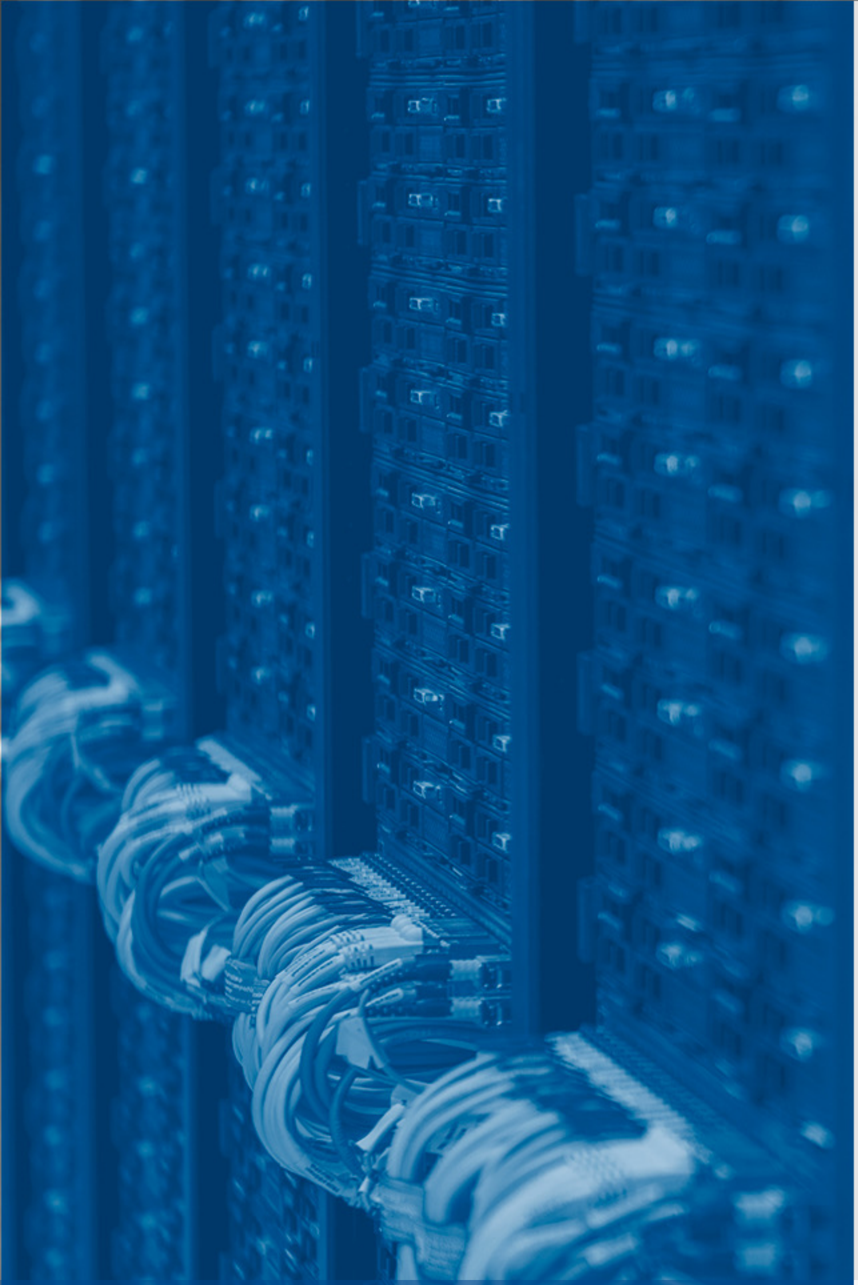


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CEEC
Centre of Excellence in Exascale CFD

Towards a framework to integrating CFD and ML in heterogeneous supercomputers

O.Lehmkuhl and B. Font

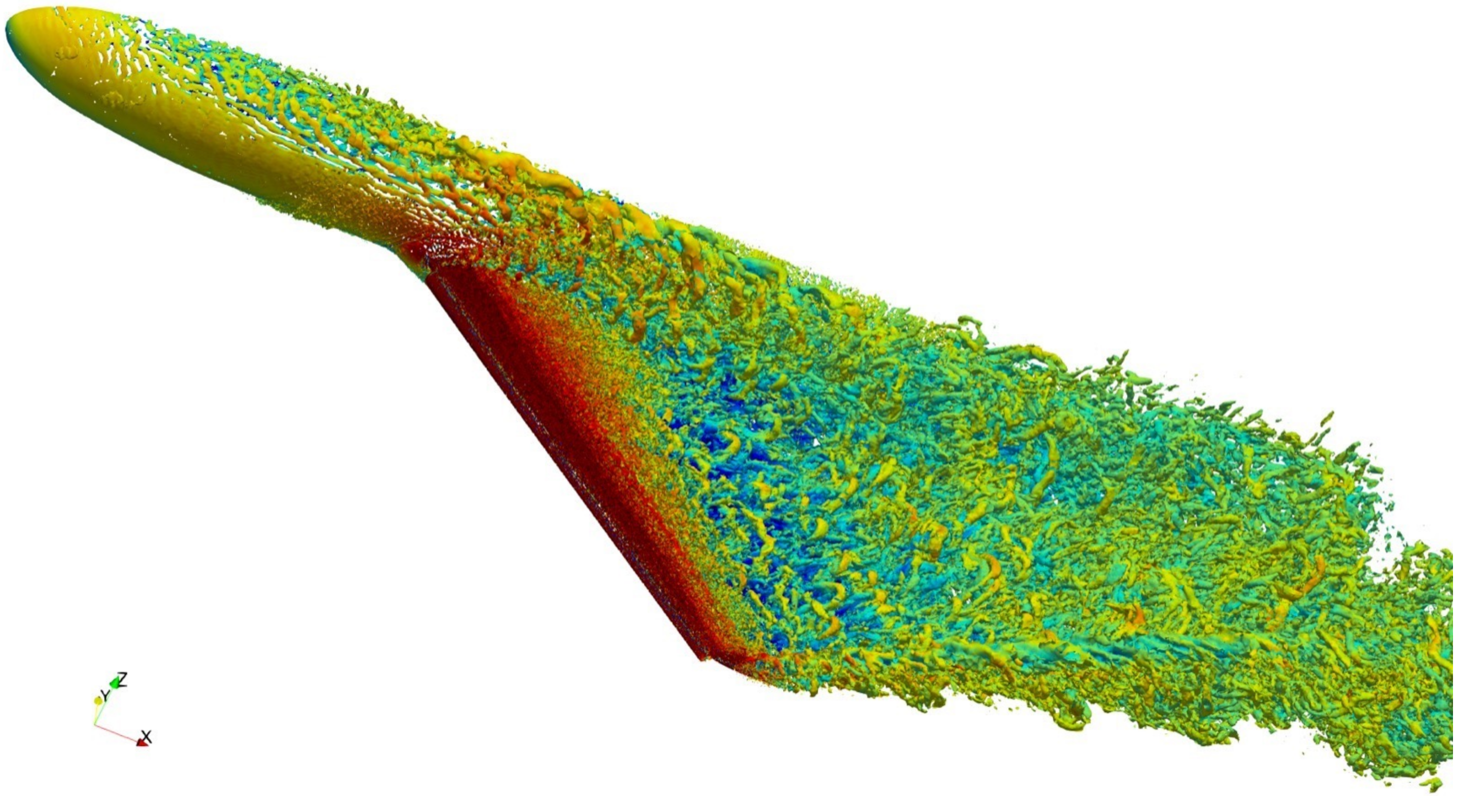
09/01/2024

eFlows4HPC workshop: HPC Workflows for Scientific Applications logistics

1. CEEC project

Some context

CEEC focuses on engineering, aeronautic and atmospheric engineering topics such as shock- boundary layer interaction and buffet on wings at the edge of the flight envelope, high fidelity aeroelastic simulation, topology optimization of static mixers.



