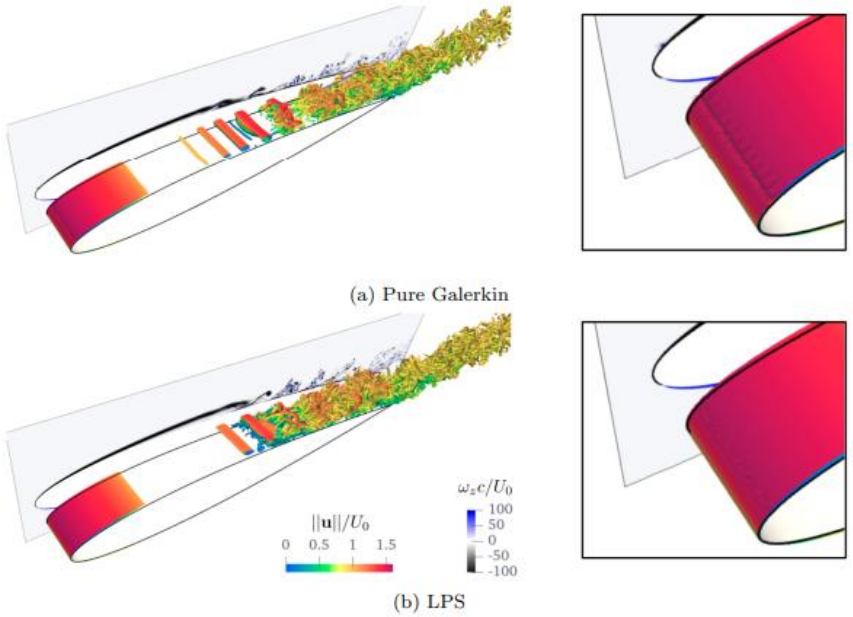


LPS



Dmitri Kuzmin and Joshua Vedral. Dissipation-based weno stabilization of high-order finite element methods for scalar conservation laws. Journal of Computational Physics, 2023

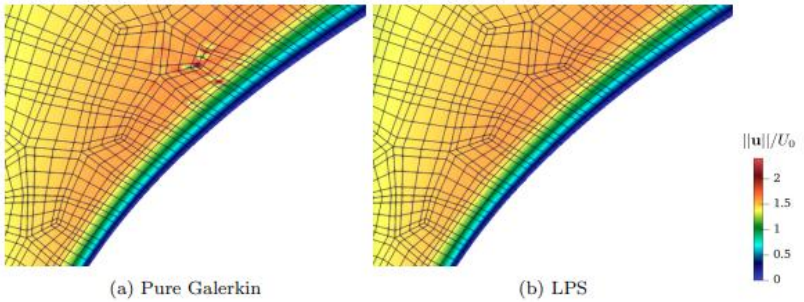
Local Projection Stabilisation



(a) Pure Galerkin

(b) LPS

Figure 9: Q-vorticiy isocontour colored by the normalized velocity magnitude. The plane shows the normalized vorticity at the mid plane in spanwise direction.

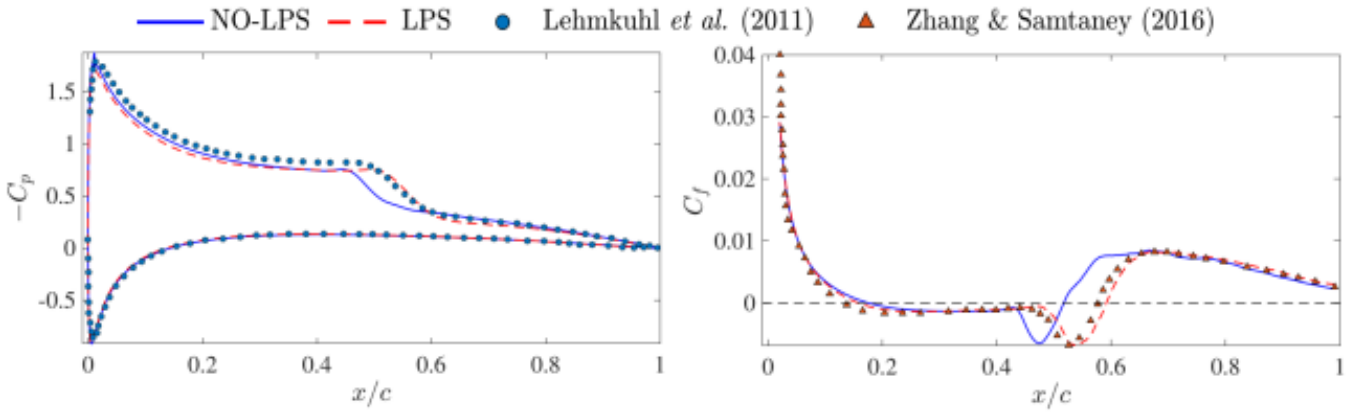


(a) Pure Galerkin (b) LPS

Figure 8: Velocity magnitude at the leading edge of the airfoil. Numerical instabilities are observed in the Pure Galerkin case.

Case	C_l	C_d	x_{sep}	x_{rea}
Lehmkuhl et al. [27]	0.569	0.0291	0.169	0.566
Zhang and Samtaney [26]	0.568	0.0285	0.141	0.580
Present, Pure Galerkin	0.558	0.0241	0.176	0.517
Present, LPS	0.558	0.0271	0.158	0.586

Table 1: Comparison of lift/drag coefficients and separation/reattachment distance in the streamline direction.

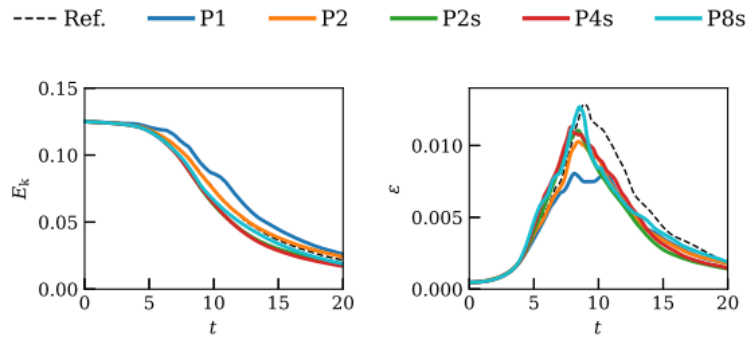


(a) Pressure coefficient (b) Friction coefficient

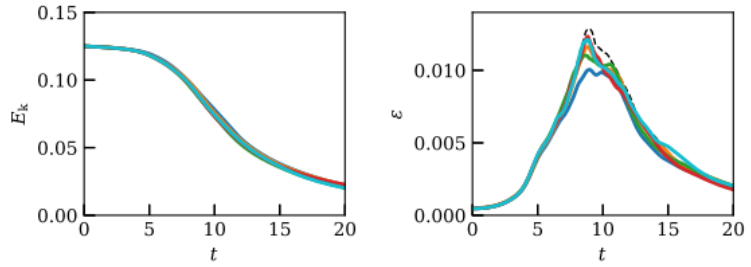
Figure 10: Distributions of the mean pressure coefficient and skin friction coefficient around the wall, plotted on the streamline direction normalized by the chord length.

Local Projection Stabilisation (M 0.1 TGV)

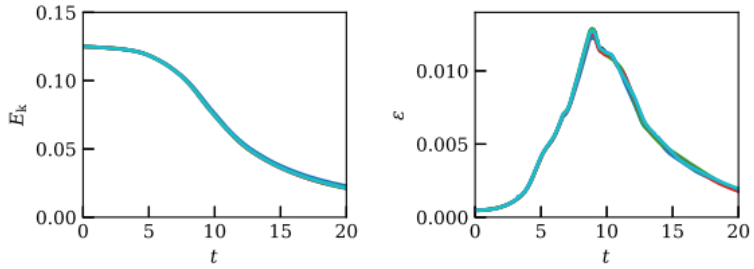
No LPS



(a) 64^3 DoFs

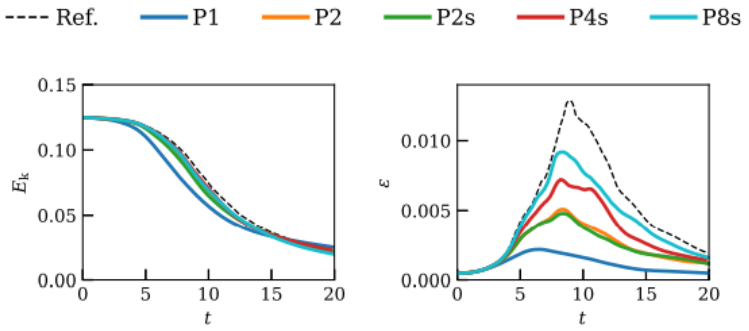


(b) 128^3 DoFs

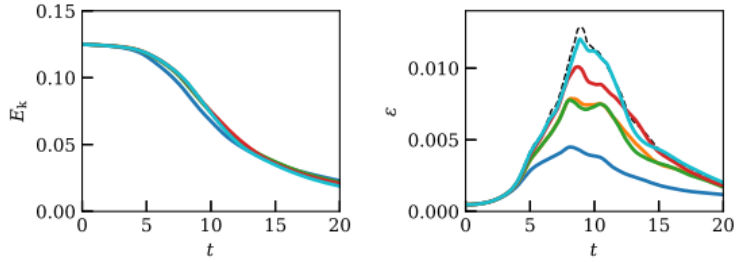


(c) 256^3 DoFs

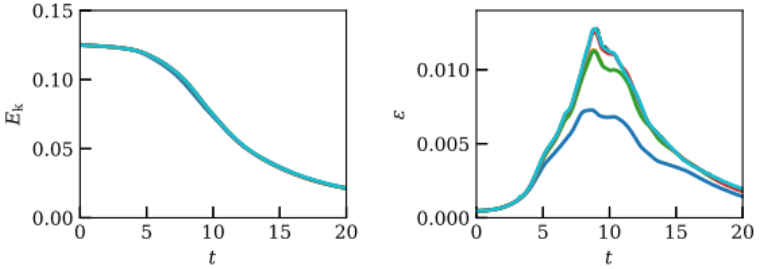
LPS



(a) 64^3 DoFs



(b) 128^3 DoFs



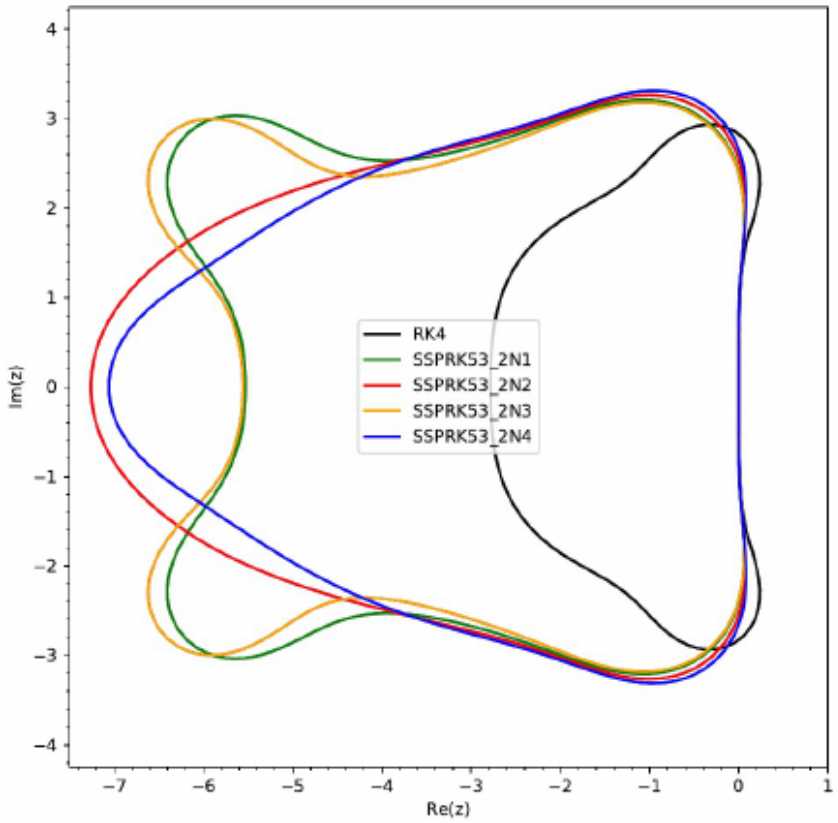
(c) 256^3 DoFs

Explicit Time Integration for Compressible Flows

- Time integration uses strong-stability-preserving, low-storage Runge-Kutta methods of the family $SSP53\_2N_n^*$. ([2][3])
- 5 stage, third order accurate schemes with low storage 2 register implementation. And larger stability regions than classical RK4 scheme.
- The scheme reads

$$
\begin{aligned}
Y_1 &= y_n, \\
Y_2 &= Y_1 + \gamma_{21}\, h\, f(Y_1), \\
Y_{i+1} &= \lambda_{i+1,1}\, Y_1 + (1 - \lambda_{i+1,1})\, Y_i + \gamma_{i+1,i}\, h\, f(Y_i), \\
y_{n+1} &= Y_{s+1}.
\end{aligned}
$$

For $i = 2, \ldots, s+1$



Incompressible Navier-Stokes: BDF-EXT3 (Explicit-Implicit)

- Incompressible equations are integrated in time with **BDF-EXT3** scheme [4].

1. Explicit convection:

$$\widehat{\mathbf{u}} = \frac{1}{\gamma_0}\sum_{q=0}^{s-1}\alpha_q \mathbf{u}^{n-q} + \frac{\Delta t}{\gamma_0}\sum_{q=0}^{s-1}\beta_q \mathbf{N}(\mathbf{u}^{n-q})$$

2. Pressure projection:

$$\frac{\Delta t}{\gamma_0}\nabla^2 p^{n+1} = \nabla\cdot\widehat{\mathbf{u}}, \qquad \widehat{\widehat{\mathbf{u}}} = \widehat{\mathbf{u}} - \frac{\Delta t}{\gamma_0}\nabla p^{n+1}$$

3. Implicit viscous solve:

$$\frac{\gamma_0\mathbf{u}^{n+1} - \widehat{\widehat{\mathbf{u}}}}{\Delta t} = \nu\nabla^2\mathbf{u}^{n+1}$$

- Pressure Poisson and viscous solves use preconditioned conjugate gradient (PCG) method.
- Two point fluxes for convective terms are also implemented (Veldman and Verstappen)

$$F^{\#} = \begin{bmatrix} \{\!\{u\}\!\} & \{\!\{u\}\!\} \\ \{\!\{v\}\!\} & \{\!\{u\}\!\} \\ \{\!\{w\}\!\} & \{\!\{u\}\!\} \end{bmatrix}, \quad G^{\#} = \begin{bmatrix} \{\!\{u\}\!\} & \{\!\{v\}\!\} \\ \{\!\{v\}\!\} & \{\!\{v\}\!\} \\ \{\!\{w\}\!\} & \{\!\{v\}\!\} \end{bmatrix}, \quad H^{\#} = \begin{bmatrix} \{\!\{u\}\!\} & \{\!\{w\}\!\} \\ \{\!\{v\}\!\} & \{\!\{w\}\!\} \\ \{\!\{w\}\!\} & \{\!\{w\}\!\} \end{bmatrix}.$$

Wall modelling formulations, from equilibrium to non equilibrium

To introduce modeling strategies, we start from the **Favre – averaged momentum equation**:

$$\overline{\rho}\frac{\partial \tilde{u}_i}{\partial t} + \overline{\rho}\tilde{u}_j\frac{\partial \tilde{u}_j}{\partial x_j} + \frac{\partial \overline{p}}{\partial x_i} = \frac{\partial \overline{\tau_{ij}}}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\overline{\rho}\widetilde{u_i''u_j''}\right)$$

We propose **three different models of increasingly physical complexity**:

Equilibrium assumptions

Time–derivative, convective, pressure–gradient and wall–parallel viscous diffusion are neglected:

$$\frac{\partial \overline{\tau_{ij}}}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\overline{\rho}\widetilde{u_i''u_j''}\right) = 0$$

We write it in the wall normal direction $y$with the Boussinesq eddy–viscosity approximation:

$$\overline{\rho}\widetilde{u_i''u_j''} = \overline{\mu}_T\frac{d\overline{u}}{dy}$$

Leading to:

$$\begin{cases} \dfrac{d}{dy}\left[\overline{\mu} + \overline{\mu}_T\dfrac{d\overline{u}}{dy}\right] = 0 \\ \overline{u}(y=0) = 0 \\ \overline{u}(y=H_{WM}) = U_{LES} \end{cases}$$

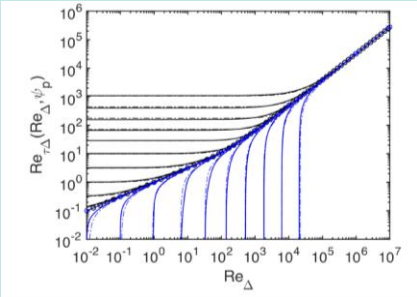
By integrating this system of equations we obtain the algebraic **Reichardt wall law**.

Meneveau's non–equilibrium model [3]

Pressure-gradient terms are also included:

$$\frac{d}{dy}\left[\overline{\mu} + \overline{\mu}_T\frac{d\overline{u}}{dy}\right] = \frac{\partial \overline{p}}{\partial x}$$

Dimensionless parameters that depend on the exchange location and local pressure gradient are introduced. **ODE is solved across a range of parameter combinations.**



A fitted algebraic expression is derived, **linking wall shear stress to the velocity and the pressure gradient** at the matching location.

Kamogawa's non–equilibrium model [4]

Within this formulation, convective terms are also included:

$$\frac{d}{dy}\left[\overline{\mu} + \overline{\mu}_T\frac{d\overline{u}}{dy}\right] = \frac{\partial \overline{p}}{\partial x} + \tilde{u}\frac{\partial \tilde{u}}{\partial x} + \tilde{v}\frac{\partial \tilde{u}}{\partial y}$$

The main assumption is considering a balance between convection and pressure gradient in the log – layer region.