Barcelona Supercomputing Center
Centro Nacional de Supercom

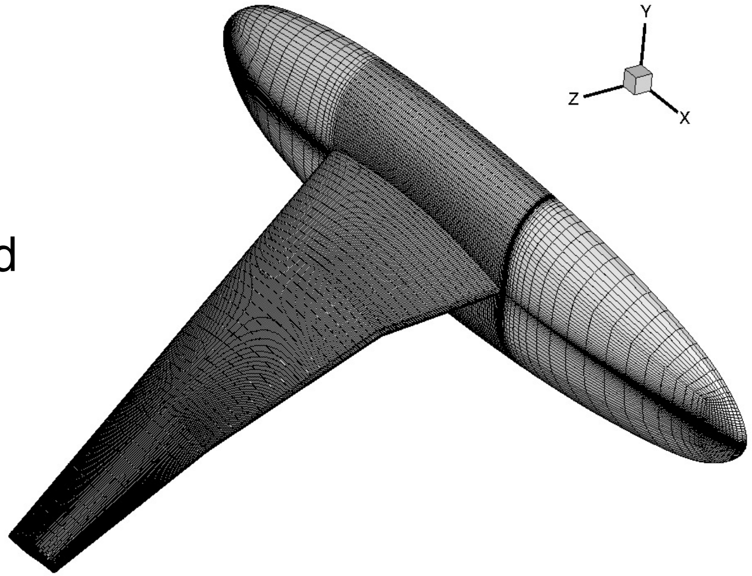
CEEC
Centre of Excellence in Exascale CFD

1. CEEC project

Some context

Description:

This LHC consists of an aeroelastic simulation of an elastic wing model in the transonic regime. This is a benchmark case derived from the HIRENASD Project (http://heinrich.lufmech.rwth-aachen.de/en). The wing configuration and the geometry is typical from large passenger transport aircraft and its dynamic test flight conditions are also equivalent to real-in-service cruise flight conditions.

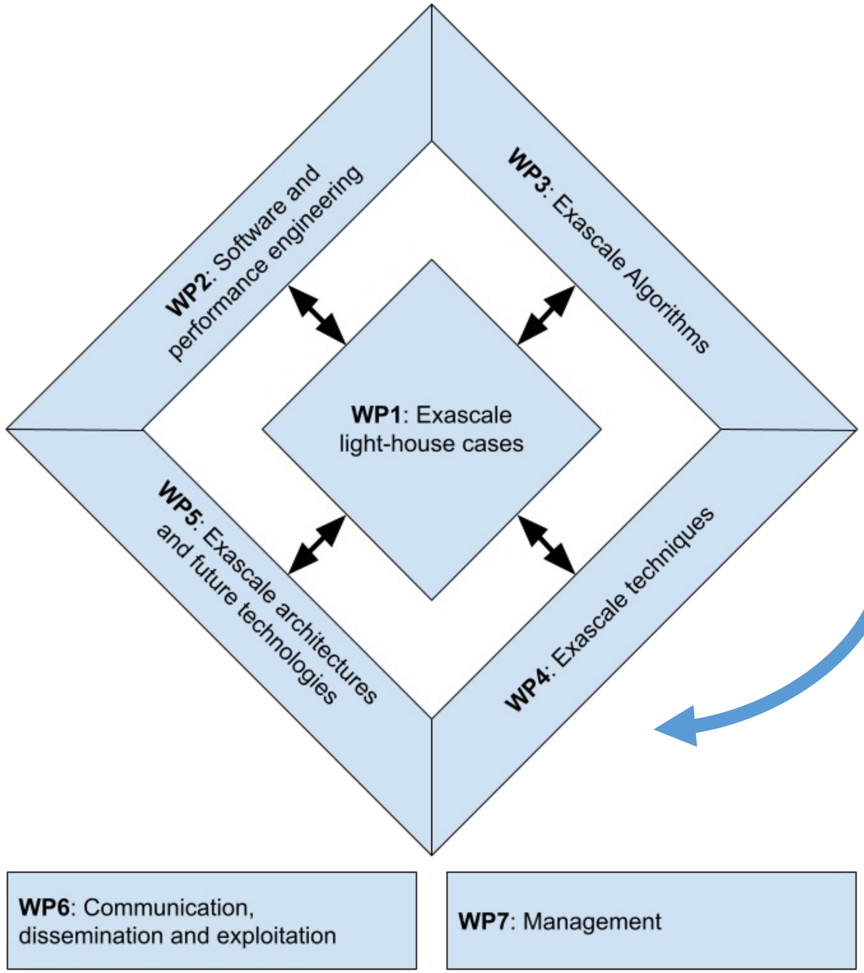


Challenges that are being addressed:

- Application of current LES turbulence models for aeroelastic cases with compressive flows under transonic regimes.
- The numerical model should be able to run in HPC clusters to obtain enhanced performance with low time-to-solution.
- Ensure efficient coupling strategies between the Flow and Solid solvers.

1. CEEC project

Some context



Task 4.2: ML-based sub-models:

Integrate ML models into the CFD workflow to accelerate physical models.

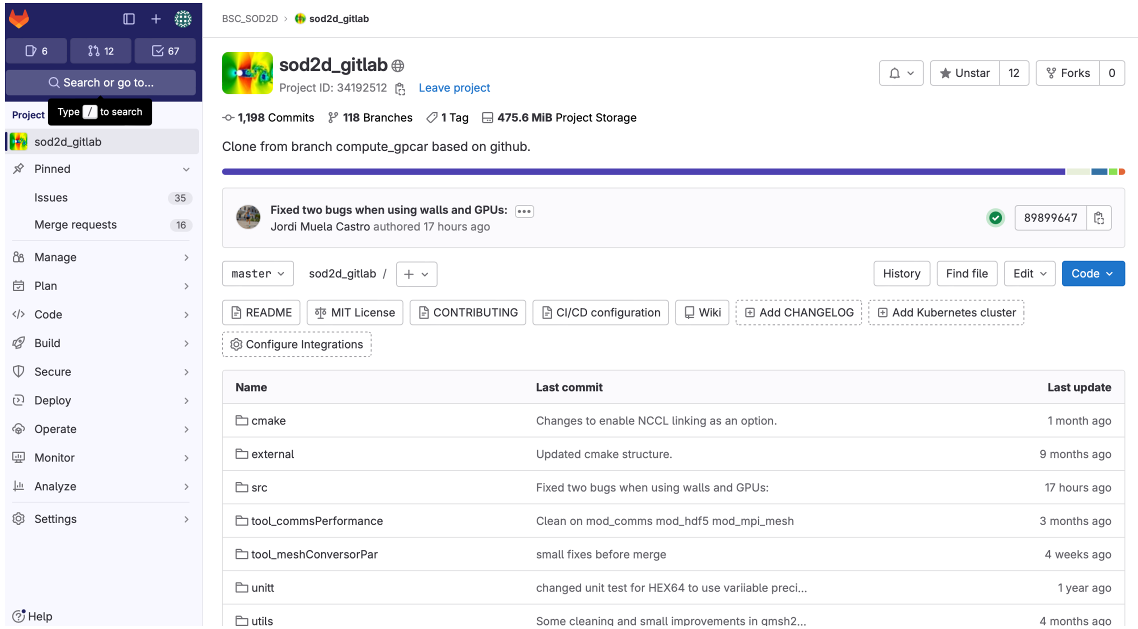
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CEEC
Centre of Excellence in Exascale CFD

2. Our CFD approach

SOD2D

- The result so far:



- Language: **Fortran**
- GPU port path: **OpenACC**
- Required libs: **HDF5, MPI**
- Git repo: https://gitlab.com/bsc_sod2d/sod2d_gitlab/
- … and btw, the code is **3D!**

SOD2D: **S**pectral high-**O**rder co**D**e **2** solve partial **D**ifferential equations

2. Physical model

- The compressible Navier-Stokes equations on a domain $\Omega\times(t_0, t_f)$:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\tau} = \mathbf{f}$$

$$\partial_t E + \nabla \cdot ((E + p)\mathbf{u}) - \nabla \cdot (\boldsymbol{\tau}\mathbf{u}) - \nabla \cdot (\kappa \nabla T) = \mathrm{S}$$

$$\boldsymbol{\tau} = \tau_{\boldsymbol{ij}} = \mu(T)\left((\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \frac{2}{3} \nabla \cdot \mathbf{u}\boldsymbol{I}\right)$$

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

2. Physical model

- Viscosity is calculated by means of Sutherland model:

$$\mu(T) = 1.458e^{-6} \frac{T^{1.5}}{T + 110.4}$$

- Additional required relations to close the system:

$$E = \rho \left[\frac{1}{2} (\mathbf{u} \cdot \mathbf{u}) + e \right] \qquad e = \frac{p}{\rho(\gamma - 1)}$$

$$p = \rho R T \qquad R = C_p - C_v \qquad \gamma = \frac{C_p}{C_v}$$

2. Physical model

By spatially filtering the NS equations:

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

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

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

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

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

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

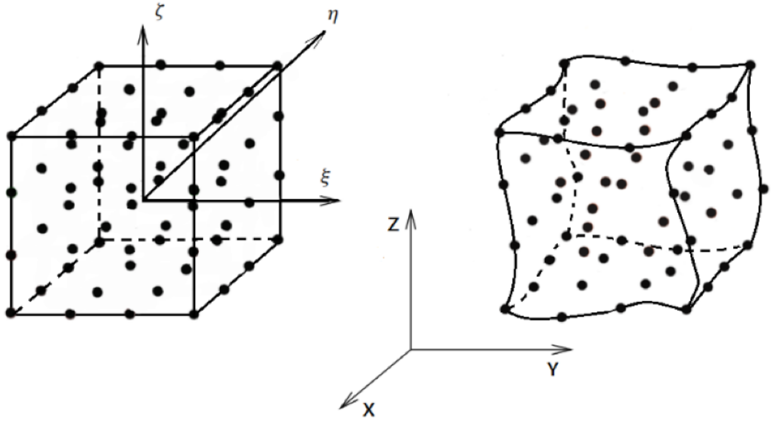
$$\beta_{ij} = \Delta_m^2 \alpha_{mi} \alpha_{mj}$$

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

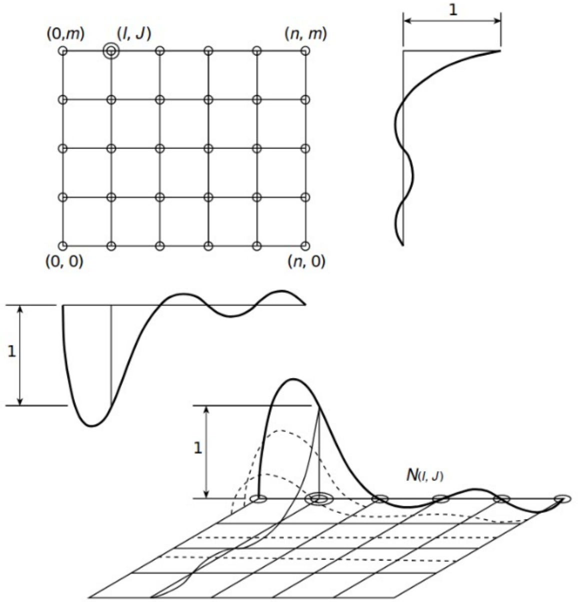
Specific challenges:

- **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**

3. Numerical model



- **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.
  - The **skew-symmetric convective operator split** detailed by Kennedy et al. [1] is employed, which **counters undesired *aliasing* effects**.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

[1] C. A. Kennedy and A. Gruber, **"Reduced aliasing formulations of the convective terms within the Navier-Stokes equations for a compressible fluid"**, Journal of Computational Physics, vol. 227, no. 3, pp. 1676–1700, 2008.

3. Numerical model

compressible flow

- We split the convective terms of momentum using the cubic kinetic energy preserving splitting:

$$\text{(Cubic split)} \quad \frac{1}{4}\left(\frac{\partial abc}{\partial x} + a\frac{\partial bc}{\partial x} + b\frac{\partial ac}{\partial x} + c\frac{\partial ab}{\partial x} + bc\frac{\partial a}{\partial x} + ac\frac{\partial b}{\partial x} + ab\frac{\partial c}{\partial x}\right)$$

- We split the convective terms of mass using the quadratic splitting:

$$\text{(Quadratic split)} \quad \frac{1}{2}\left(\frac{\partial ab}{\partial x} + a\frac{\partial b}{\partial x} + b\frac{\partial a}{\partial x}\right)$$

- In the case of the energy equation, we do the following modification:

$$\frac{\partial E}{\partial t} + \frac{\partial (E+p)u_j}{\partial x_j} = \frac{\partial E}{\partial t} + \frac{\partial \rho u_j (h + \frac{u_i u_i}{2})}{\partial x_j}$$

and then we apply cubic split to the $\dfrac{\partial \rho u_j \frac{u_i u_i}{2}}{\partial x_j}$ and $\dfrac{\partial \rho u_j h}{\partial x_j}$ independently.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Yuichi Kuya, Kosuke Totani, Soshi Kawai, Kinetic energy and entropy preserving schemes for compressible flows by split convective forms, Journal of Computational Physics, Volume 375, 2018, Pages 823-853, ISSN 0021-9991,