Convection is then modeled as a function of pressure gradient and a blending function:

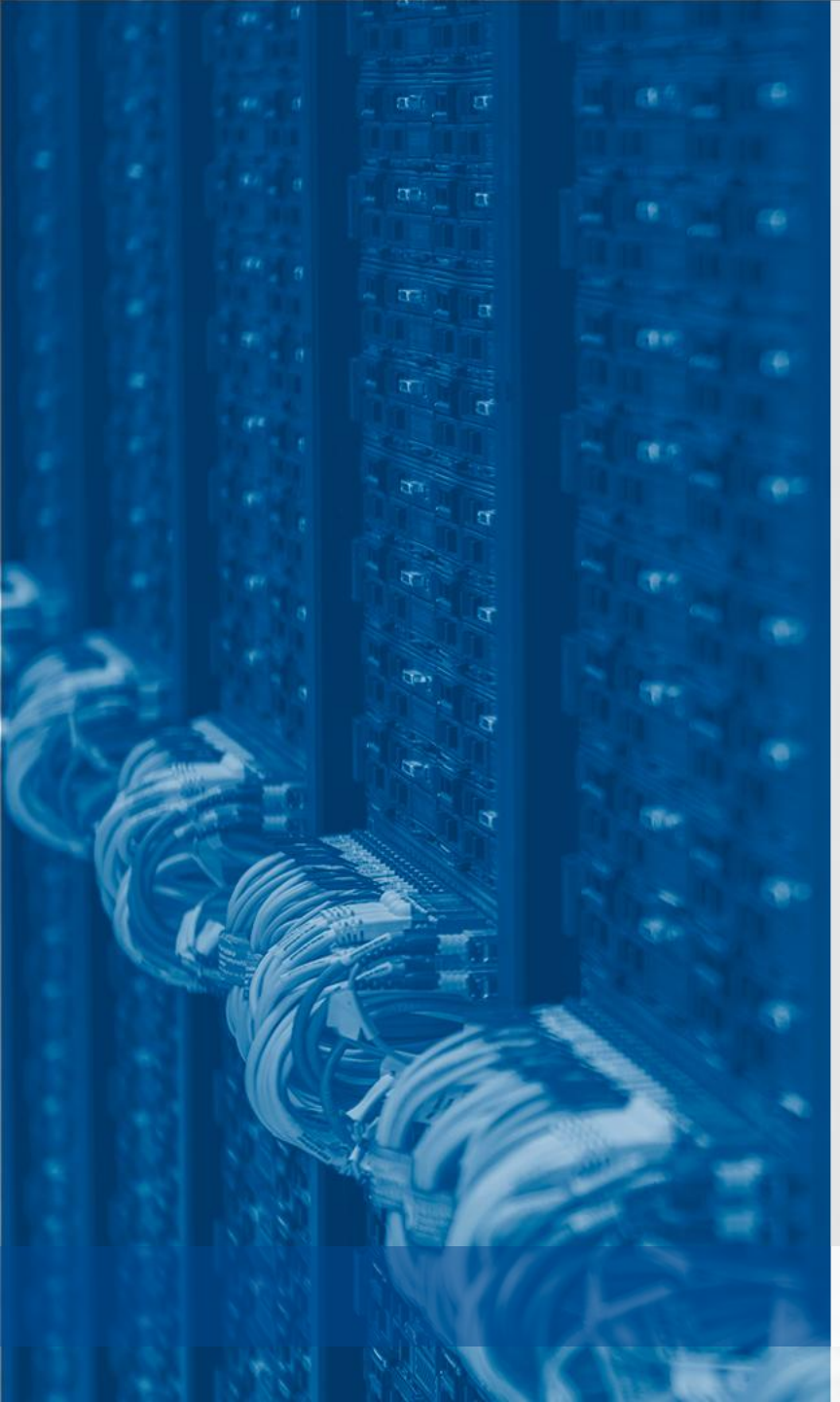
$$\tilde{u}\frac{\partial \tilde{u}}{\partial x} + \tilde{v}\frac{\partial \tilde{u}}{\partial y} = -\frac{\partial \overline{p}}{\partial x}F(u)$$

Leading to:

$$\frac{d}{dy}\left[\overline{\mu} + \overline{\mu}_T\frac{d\overline{u}}{dy}\right] = \frac{\partial \overline{p}}{\partial x} - \frac{\partial \overline{p}}{\partial x}\left(\frac{\rho u^2}{\rho U_{H_{WM}}^2}\right)$$

[3] Meneveau, C. (2020). A note on fitting a generalised Moody diagram for wall modelled large-eddy simulations. *Journal of Turbulence*, 21(11), 650-673.

[4] Kamogawa, R., Tamaki, Y., & Kawai, S. (2023). Ordinary-differential-equation-based nonequilibrium wall modeling for large-eddy simulation. *Physical Review Fluids*, 8(6), 064605.

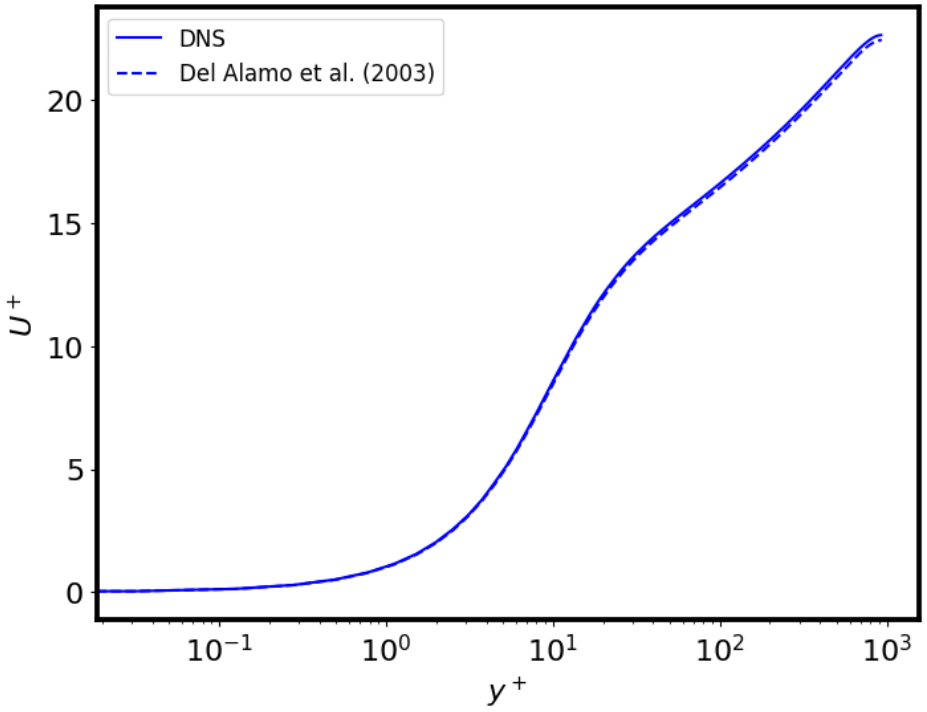


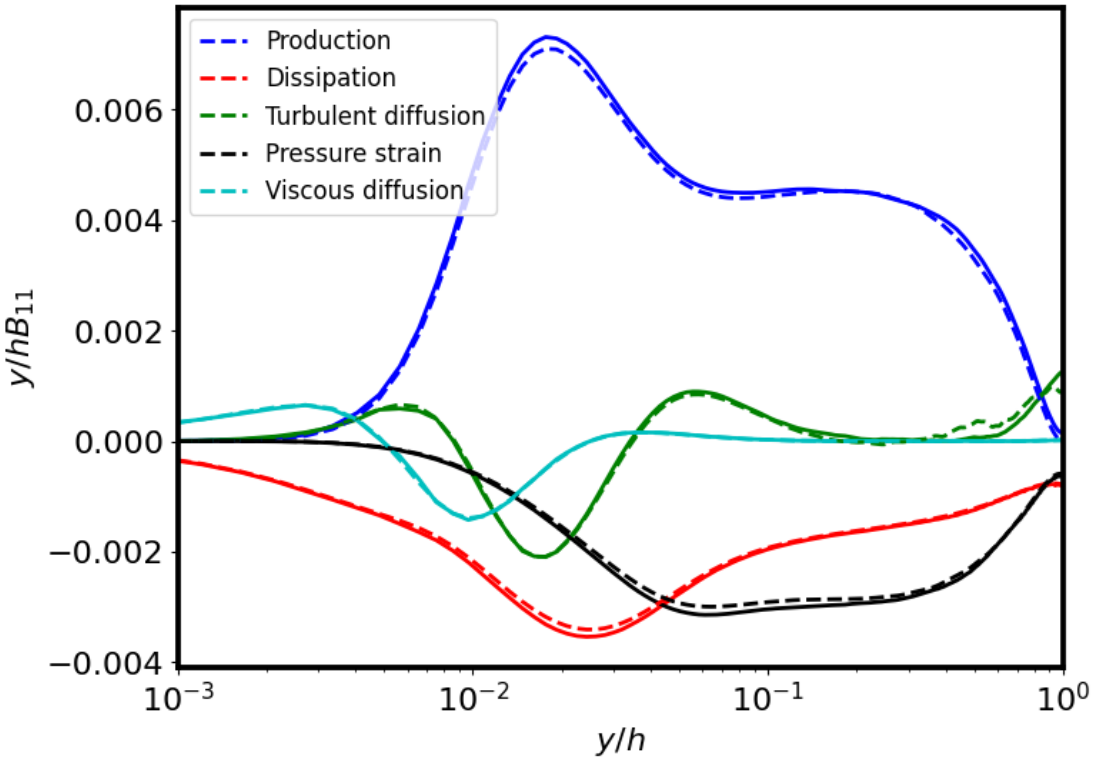
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Basic Validations

Large-scale Computational Fluid Dynamics Group at **BSC-CNS**

Channel Flow (Retau = 950)

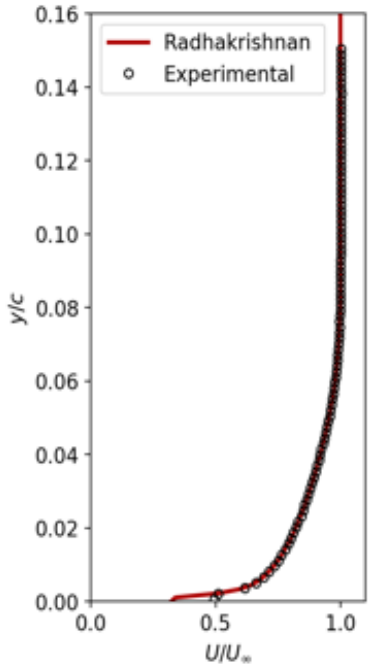




– Current work
- -Del Alamo et al.

- Polynomial order 6 with about 250 M DoF (y+ ~ 1 Dx+ ~ 8 Dz+ ~ 8) **2h run with 32 GPUs !!**
- No SGS model used
- Minimal differences given we use a smaller box (6hx2hx3h)
- After internal communication with Hoyas and Lozano-Duran, we found out that we see the same effects reported in : Adrián Lozano-Durán, Javier Jiménez; Effect of the computational domain on direct simulations of turbulent channels up to Re_τ = 4200. *Physics of Fluids* 1 January 2014; 26 (1): 011702. https://doi.org/10.1063/1.4862918

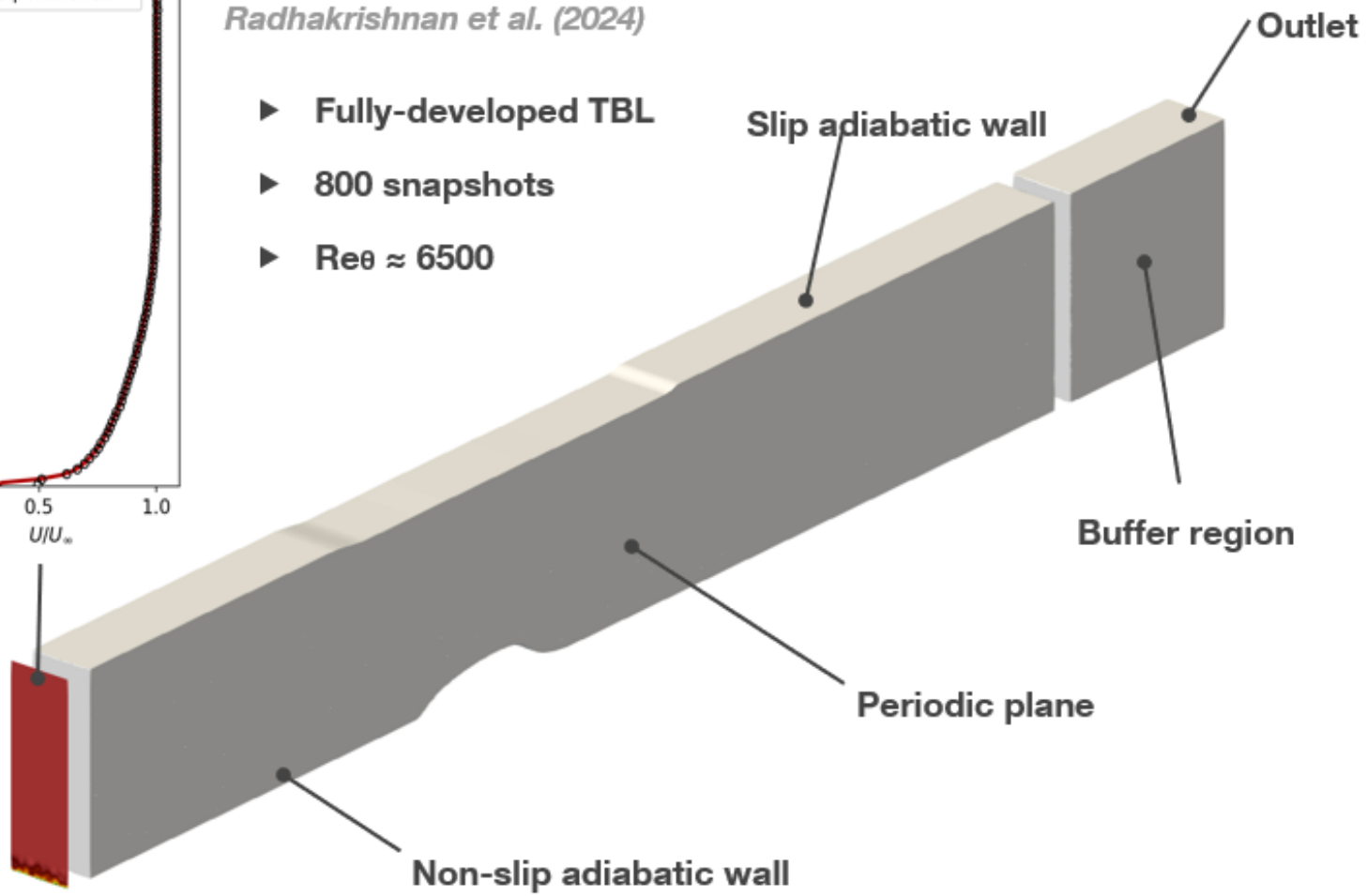
NASA hump



Unsteady inlet inflow condition

Radhakrishnan et al. (2024)

- Fully-developed TBL
- 800 snapshots
- $Re_\theta \approx 6500$



NASA hump

Wall resolved computations

GRID

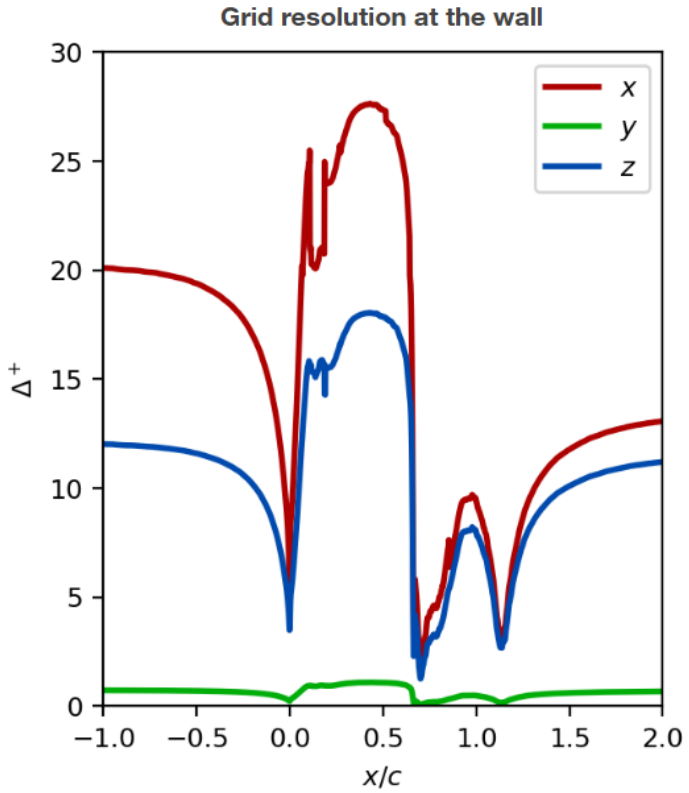
- $(L_x, L_z) = 7.14c, 0.3c$
- Node count = 2 165 410 873
- $(\Delta x^+, \Delta y^+, \Delta z^+) \approx$ 20, 0.7, 12

STATISTICS

- Steady-state initialization
- Transient period of 5 C.U.
- Averaging period of 20 C.U.

RESOURCES

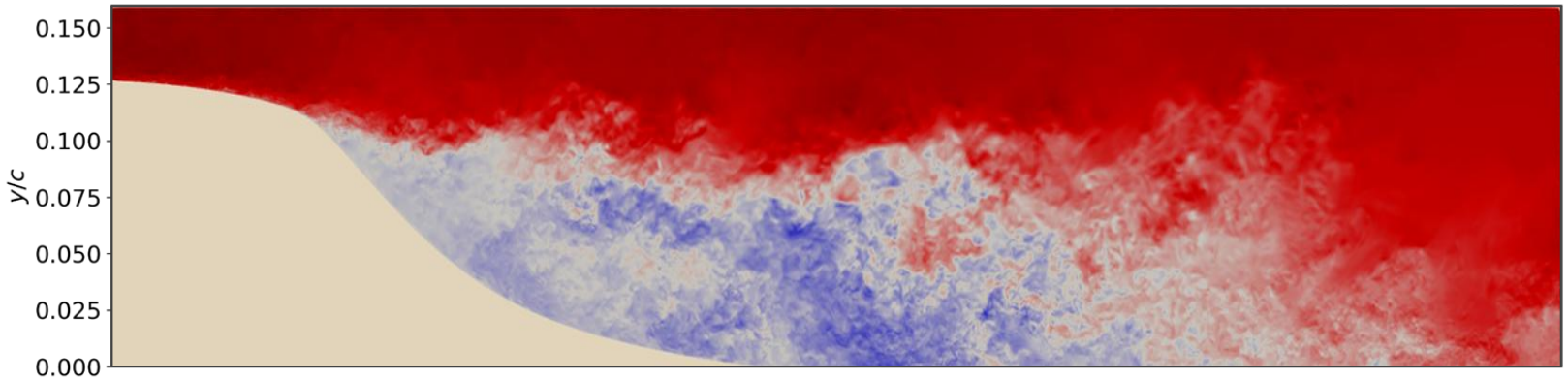
- $\Delta t \approx 5.8 \cdot 10^{-6}$ C.U.
- GPU count = 100 Nvidia H100
- Simulation speed ≈ 1.5 C.U./d
- Disk space ≈ $O(10^1\text{-}10^2)$ TB



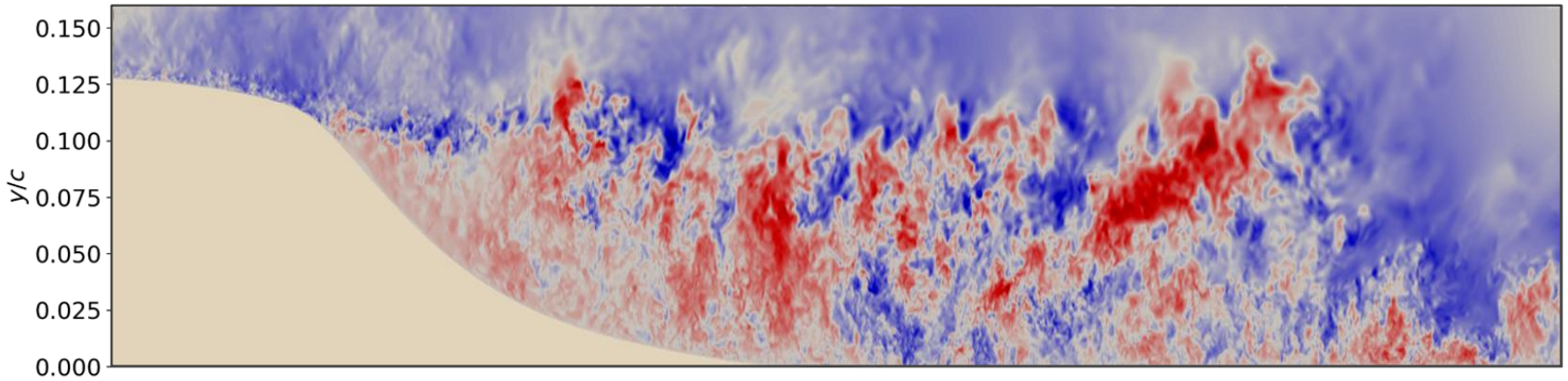
NASA hump

Wall resolved computations

Streamwise velocity profiles shifted $\Delta U/U_{ref} = 1.5$ in the horizontal axis

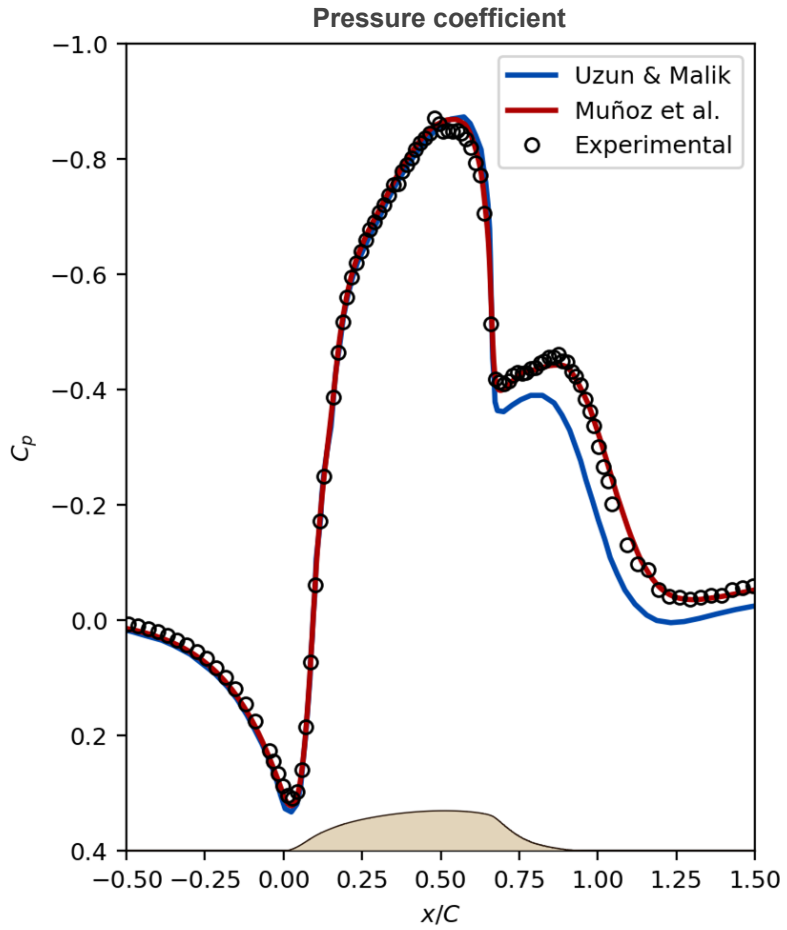


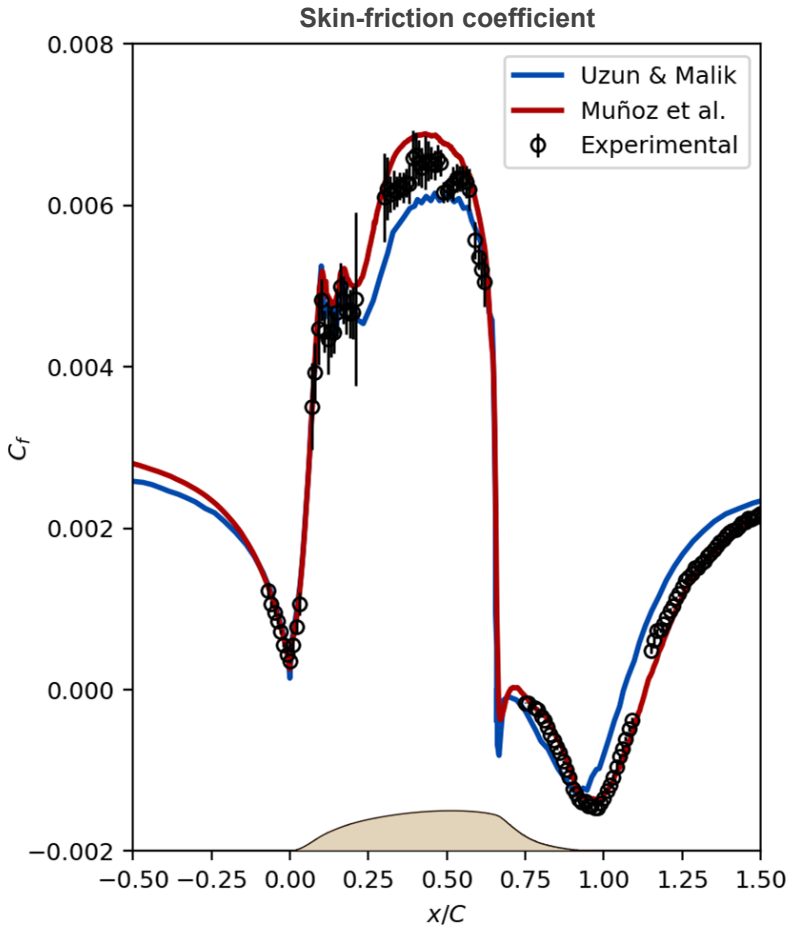
Wall-normal velocity profiles shifted $\Delta V/U_{ref} = 0.25$ in the horizontal axis

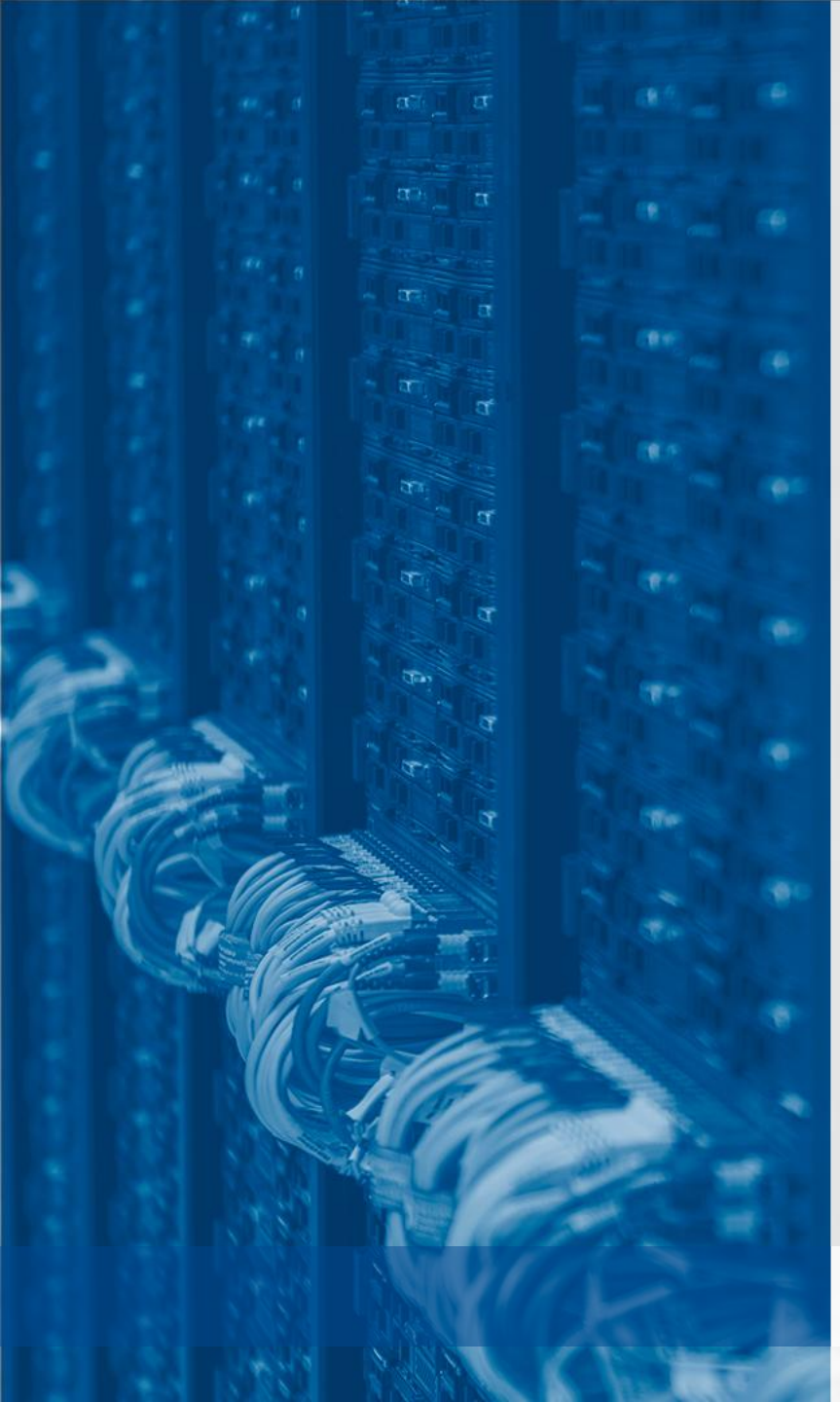


NASA hump

Wall resolved computations





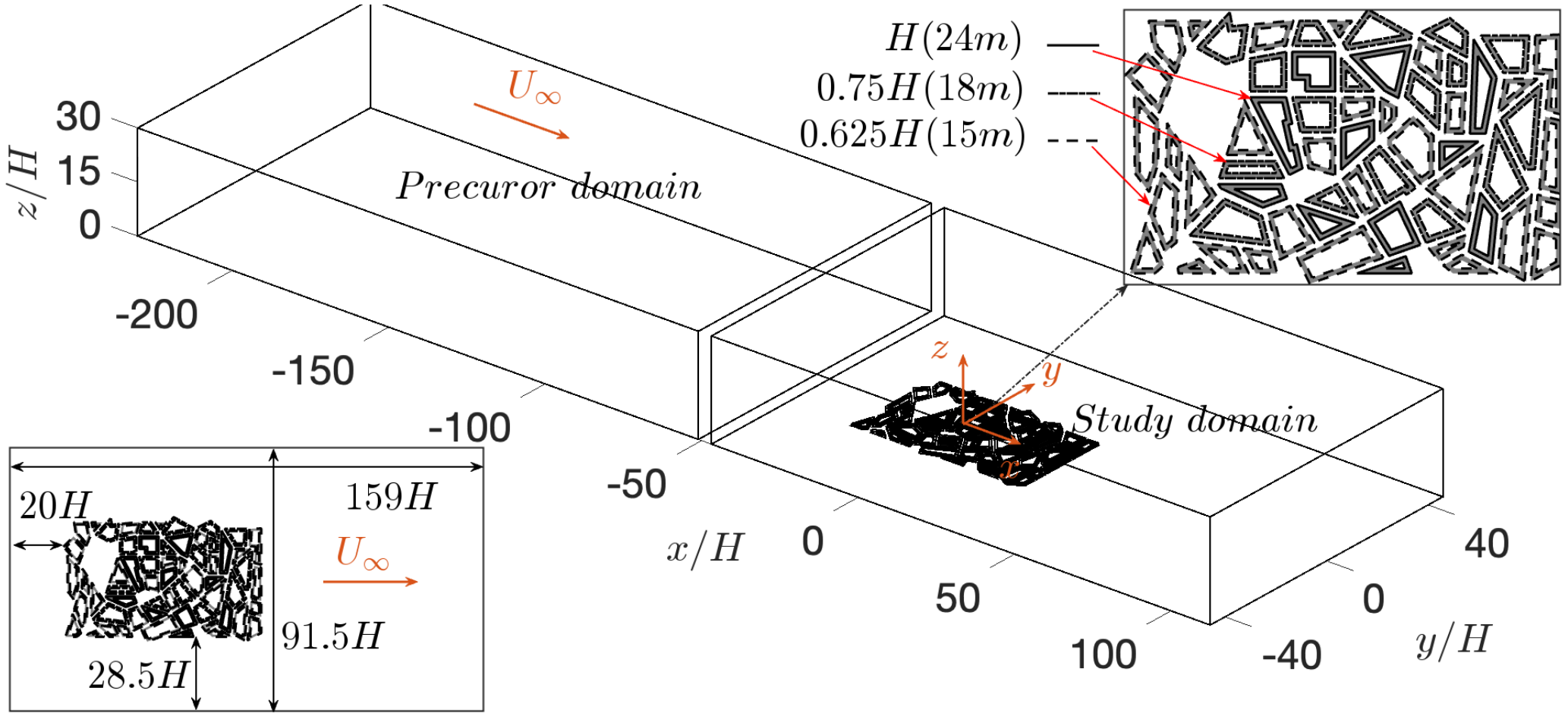


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

WMLES on complex geometries: Urban Flows

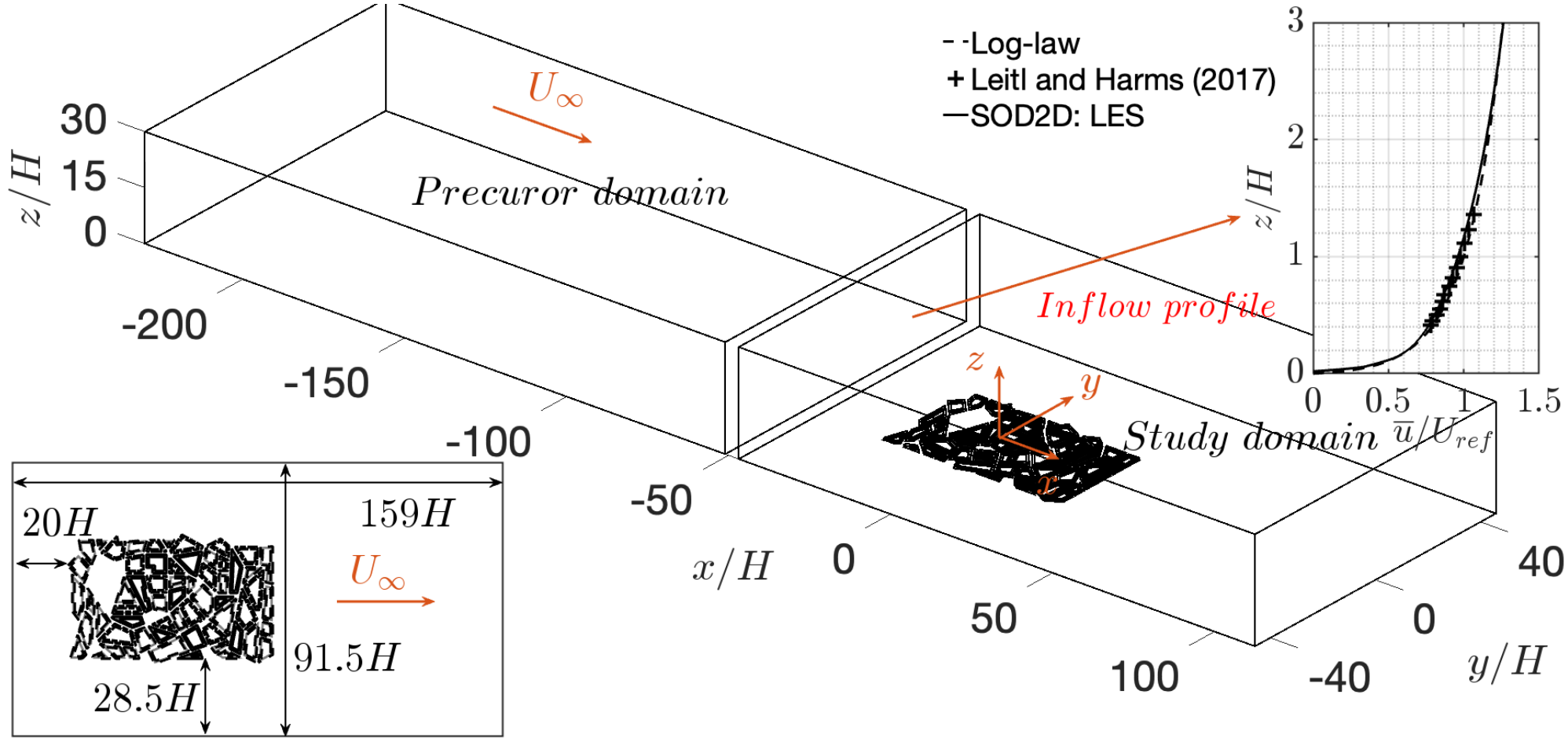
Large-scale Computational Fluid Dynamics Group at **BSC-CNS**