3. Numerical model

compressible flow

- Stabilisation using entropy viscosity concepts:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \eta \mathbf{u} \leq 0 \quad \Rightarrow \quad \eta = \frac{\rho}{\gamma - 1} \ln\left(\frac{p}{\rho^{\gamma}}\right) \quad \Rightarrow \quad R(\eta)_b = M_{bc}^{-1} C_{bc} f_b$$

$$\phi_a^e = \min(\beta_a^e, V_a^e) \quad \begin{cases} \beta^e = \frac{1}{2} ||\rho^e \left(\sqrt{\mathbf{u}^e \cdot \mathbf{u}^e} + c^e\right) ||_\infty \\ V_a^e = C_e \dfrac{||R(\eta)_a^e||_\infty \rho_a^e h_e^2}{||\eta_b - \overline{\eta}_b||_\infty} \end{cases}$$

$$\mu_a^e = \sum_{l=-1}^{1} \sum_{m=-1}^{1} \sum_{n=-1}^{1} \alpha_l \alpha_m \alpha_n \phi_{i+l,j+m,k+n}^e$$

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Jean-Luc Guermond, Richard Pasquetti, Bojan Popov, Entropy viscosity method for nonlinear conservation laws, Journal of Computational Physics, Volume 230, Issue 11, 2011, Pages 4248-4267

3. Numerical model
compressible flow

- Time integration using Runge-Kutta:

$$\phi^{\nu+1} = \phi^n + \alpha_\nu \Delta t M^{-1} \mathbf{R}^\nu,$$
$$\mathbf{R}^{\nu+1} = \mathbf{R}\left(\phi^{\nu+1}\right)$$

$$\phi^{n+1} = \phi^n + \Delta t \sum_{\nu=0}^{3} \beta_\nu \mathbf{R}^\nu$$

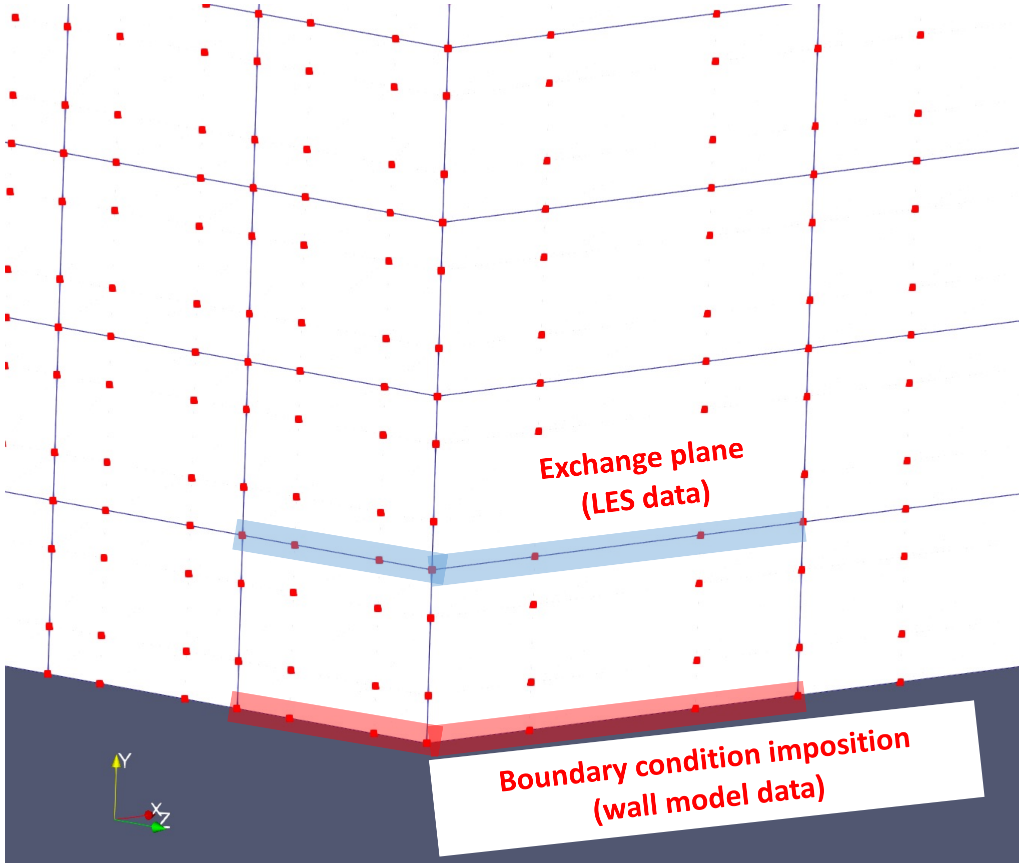
4th order RK tableu				
Coeff.	ν_0	ν_1	ν_2	ν_3
α_ν	0	1/2	1/2	1
β_ν	1/6	2/6	2/6	1/6

3. Numerical model

WMLES

First steps:

- Only algebraic wall models considered
- Two different strategies for the exchange location (interface between LES and the wall model):
  1. At a given position
  2. Just at the interface of the first off wall element
- After testing **strategy 2** is preferred
- Validations on benchmark cases
- Next steps are implementation of velocity transformations functions, and testing of NEQ wall models.



3. Numerical model

WMLES

What we would like in CEEC

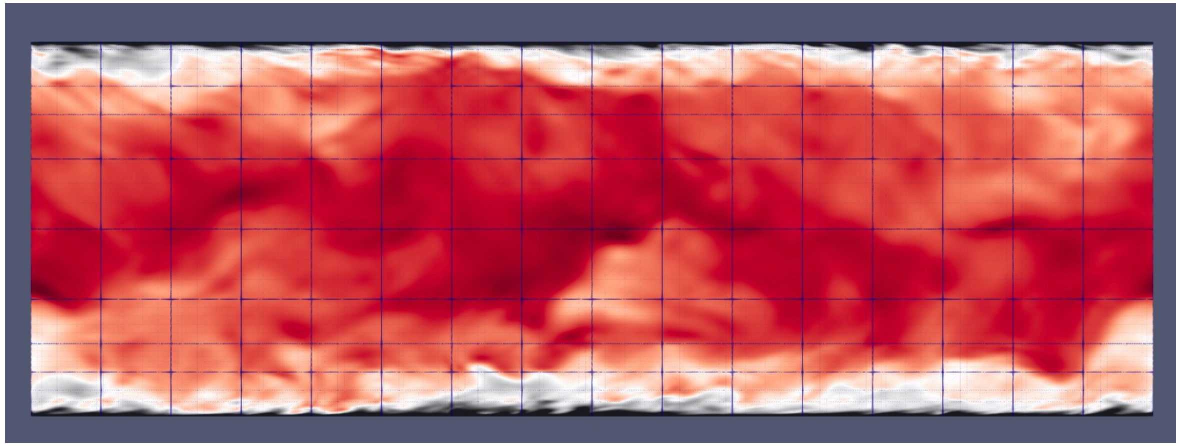
- To replace the analytical wall model by a data-driven one based on RL
- Possible advantages:
  1. Introduce numerical errors on the model optimisation
  2. No need of high-fidelity data sets to do the training
- Bottlenecks:
  1. Really intensive from the computational point of view
  2. Not available workflows for RL at very large scale