Atmospheric boundary layer over urban roughness: validation of large-eddy simulation – *Phys. Fluids* (2025)



Computational domain for the idealized urban: Michelstadt

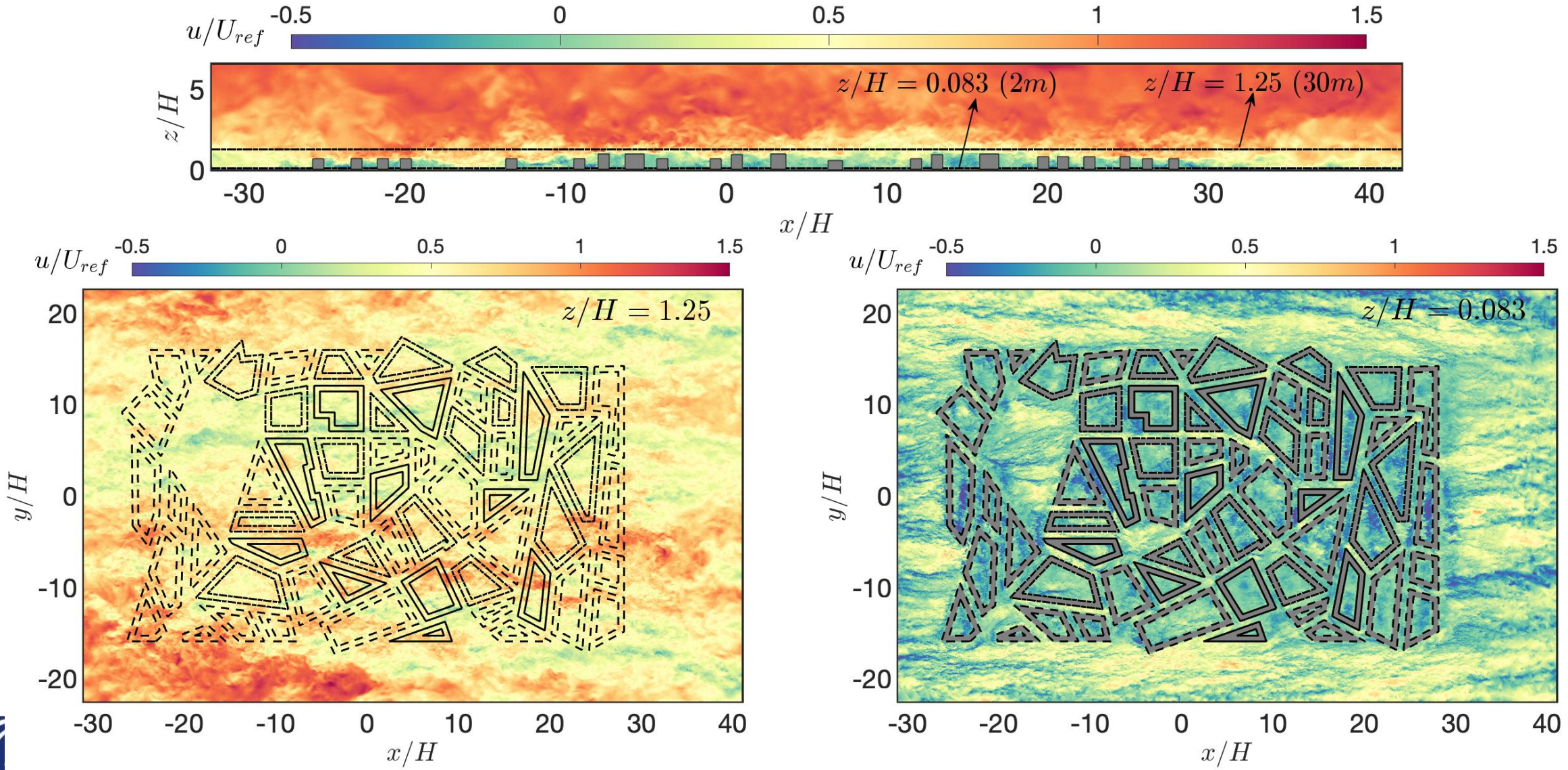
Atmospheric boundary layer over urban roughness: validation of large-eddy simulation – *Phys. Fluids* (2025)



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Numerical settings: $Re_H = 8.0 \times 10^6$; *DoF* = 217.8×10^6

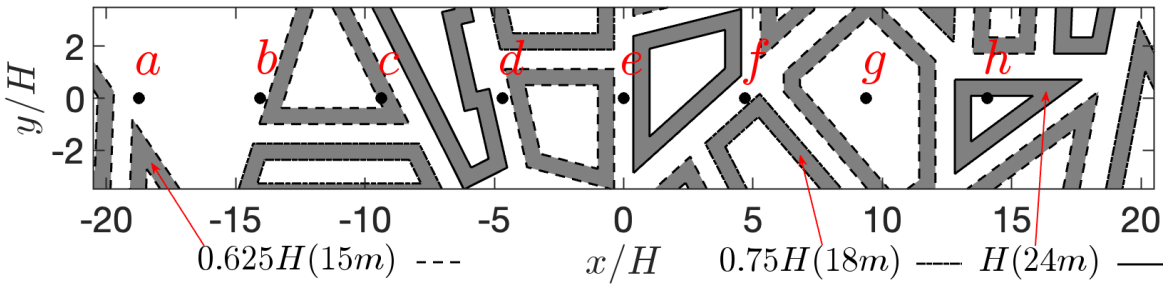
Atmospheric boundary layer over urban roughness: validation of large-eddy simulation – *Phys. Fluids* (2025)

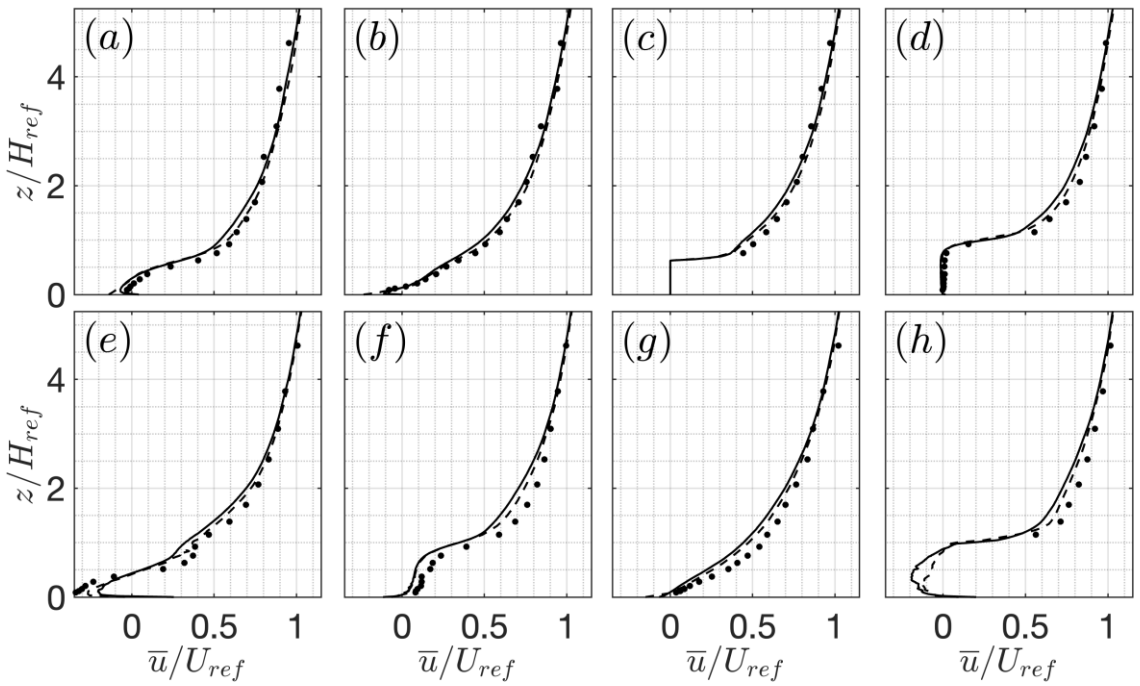


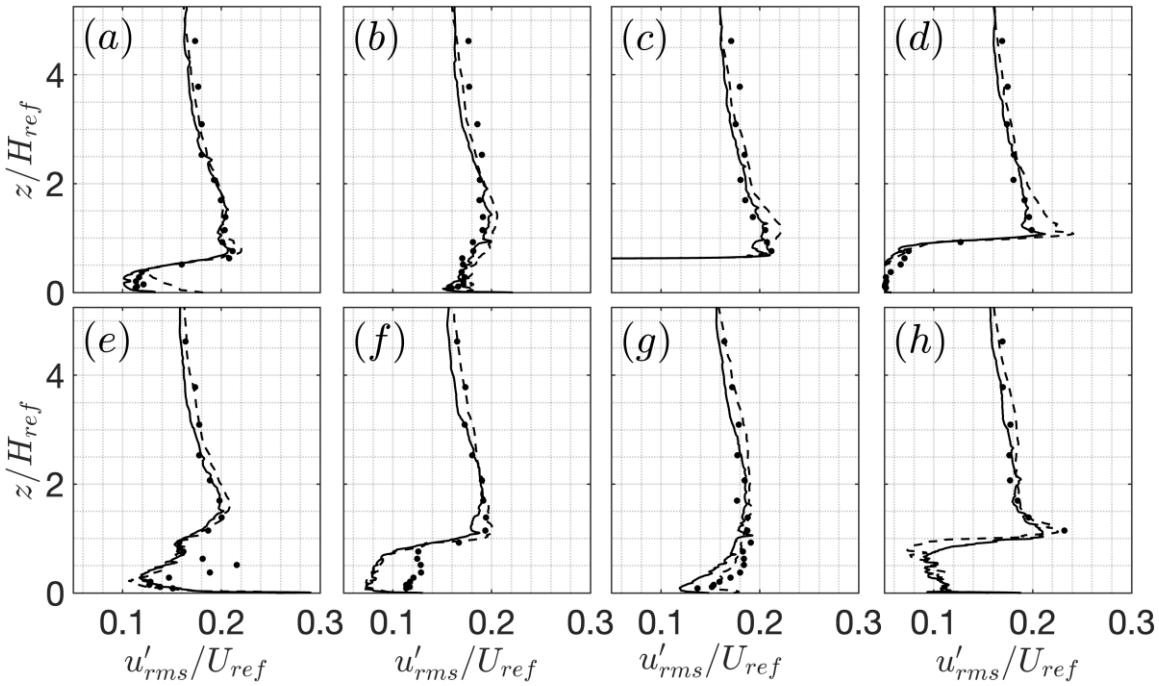
Centro Nacional de Supercomputación

Michelstadt: instantaneous streamwise velocity

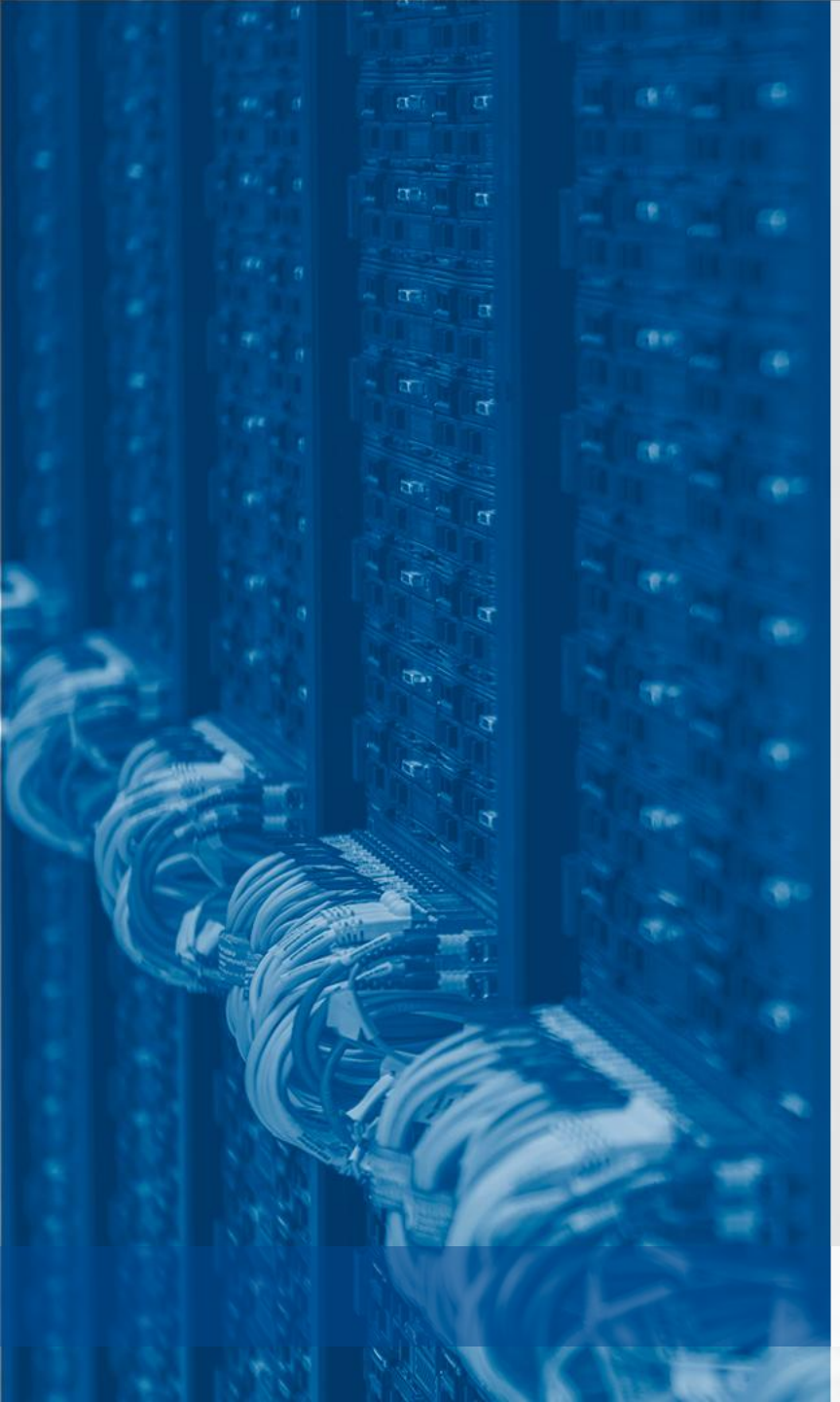
Atmospheric boundary layer over urban roughness: validation of large-eddy simulation – *Phys. Fluids* (2025)







Validation: --- coarse mesh, – fine mesh, • Leitl & Harms (2017)

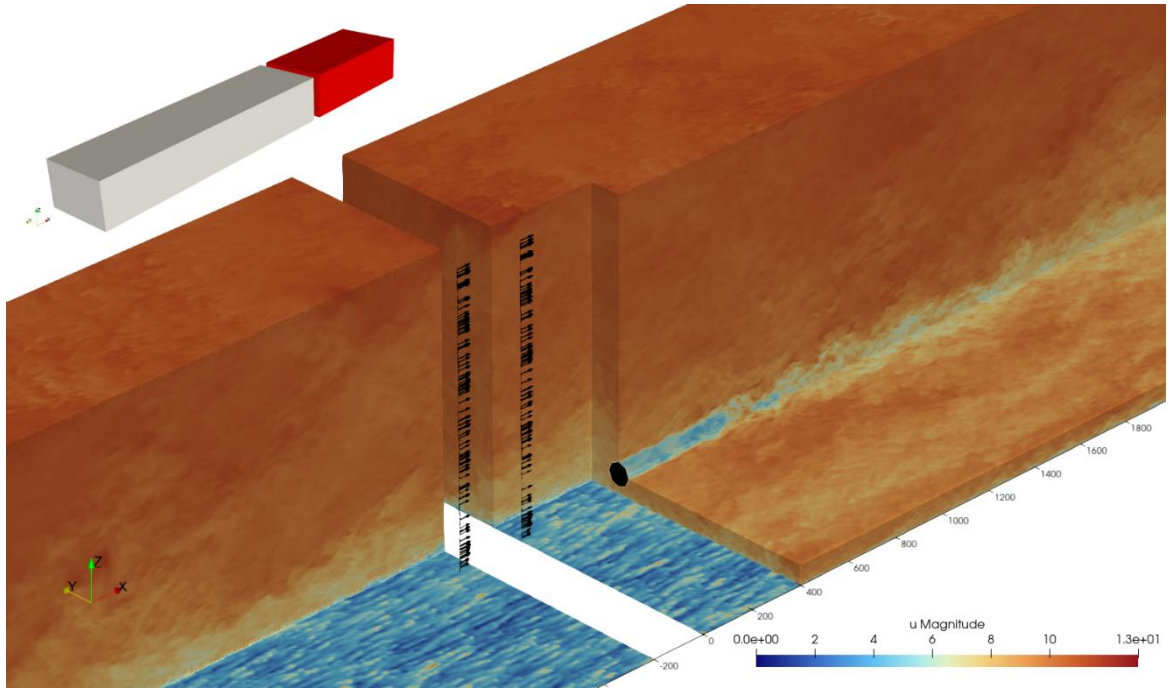


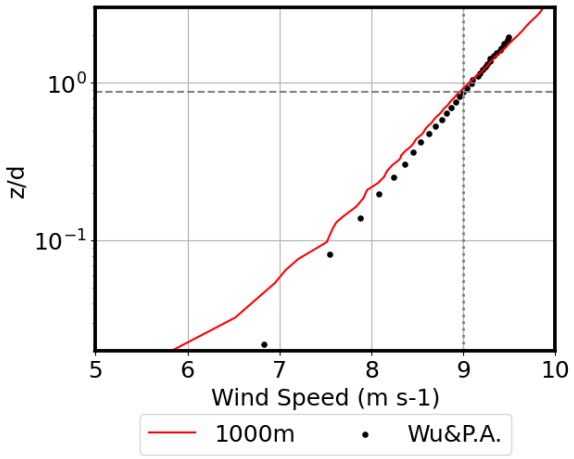
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

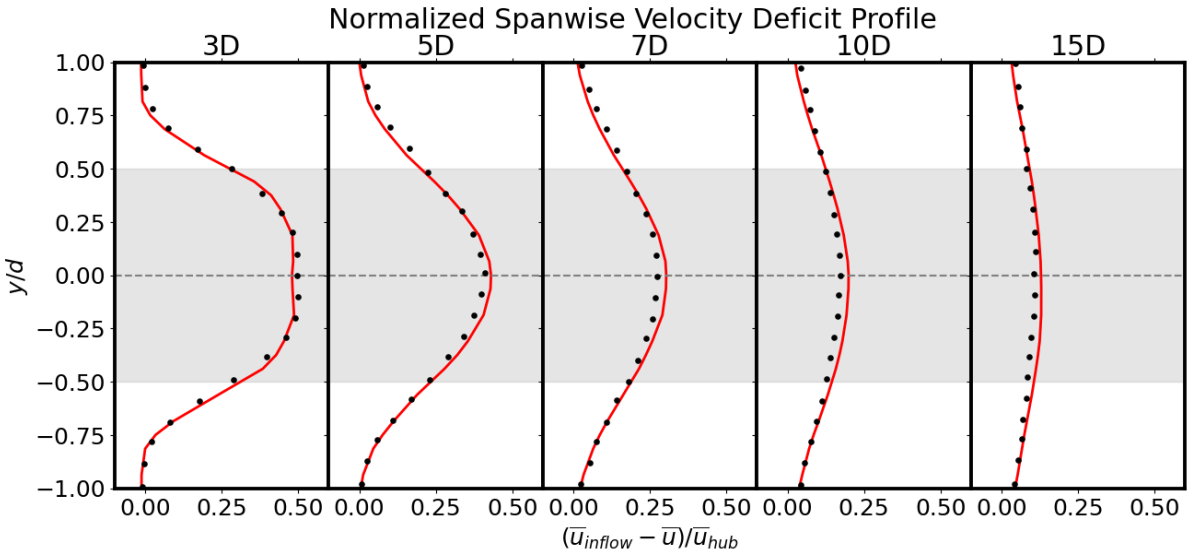
WMLES on complex geometries: Wind Energy

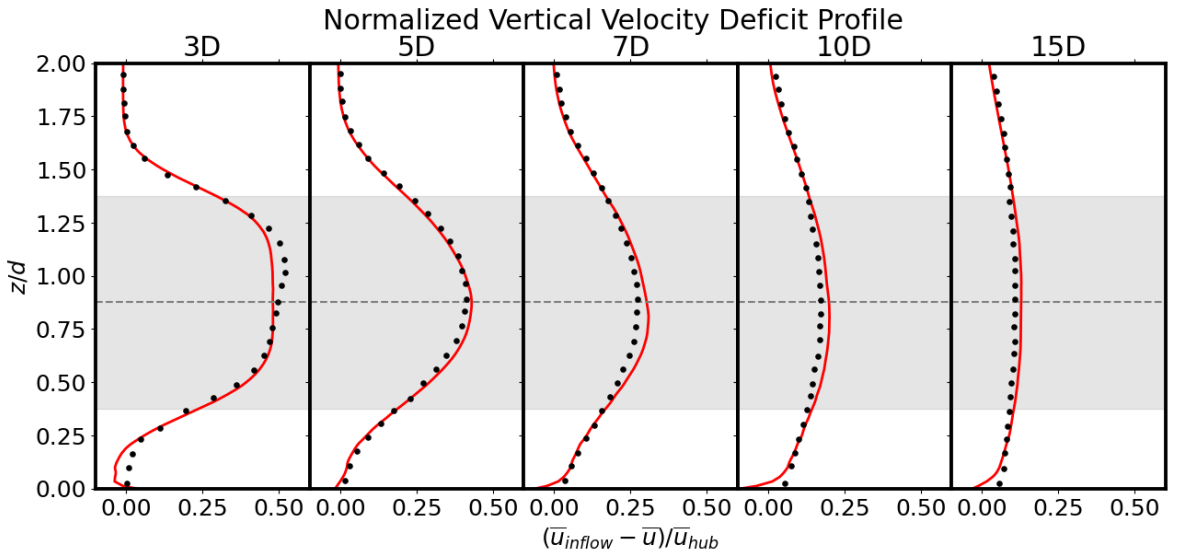
Large-scale Computational Fluid Dynamics Group at **BSC-CNS**