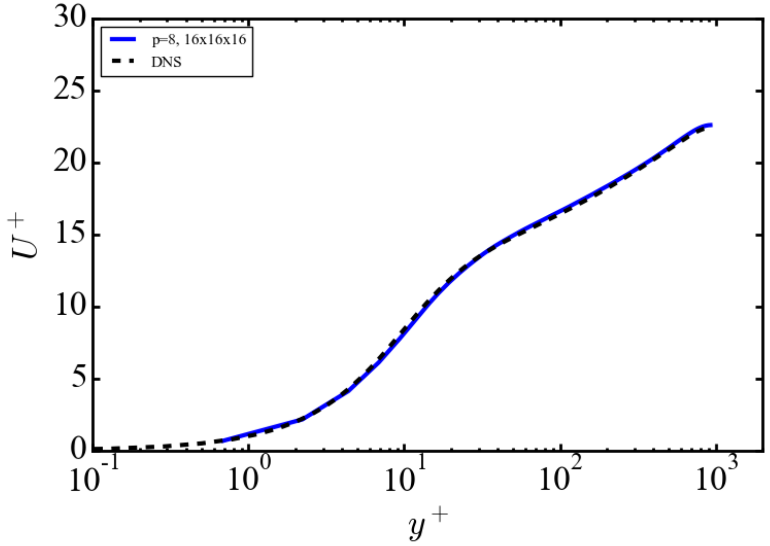
- **As a first step, a simpler flow control problem will be tested to demonstrate the workflow.**

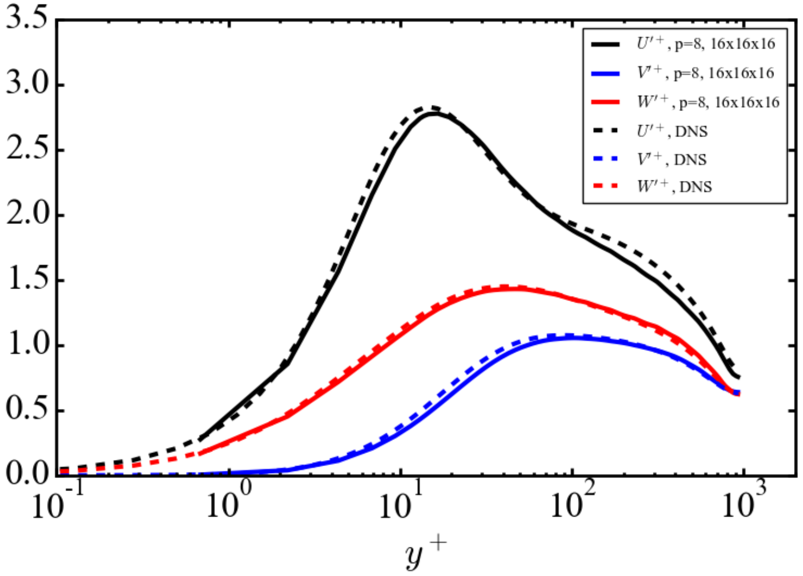
4. Code validation

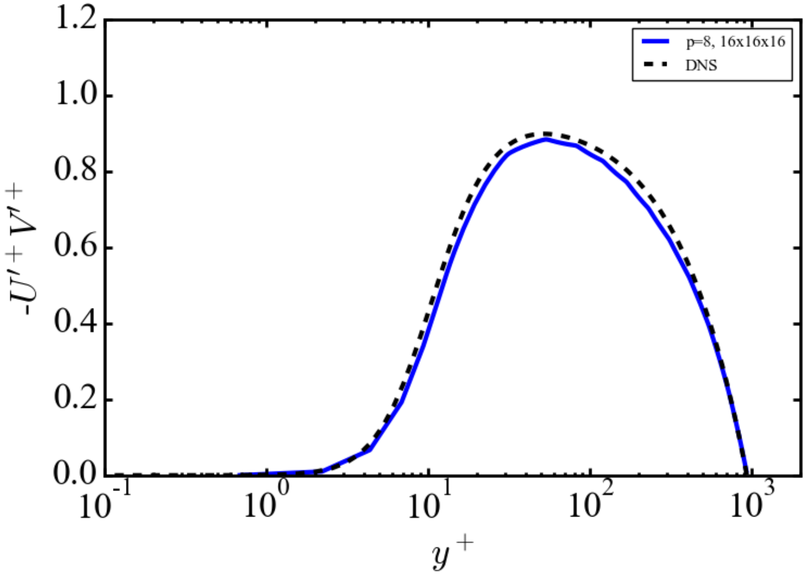
Channel flow ($Re_\tau = 950$ & $\mathrm{Ma} = 0{,}1$) , compressible

```
P = 8
tw  0.0025879371628599007
theory tw  0.002589595812175917
err tw % 0.06405050966708589
utau  0.050871771768436574
Re_tau  949.6957113466683
Vreman SGS
```







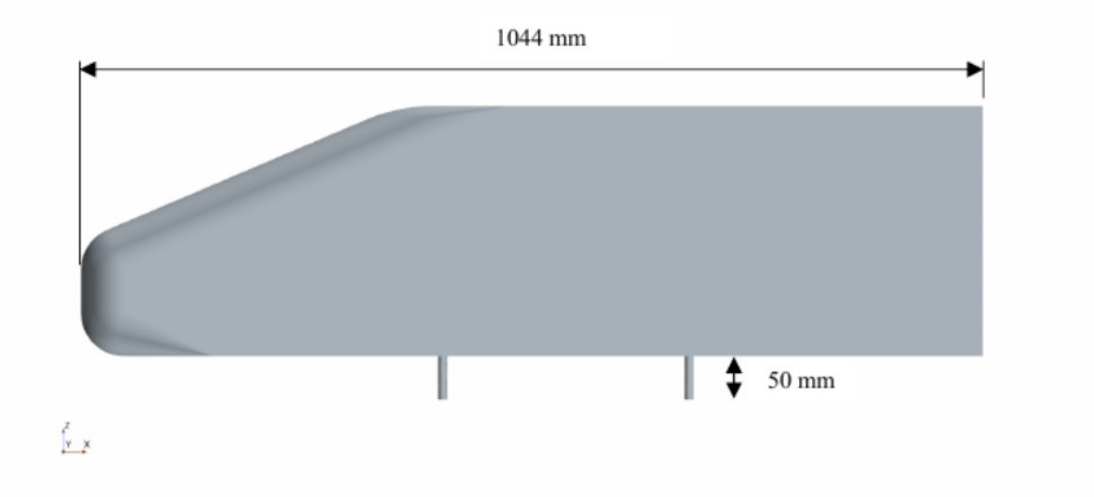


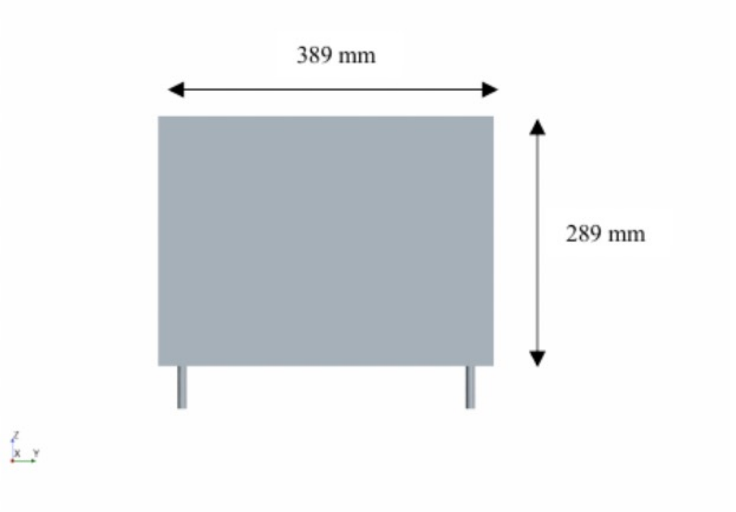
4. Code validation

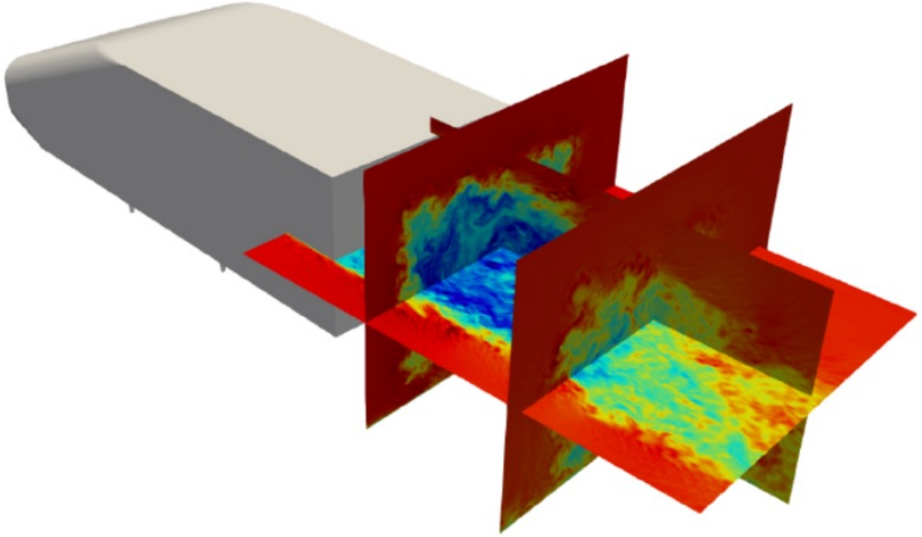
Windsor body, compressible, p4 and WMLES+Vreman

Case definition

- 2.5º of yaw
- Re = 2.9 x 10^6
- U = 40 m/s
- No wheels configuration
- No slip road





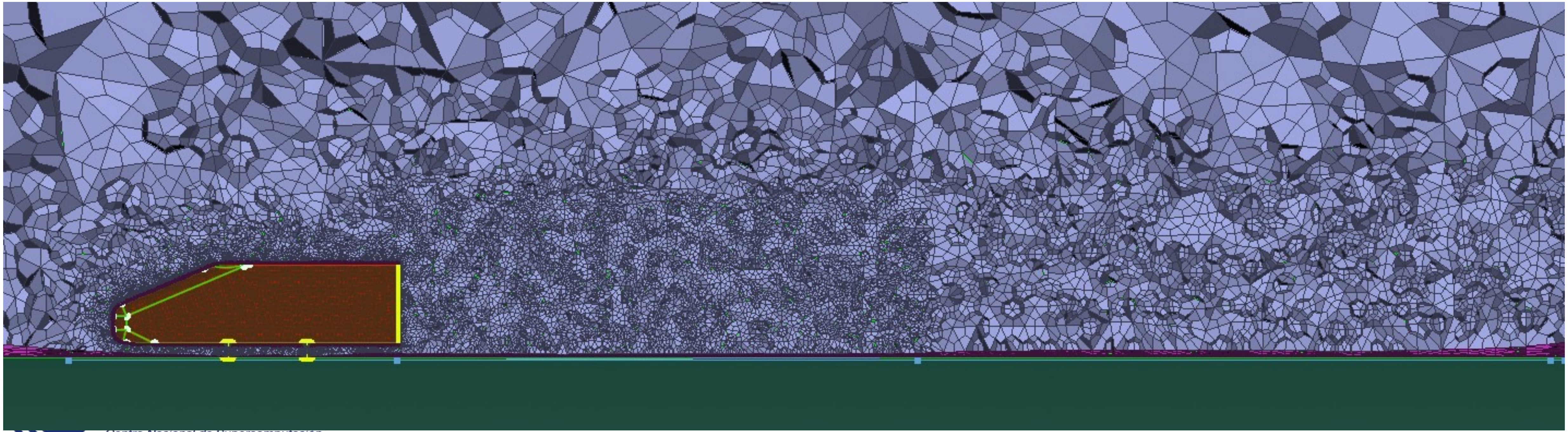


Barcelona Supercomputing Center
Centro Naci

4. Code validation

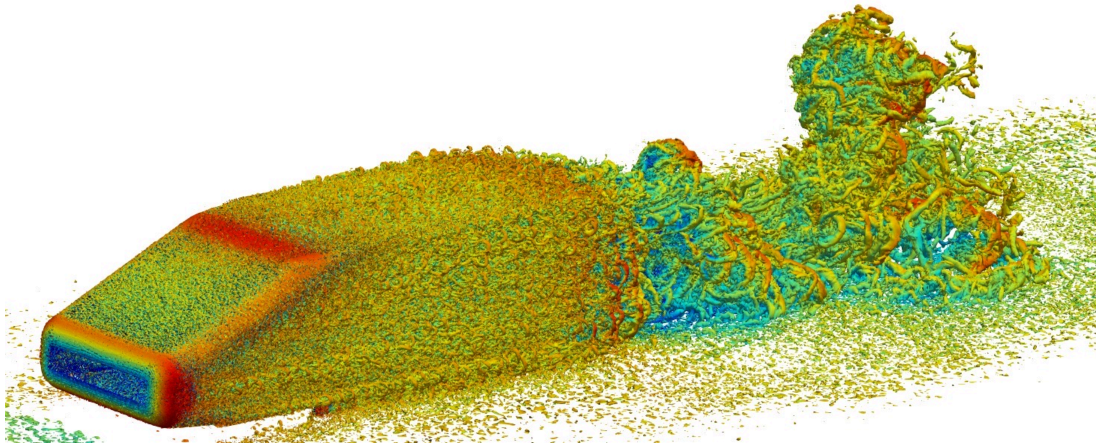
Windsor body, compressible, p4 and WMLES+Vreman

- Two meshes considered based on unstructured hexaedra (p4):
  - Coarse: 13 M DOF
  - Fine: 150 M DOF
- Vreman SGS model used
- Equilbrium wal model (exchange location at off-wall node 3)



4. Code validation

Windsor body, compressible, p4 and WMLES+Vreman



	Lift	Drag
Exp. From AUTOCFD3	**-0.0382**	**0.3298**
Present-coarse	-0.0552	0.3485
Present-fine	-0.0983	0.3247