OCEANOSTRUM project

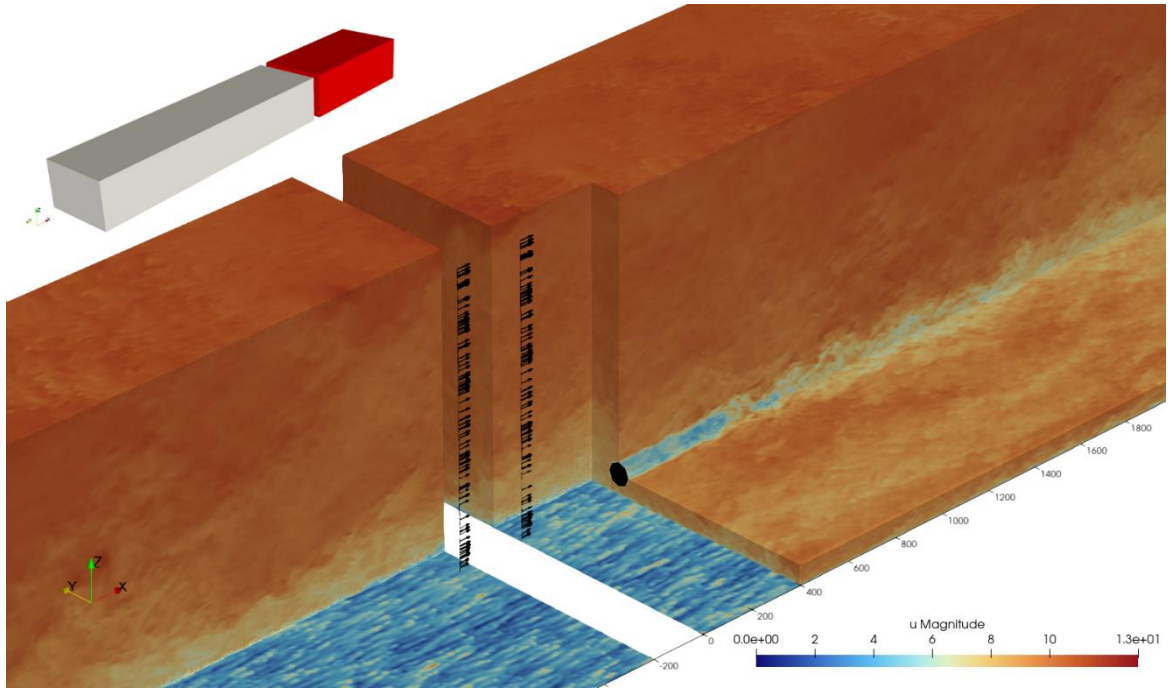


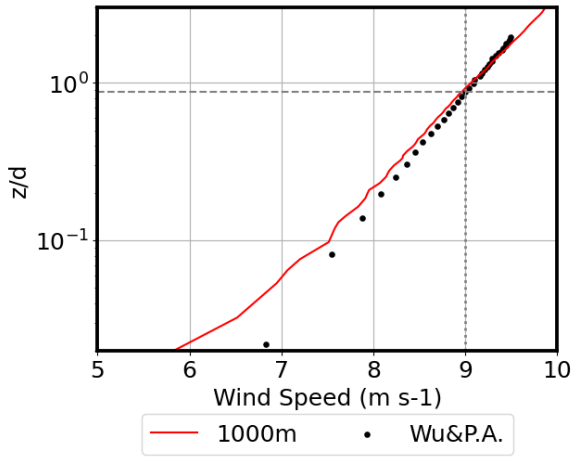


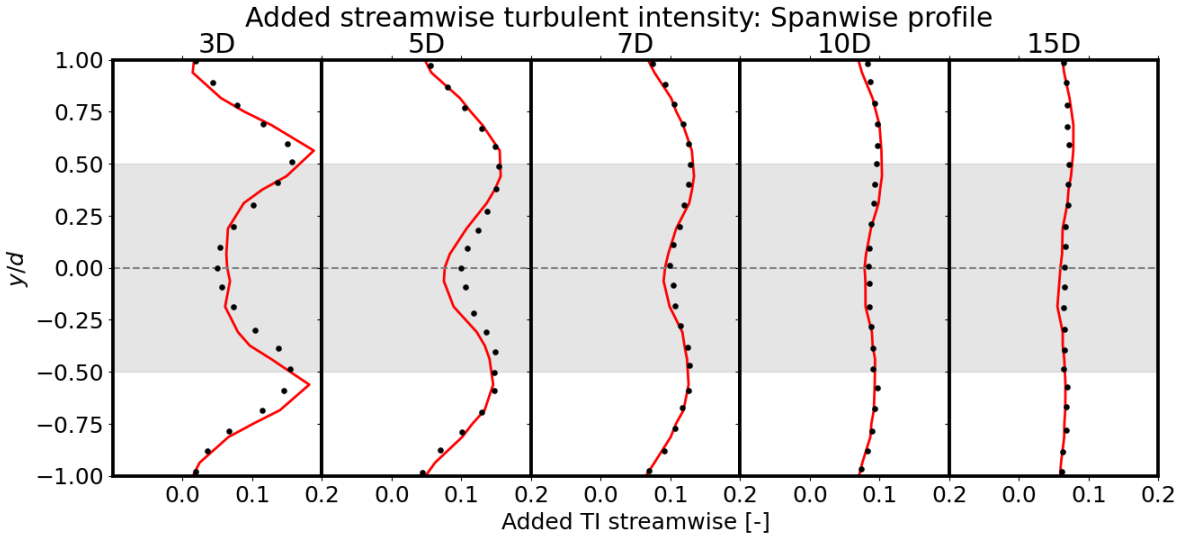


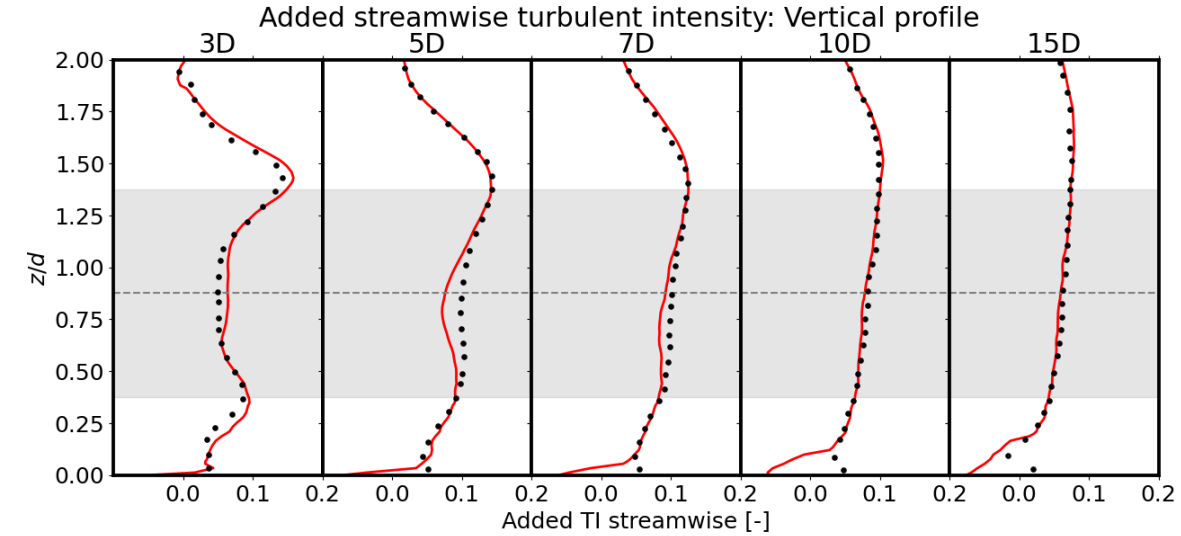


OCEANOSTRUM project



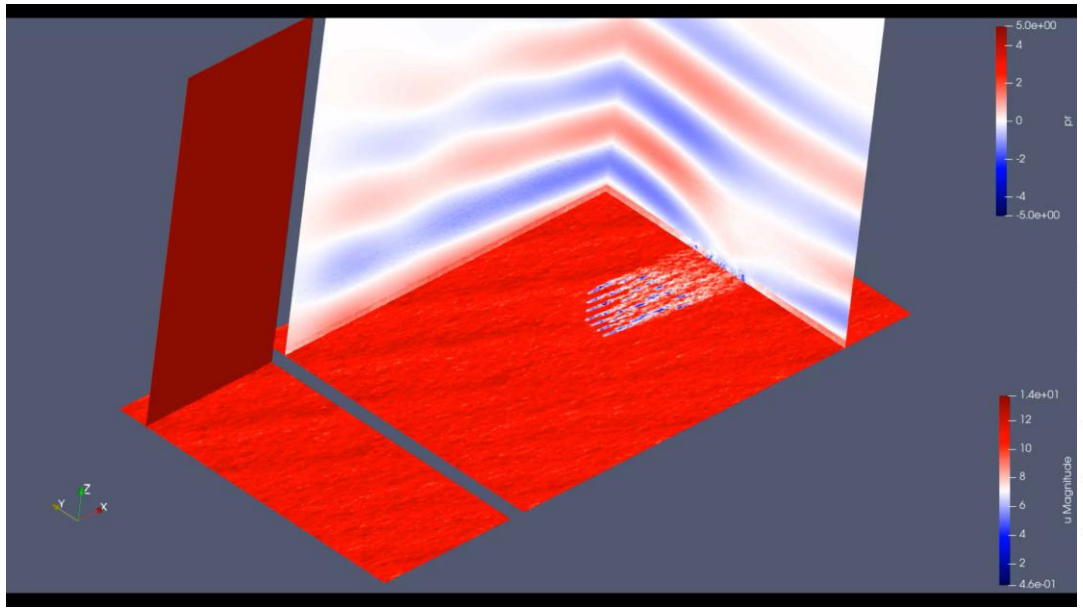


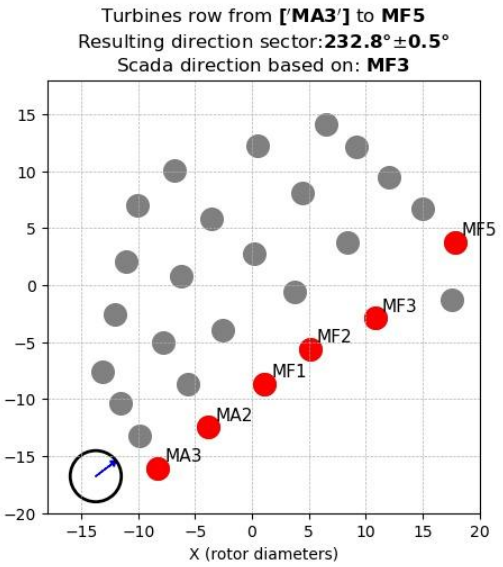


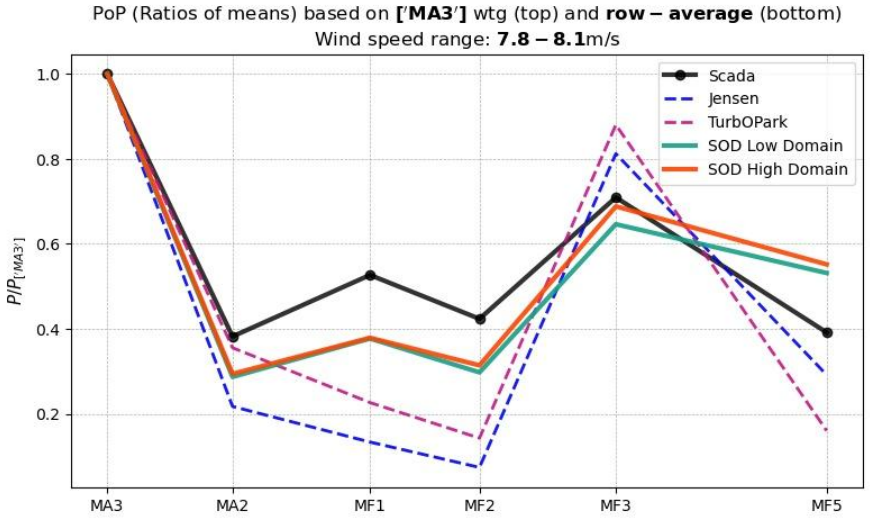


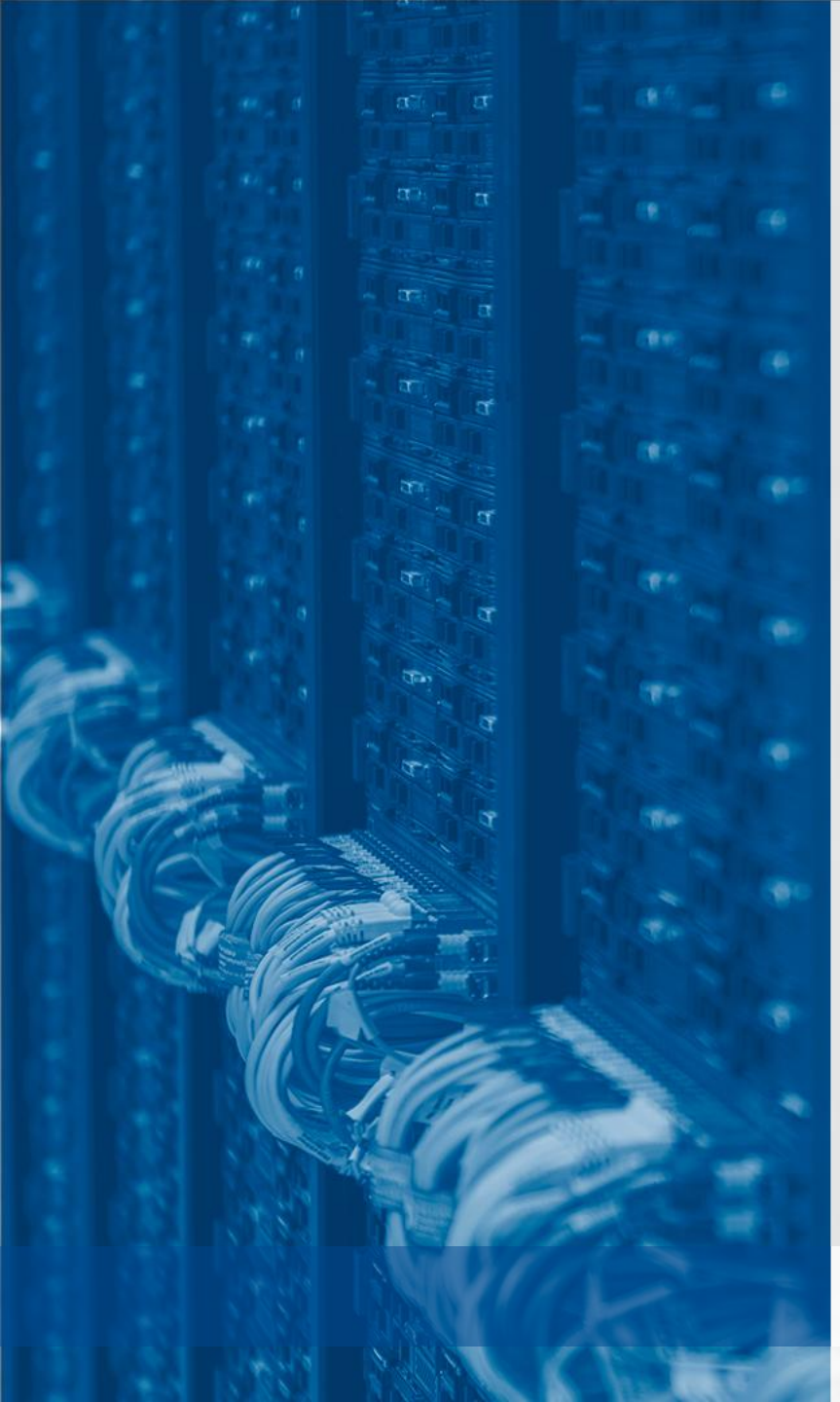
SOD2D Wu & Porte-Agel Rotor-range

OCEANOSTRUM project









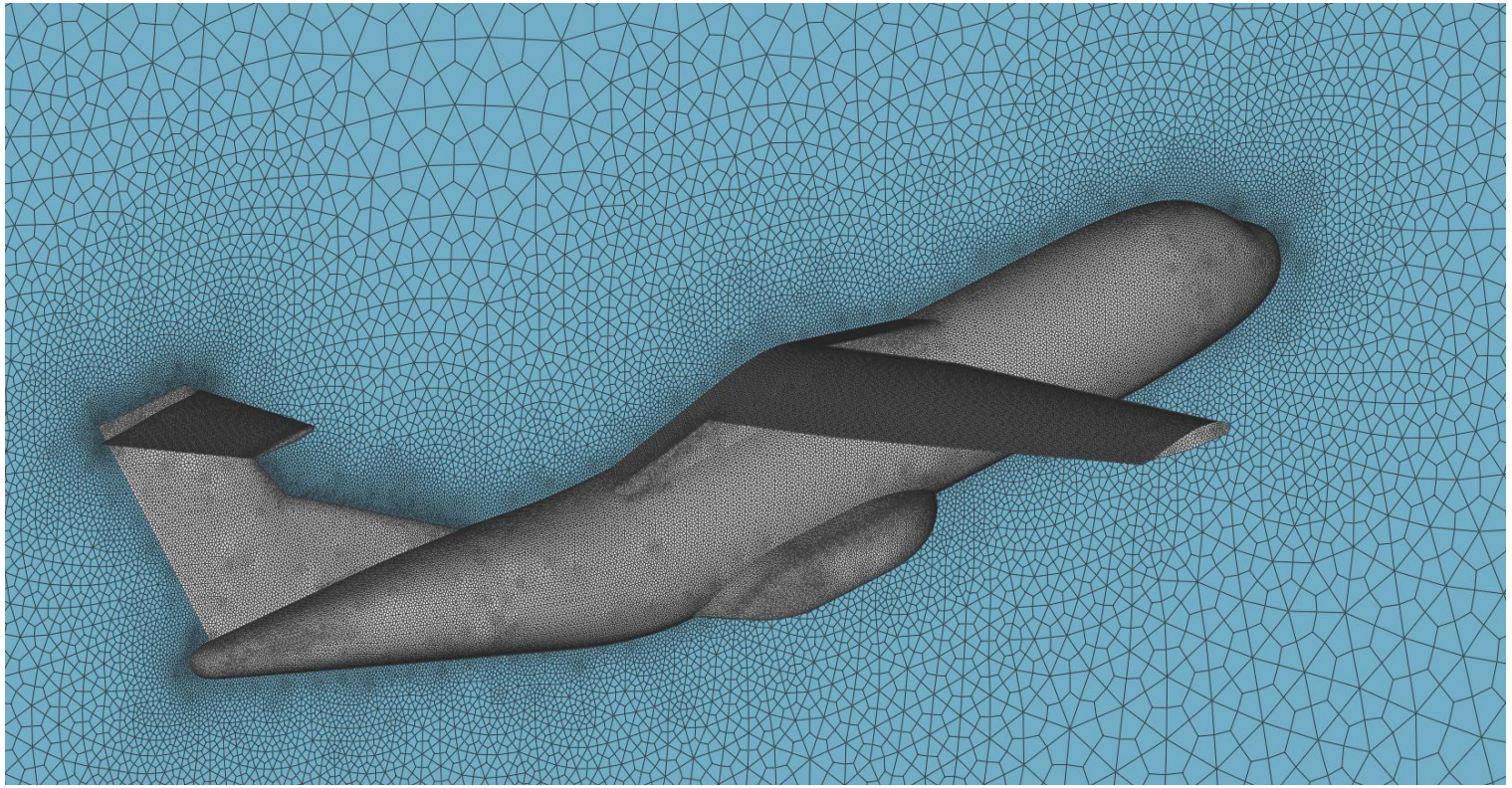
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

WMLES on complex geometries: Aeronautics

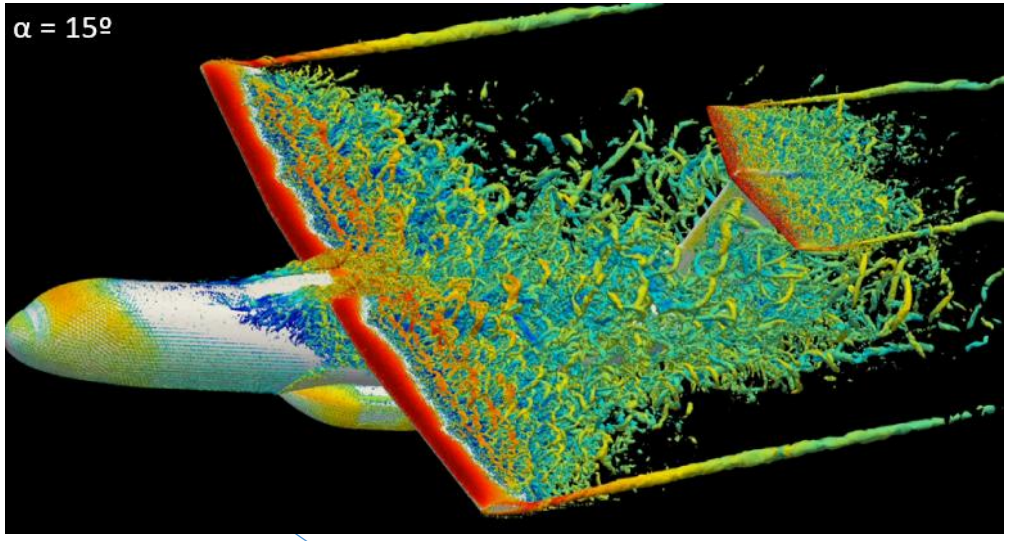
Large-scale Computational Fluid Dynamics Group at **BSC-CNS**

WMLES of Airbus DS geometries

- Mesh generated for p2, p3 and p4 SEM with low dissipation entropy viscosity
- About 100M to 1.5B of DoF for the mesh no symmetry domain
- Clean geometry and high lift configurations considered (Re from 5 to 20 M, aoa from -5 to 30)
- Mesh tested in the new Marenostrum 5 machine (MN5), simulations times from 12 to 48h.



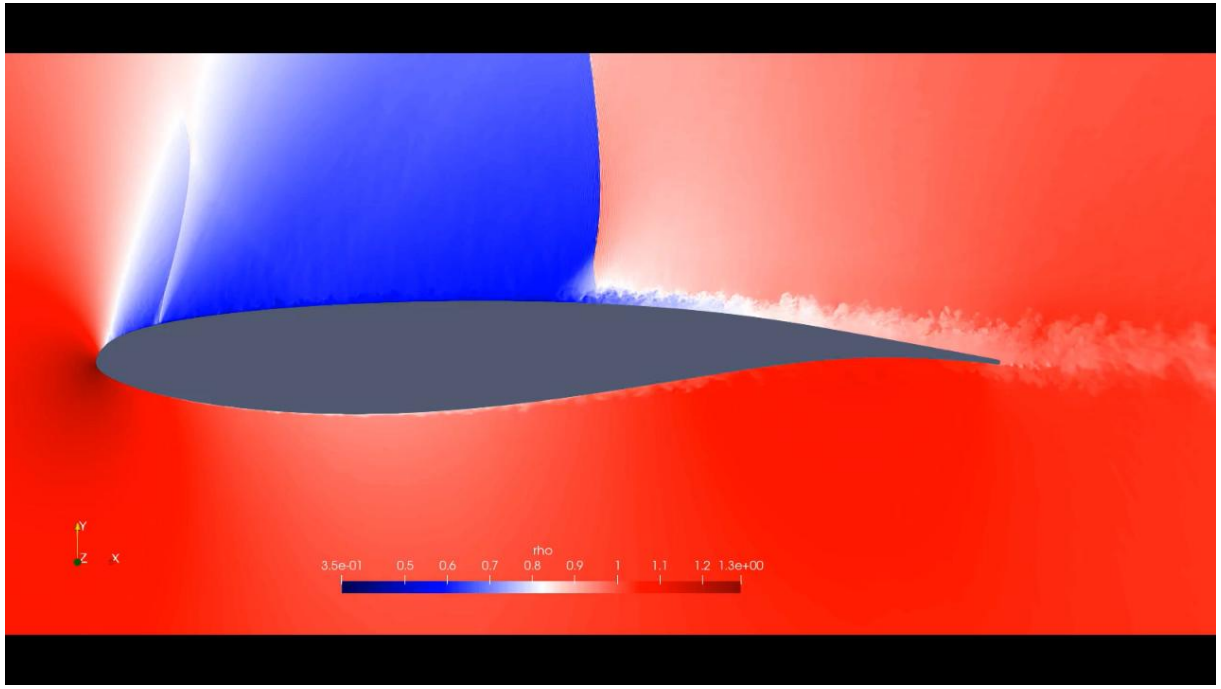
WMLES of Airbus DS geometries



From simple and clean geometries (100M) to high lift configurations with very detailed geometries (1.5B)

WMLES for Predicting Transonic Buffet Dynamics Over the OAT15A Airfoil

The transonic buffet is an unsteady aerodynamic phenomenon occurring at transonic speeds ($0.7 < M < 0.9$), typically over supercritical airfoils.



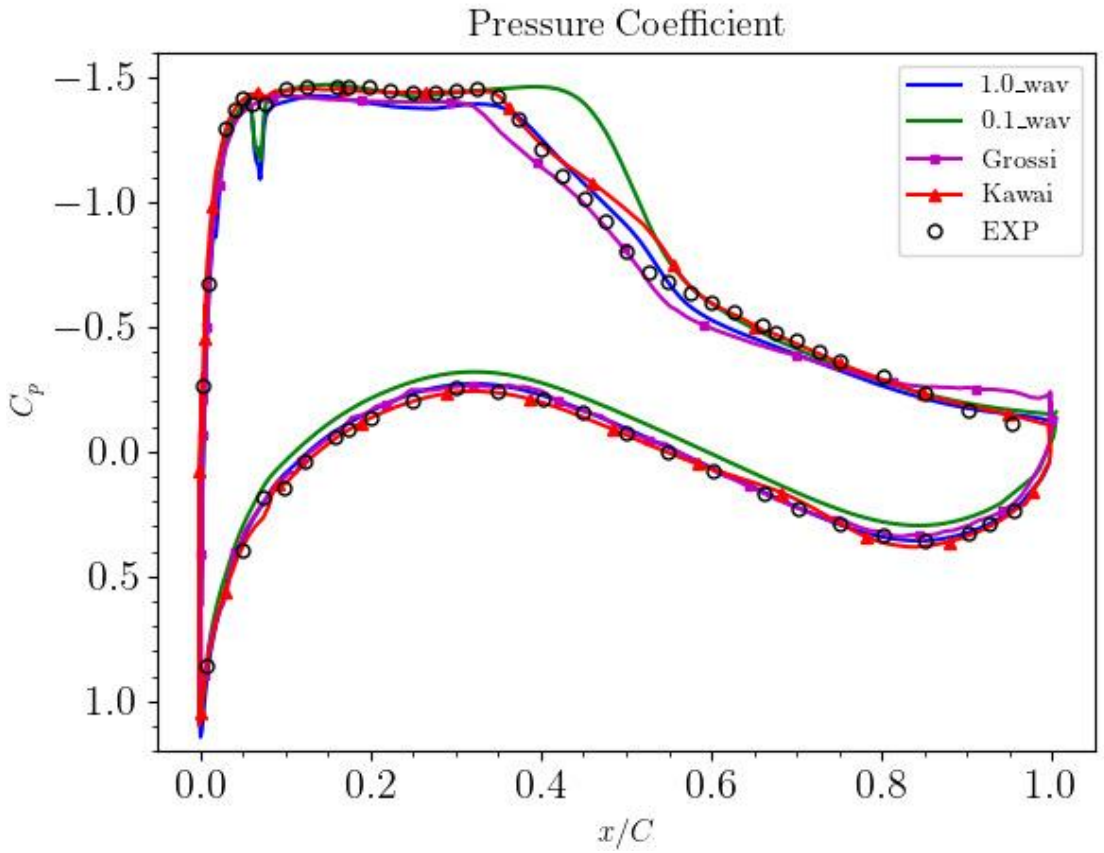
Key Features:

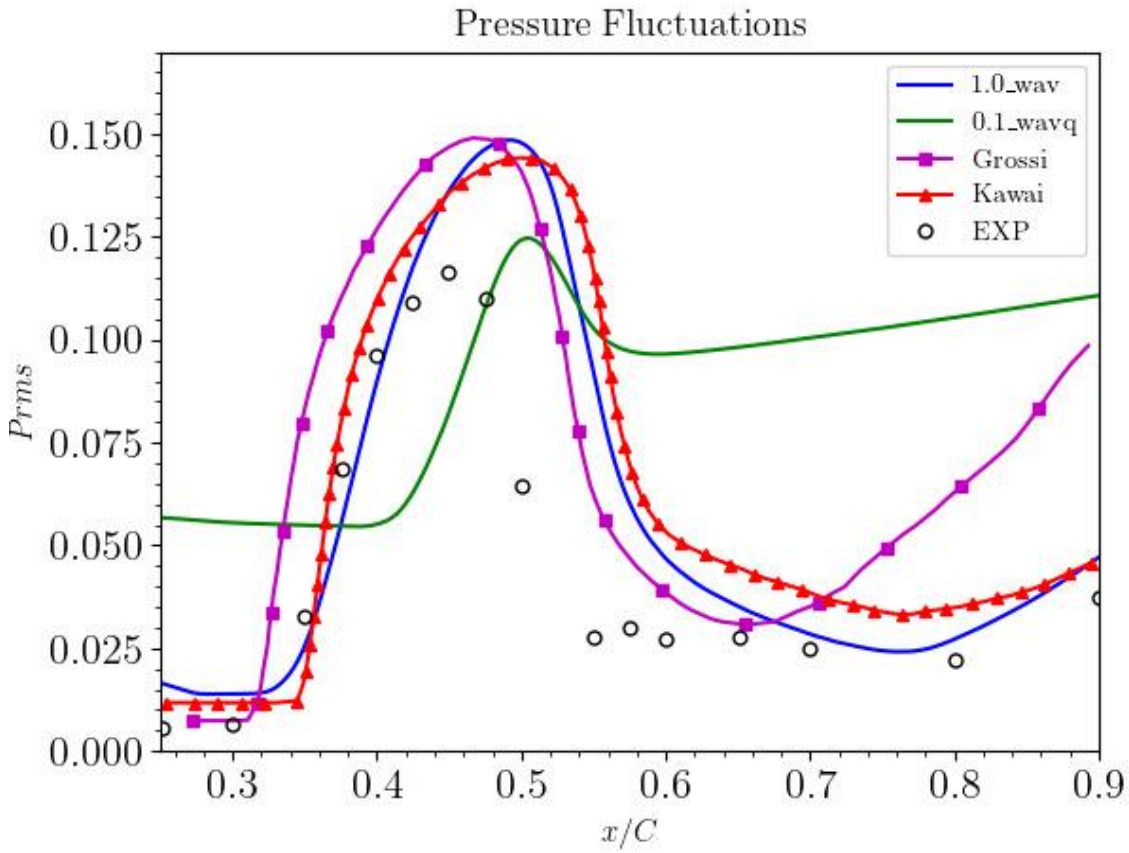
- Shock-boundary layer interaction.
- Unsteady shock oscillations.
- Shock-induced separation and reattachment.
- Pressure and lift fluctuations affecting aircraft stability.

Critical for high-speed aircraft design and control. Requires high-fidelity CFD methods (e.g., WMLES) to predict and analyze.

Figure 1: Shock-induced separation on a transonic airfoil.