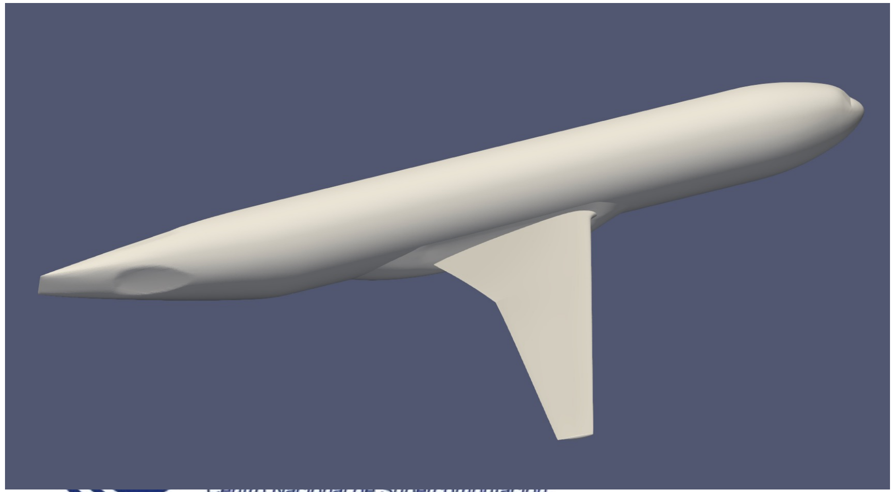
4. Code validation

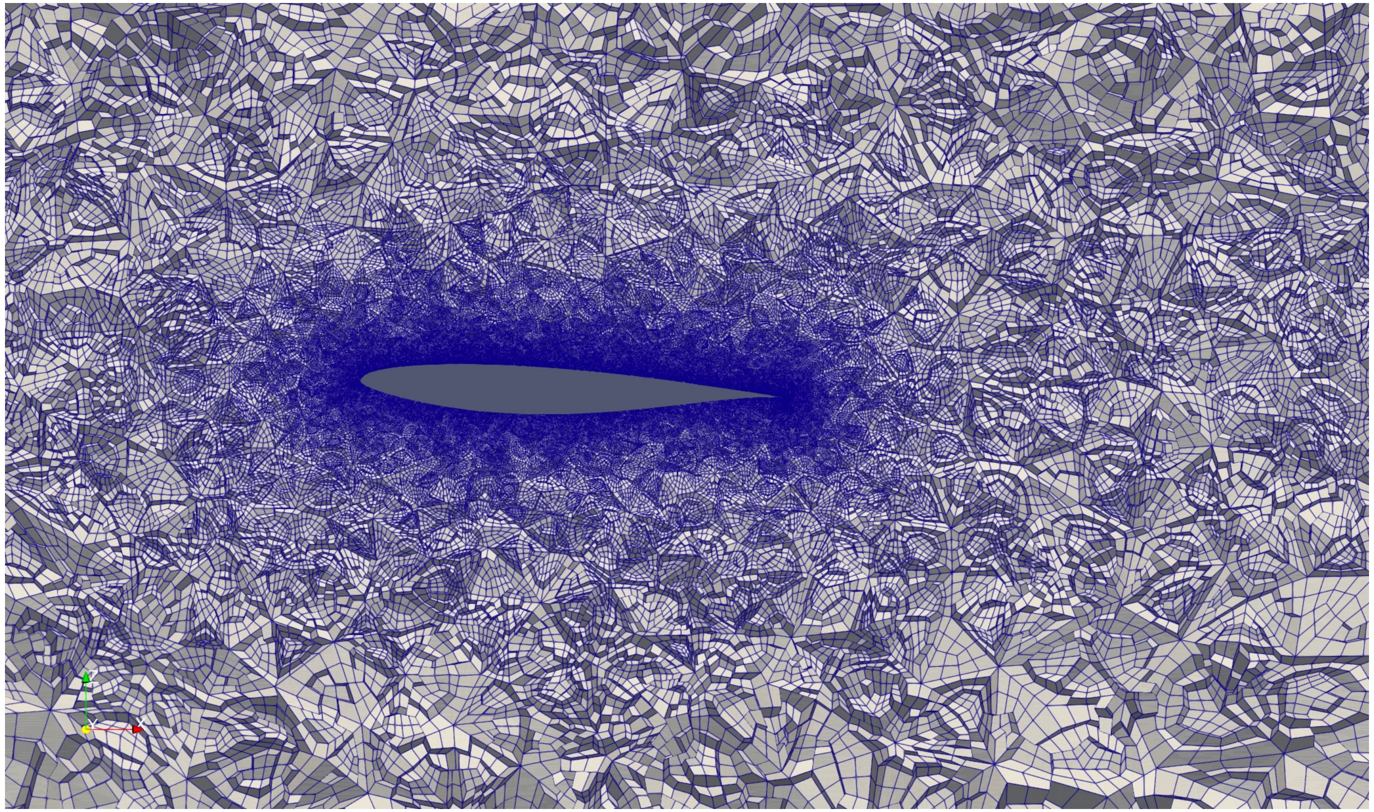
CRM HL Case1, compressible, 150M DOF p2 and WMLES+Vreman

Case Parameters and Requirements

Geometry	CRM-HL-WB
Mach Number	0.20
Chord Reynolds Number	5.6×10^6
Angle of Attack	11°
Reference Static Temperature	521 °R

Important Details	• Geometry is provided in full-scale inches
	• When using a dimensional code, it is recommended to adjust viscosity to a non-physical value to match requested Reynolds number
	• All simulations are "free air" only
	• When using RANS:
	  ○ $\hat{\nu}_{farfield}/\nu_{ref} = 3$ for SA-based models
	  ○ Adiabatic wall BC (not isothermal)
	  ○ SA-neg-QCR2000-R is recommended; coefficient for rotation correction (-R) should be changed to C_{rot}=1 for verification (standard value of C_{rot}=2 can be used as an "optional" case)