WMLES for Predicting Transonic Buffet Dynamics Over the OAT15A Airfoil





WMLES for Predicting Transonic Buffet Dynamics Over the OAT15A Airfoil

Velocity Profiles

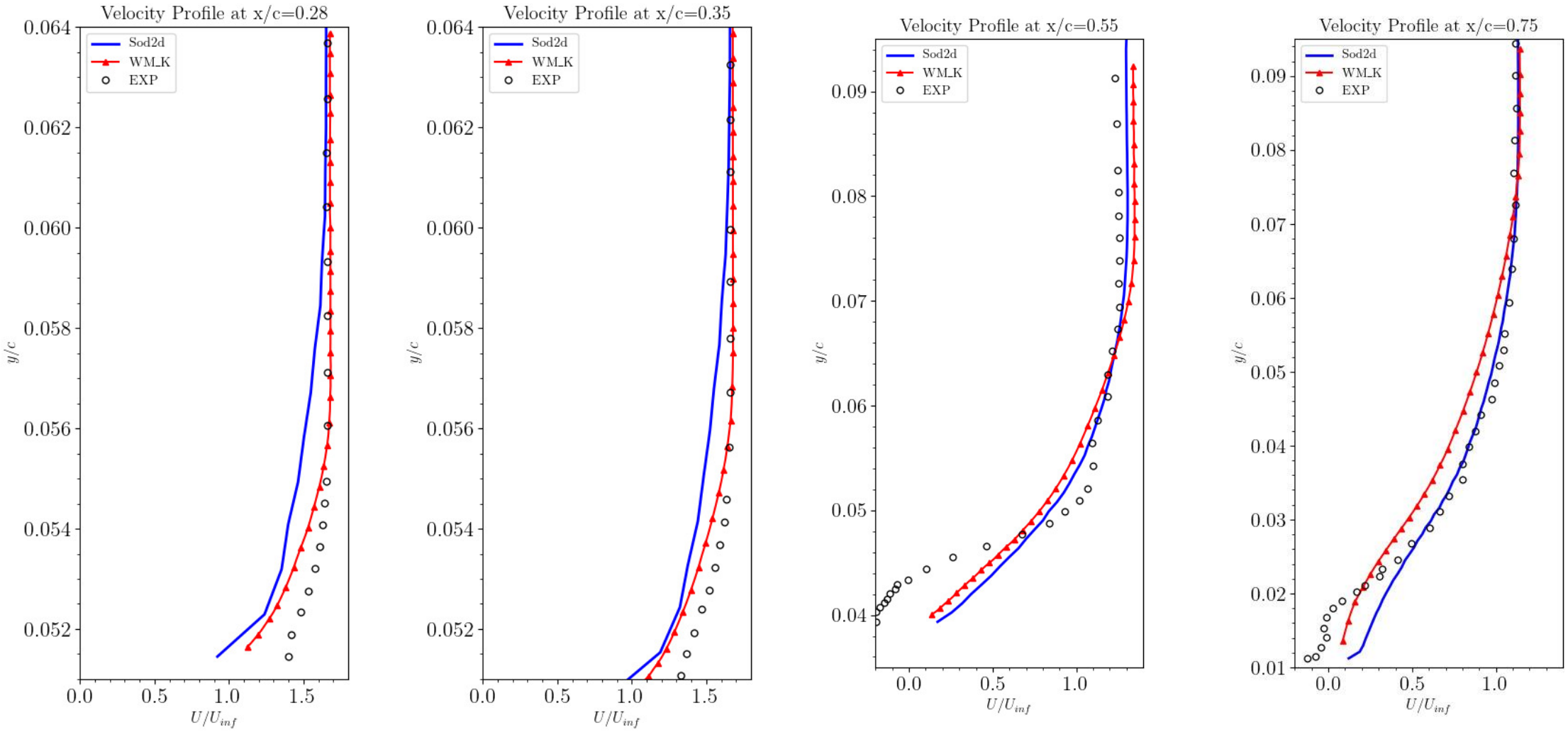


Figure 7: Time-averaged velocity profiles .

WMLES for Predicting Transonic Buffet Dynamics Over the OAT15A Airfoil

Velocity Profiles

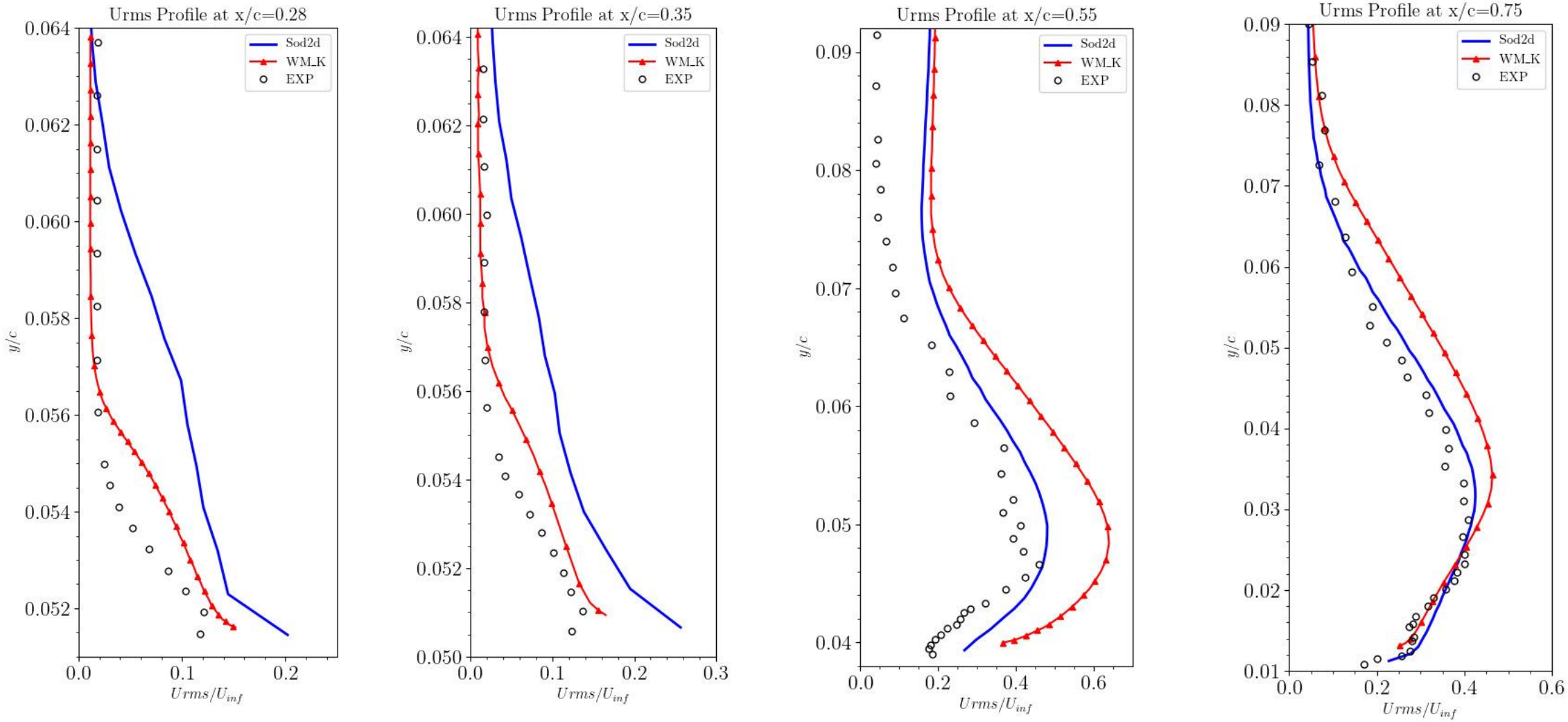
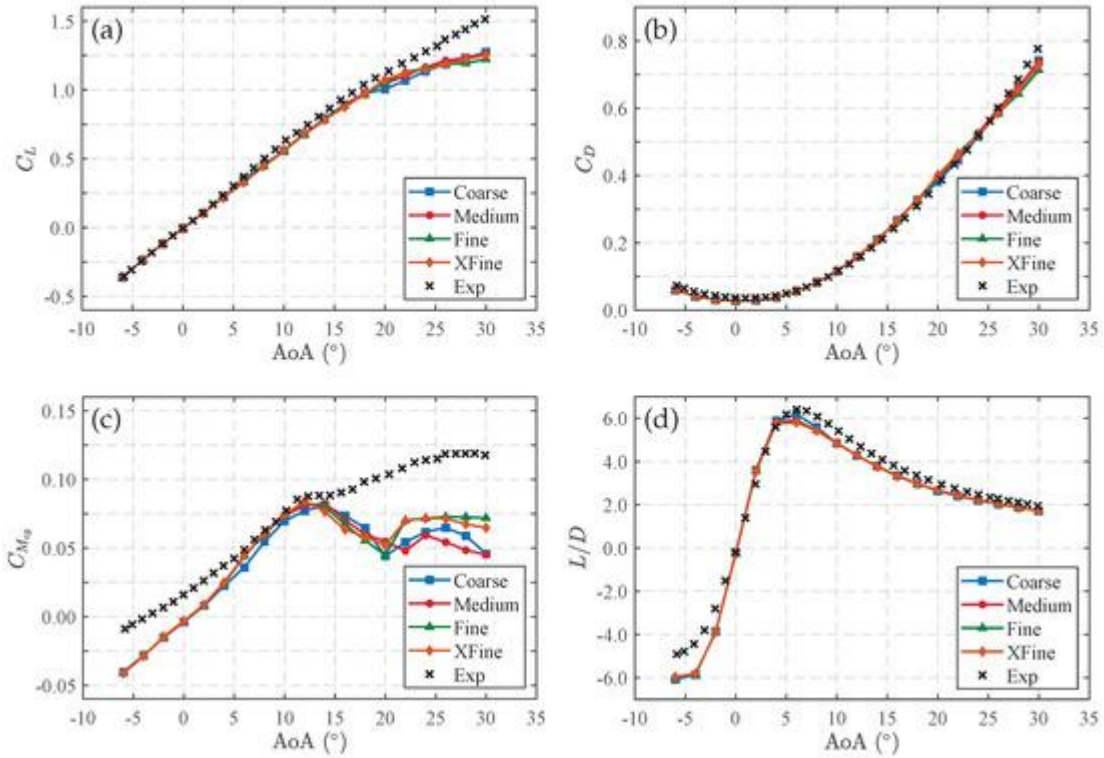
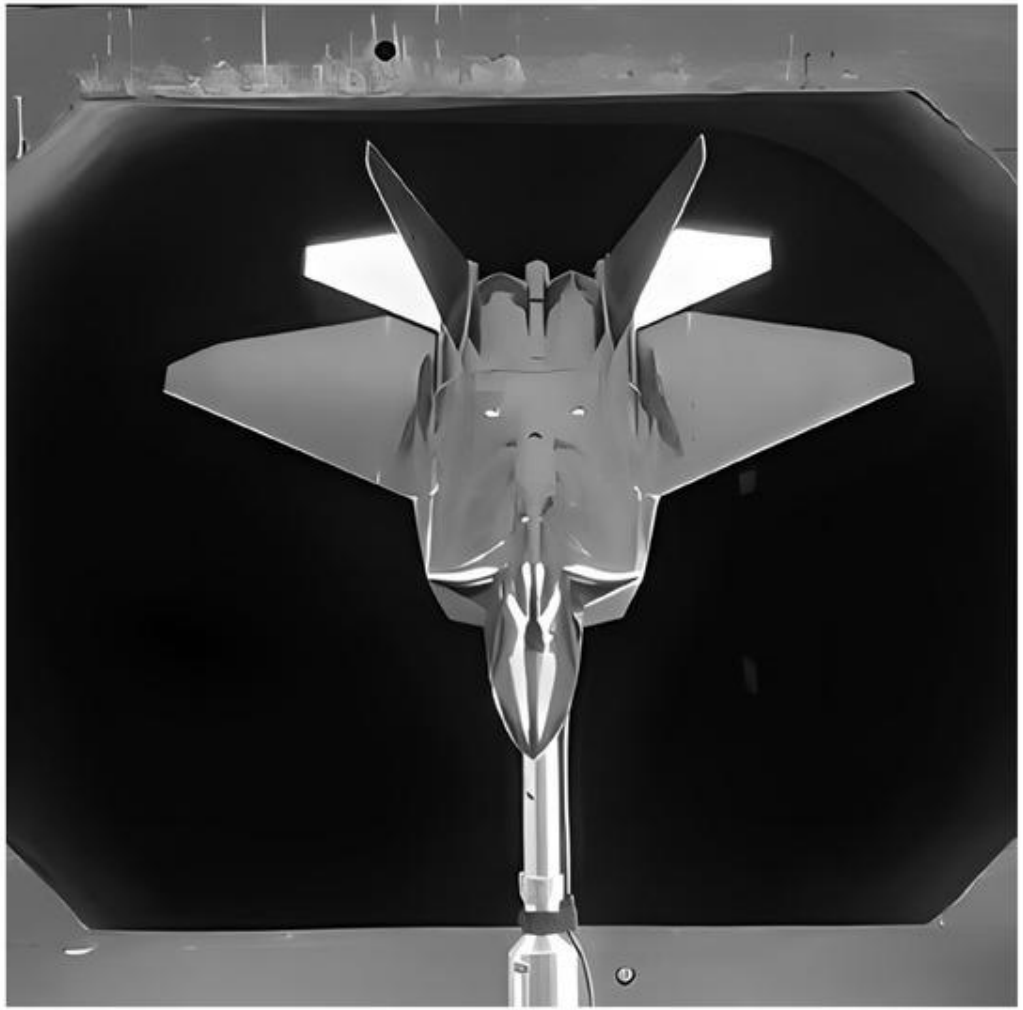


Figure 8: Time-averaged velocity fluctuations profiles .

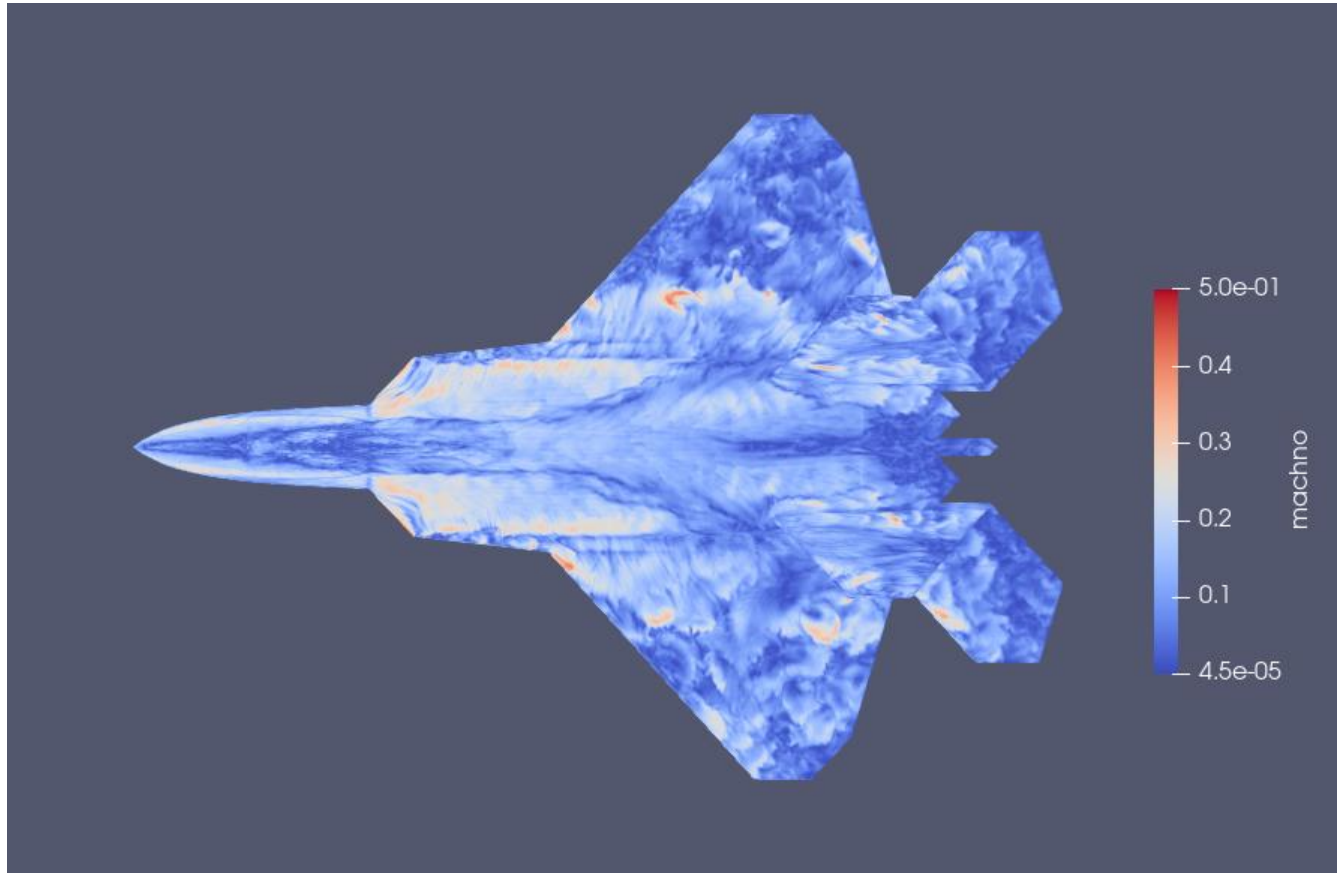
TIFON use case





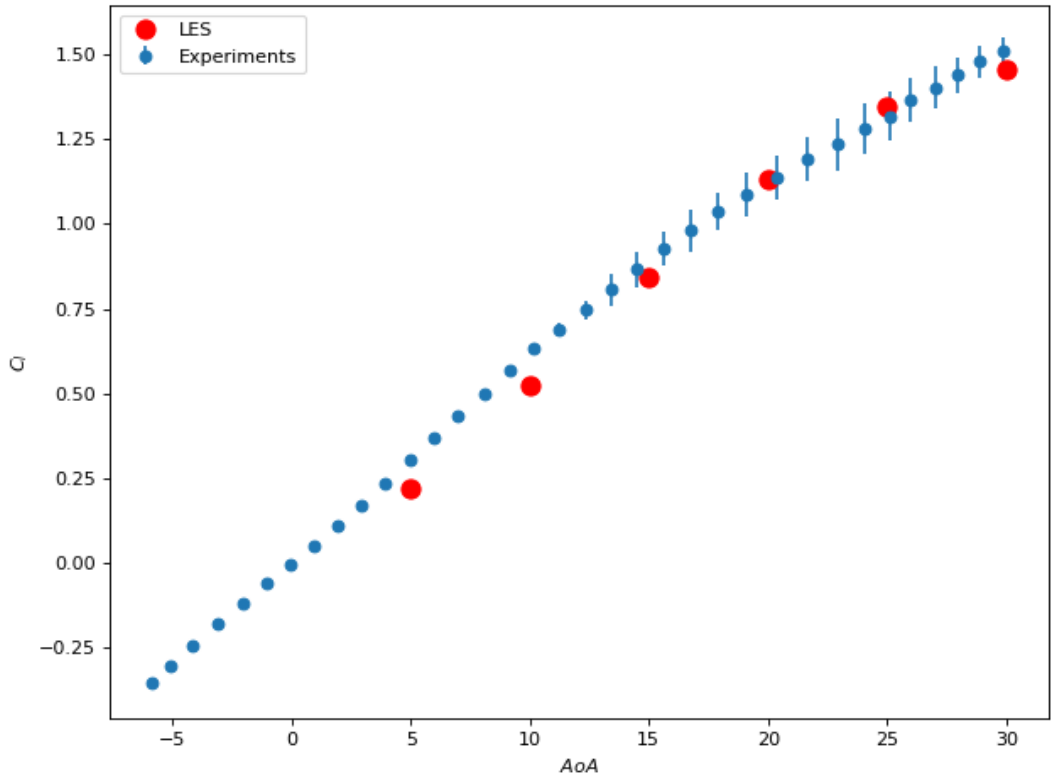
Giannelis, N.F.; Bykerk, T.; Vio, G.A. A Generic Model for Benchmark Aerodynamic Analysis of Fifth-Generation High-Performance Aircraft. *Aerospace* **2023**, *10*, 746. https://doi.org/10.3390/aerospace10090746

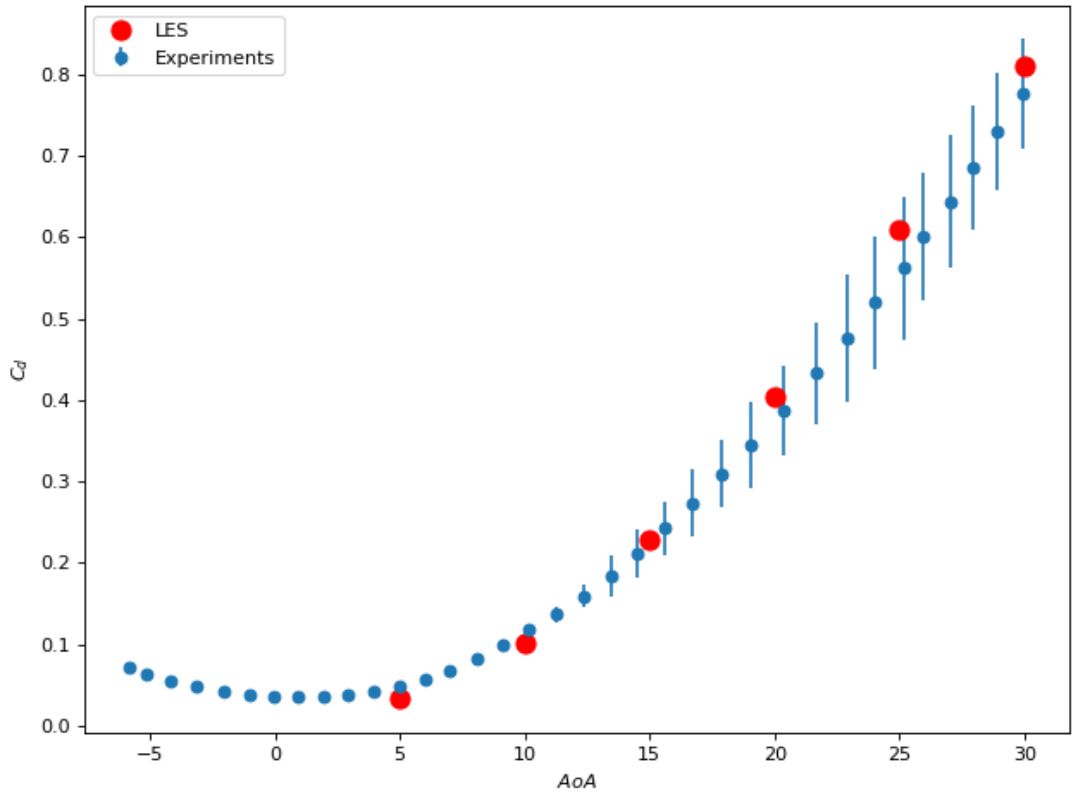
TIFON use case



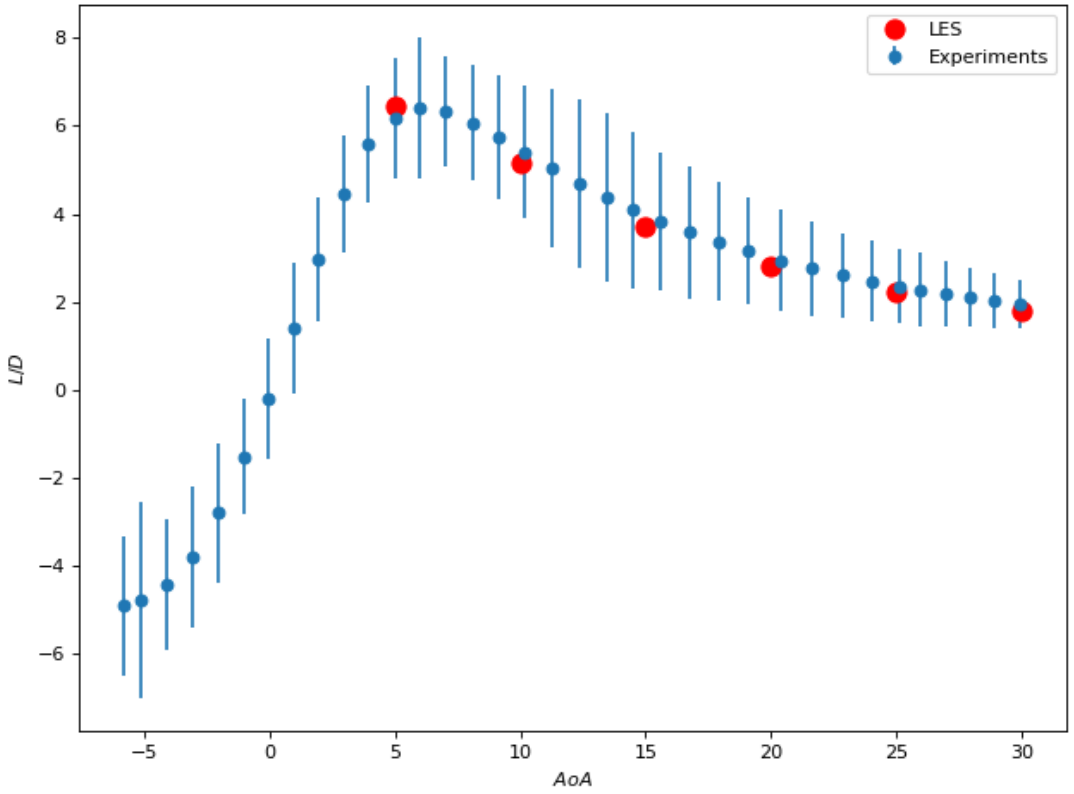
- Mach 0.2
- Reynolds 300k
- 20M DoF
- P2 SEM
- EQ wall model used

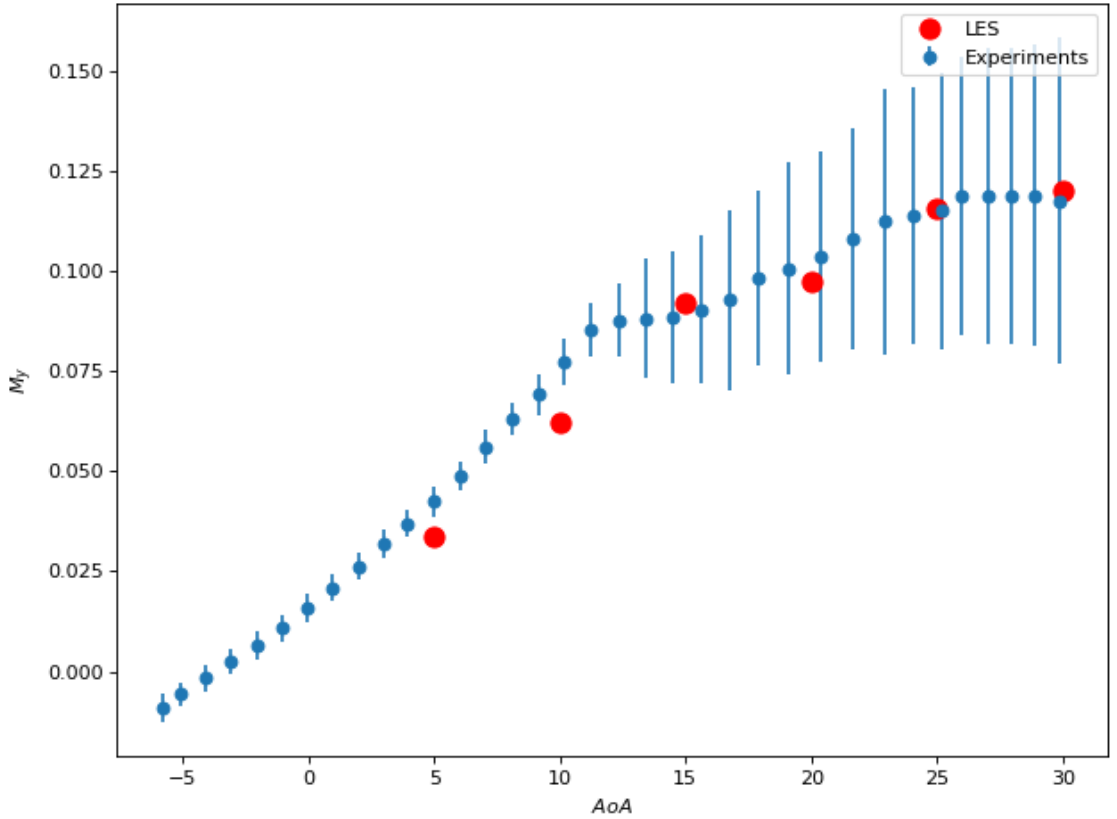
TIFON use case





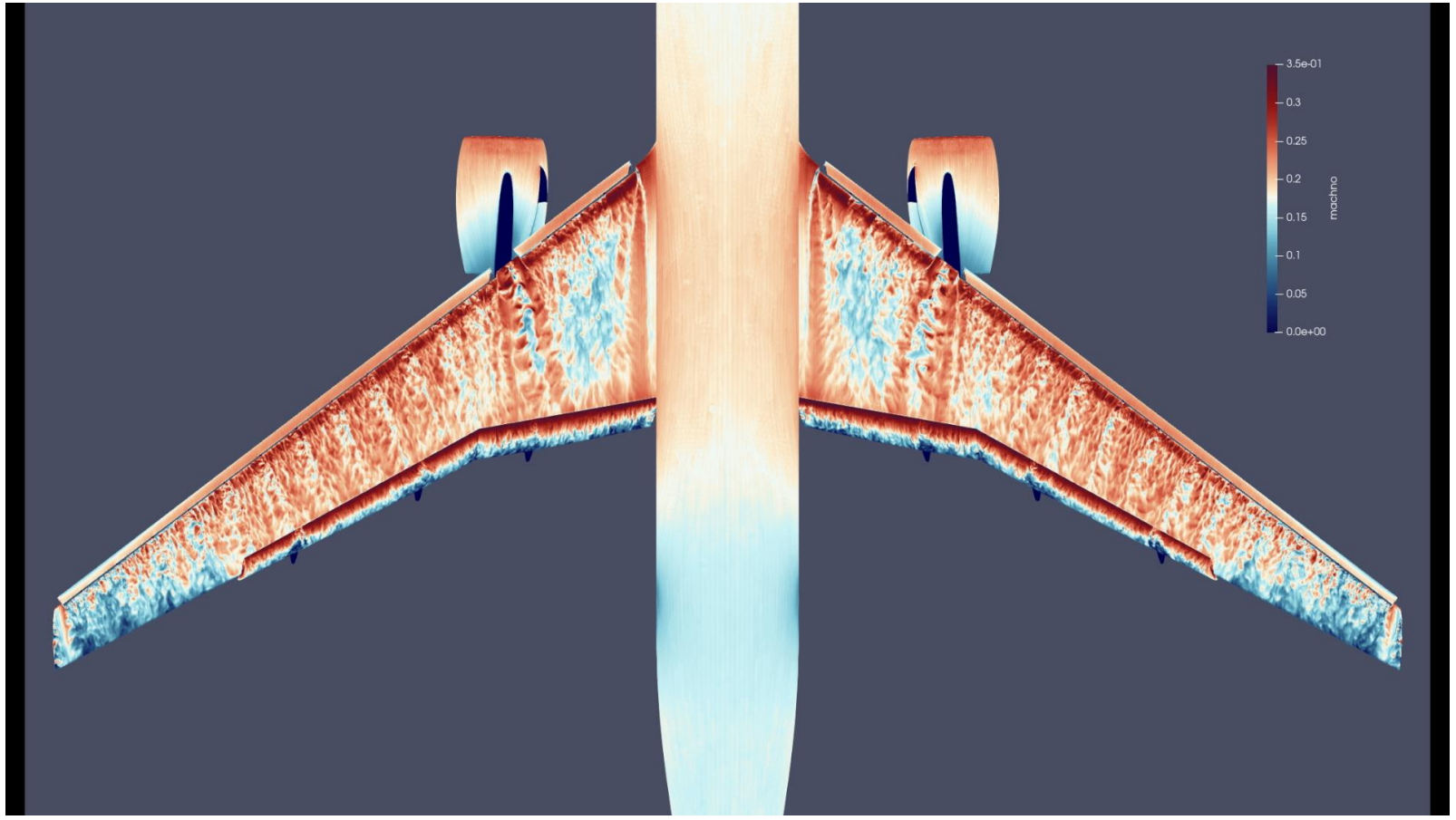
TIFON use case





NASA CRM workshop

HL geometry



ONERA HLPW5 TC2.4 WT

AoA	CL	CD
6.06	1.649874	0.1753716

LES: 175 M DoF (Re 5.4M)

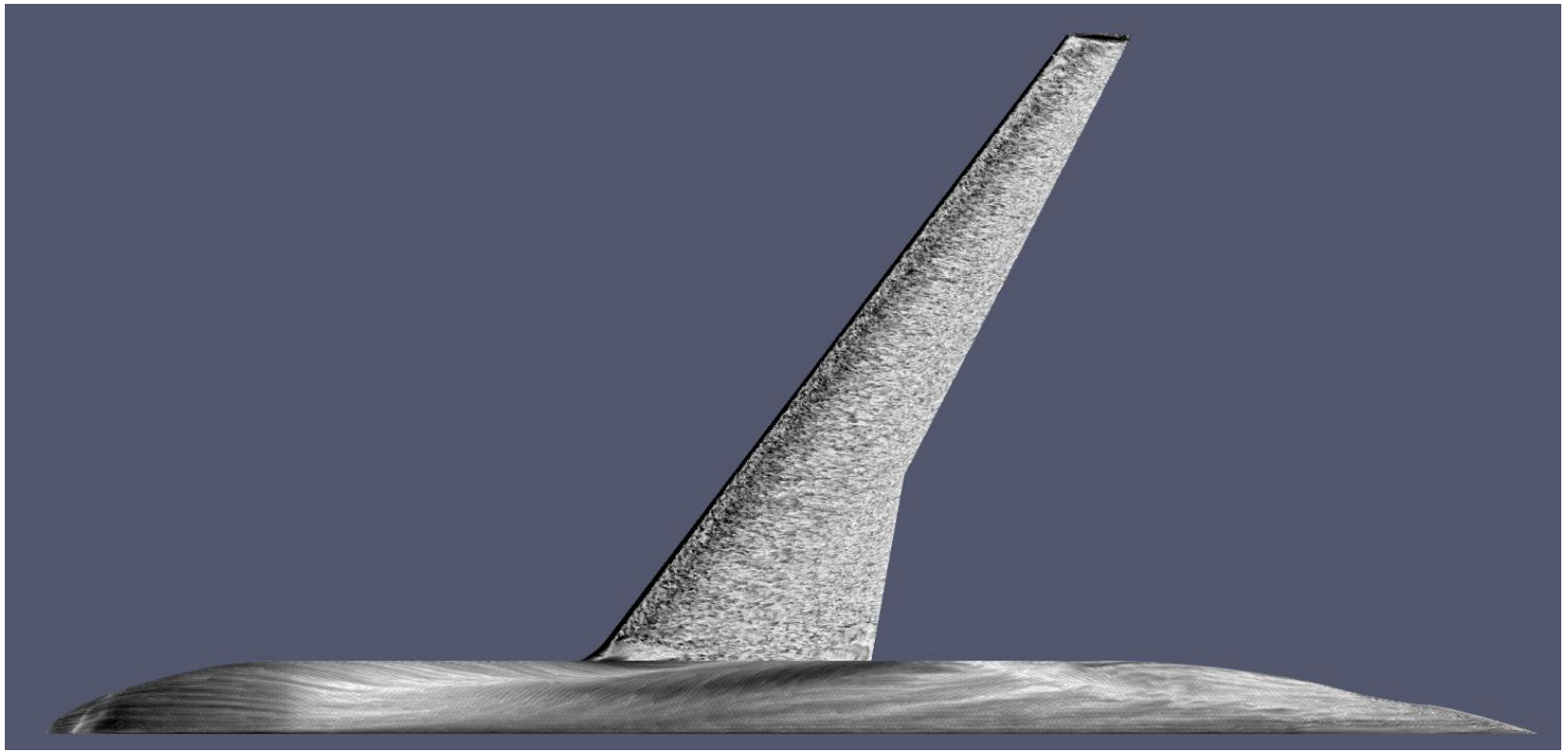
AoA	CL	CD
6.00	1.636332	0.177894

CODA: RANS Committee mesh(Re 5.4M)

AoA	CL	CD
6.00	1.636222	0.17129833

NASA CRM workshop

Clean geometry



LES: (Re 5.4M)

AoA	CL	CD
11.00	1.004293	0.061245568

CODA: RANS Committee mesh(Re 5.4M)

AoA	CL	CD
11.00	1.081780	0.064820905

Conclusions

- High-fidelity methodologies for large scale simulation of turbulent flows by means SEM has been presented.
- The methodologies has been applied to carry out scale resolving simulation of industry relevant problems and validated with experimental data.

Thanks!

erc

MINISTERIO DE CIENCIA E INNOVACIÓN

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

AGENCIA ESTATAL DE INVESTIGACIÓN