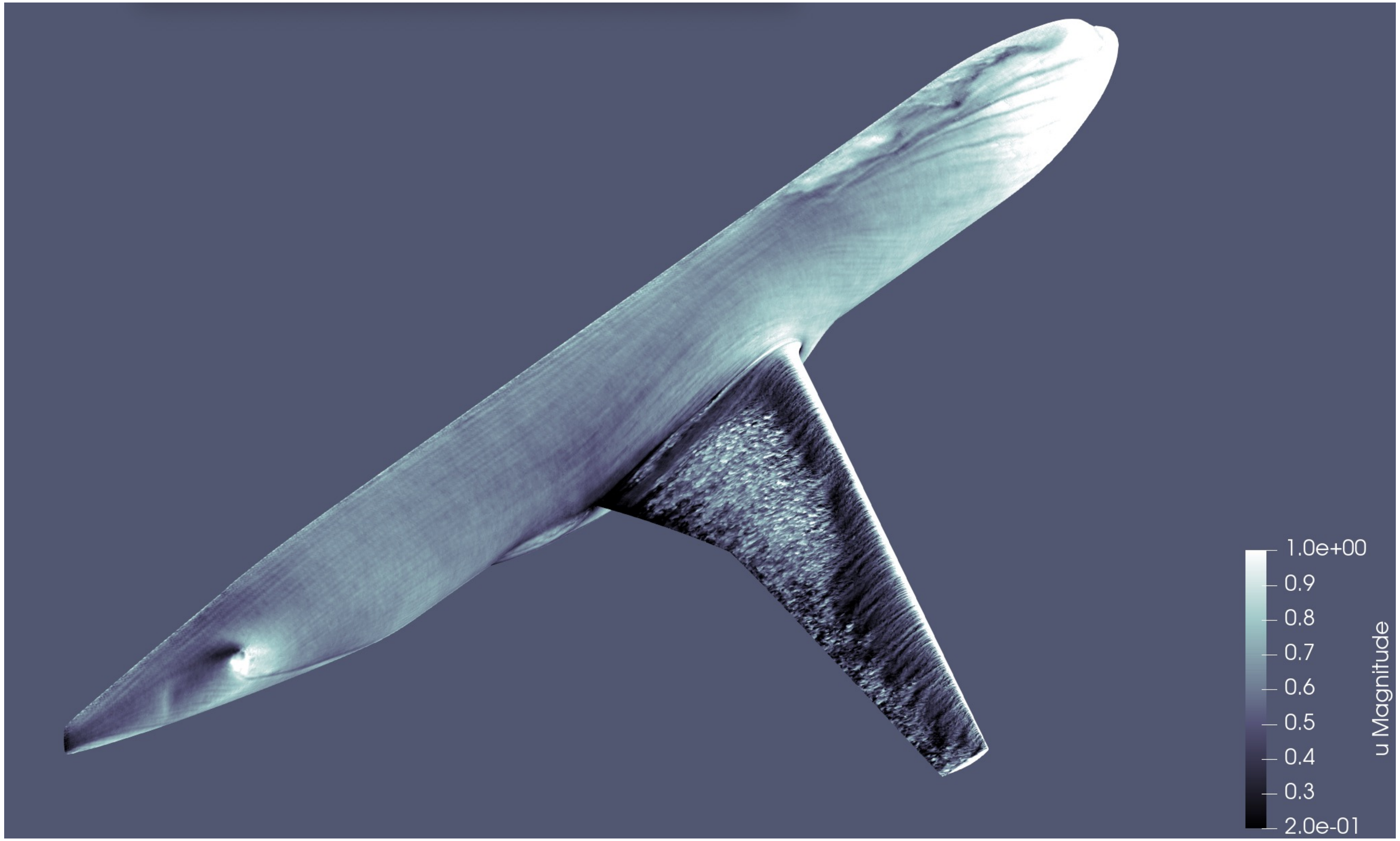




https://hiliftpw.larc.nasa.gov/Workshop5/Documents/HLPW5_Test_Cases_v1.7.pdf

4. Code validation

CRM HL Case1, compressible, 150M DOF p2 and WMLES+Vreman



- Very early
- Turbulence is developping on the wing
- Where is our BIG GPU machine!? ☺

https://hiliftpw.larc.nasa.gov/Workshop5/Documents/HLPW5_Test_Cases_v1.7.pdf

4. Code validation

Early dissemination

- Paper submitted to: Computer Physics Communications
- First reviews available , is positive, we expect to have the paper in the upcoming months.
- **Focus on the main kernels' performance on GPU vs CPU:**

Timings for explicit 85^3 runs			
	full step	diffu kernel	convec kernel
H100	123.246	7.681	13.099
A100	182.116	13.304	16.33
V100	443.504	24.837	22.949

- Times in (ms)
- p3 elements
- Single-precision runs using CUDA managed memory
- Explicit RK4 time-advance scheme
- MN4 reference: 1.68s/step (48 cores, 85**3 mesh)

ELSEVIER
Computer Physics Communications
Volume 297, April 2024, 109067

Computational Physics

SOD2D: A GPU-enabled Spectral Finite Elements Method for compressible scale-resolving simulations ☆

L. Gasparino [a], F. Spiga [b], O. Lehmkuhl [a]

Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/j.cpc.2023.109067

Get rights and content

Abstract

As new supercomputer architectures become more heavily focused on using hardware accelerators, in particular general-purpose graphical processors, it is therefore relevant that algorithms for computational fluid dynamics, especially those targeting scale-resolving simulations, be designed in such a way as to make efficient use of such hardware.

4. Code validation

Early dissemination

- Externals users are already playing SOD2D: UPC, KTH and Agronne are the most notable.
- First examples of external applications:

ANL-23/59

Turbulence Modeling with Nek5000/RS, SOD2D and Alya

Vishal Kumar[1], Oriol Lehmkuhl[2], Ananias Tombouldies[3], Paul Fischer[1,4,5], and Misun Min[1]

[1] Mathematics and Computer Science Division, Argonne National Laboratory
[2] Barcelona Supercomputing Center (BSC)
[3] Department of Mechanical Engineering, Aristotle University of Thessaloniki
[4] Department of Computer Science, University of Illinois Urbana-Champaign
[5] Mechanical Science & Engineering, University of Illinois Urbana-Champaign

September 25, 2023

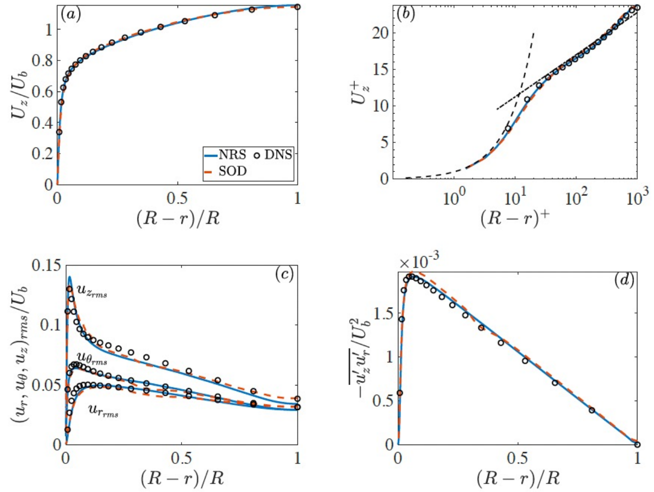


Figure 15: Comparison of turbulent pipe flow results between NekRS and SOD2D at $Re_\tau = 1000$ on the Fine grid resolution; *(a)* axial velocity (U_z/U_b) in outer units; *(b)* axial velocity in inner units; *(c)* normal turbulent stresses; *(d)* resolved Reynolds stress.

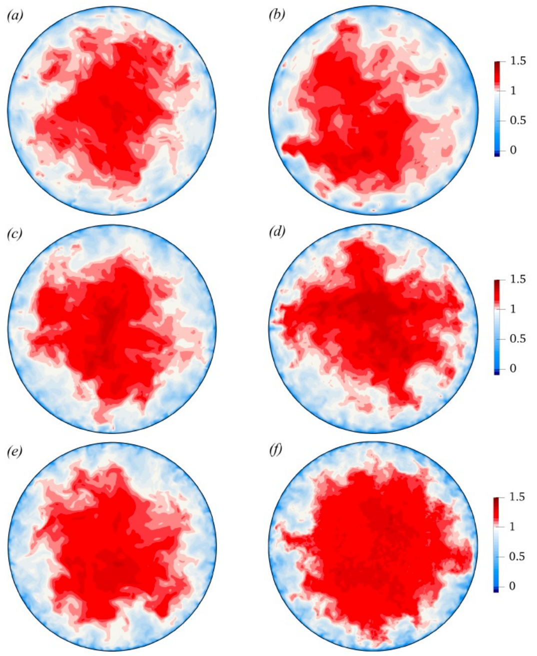


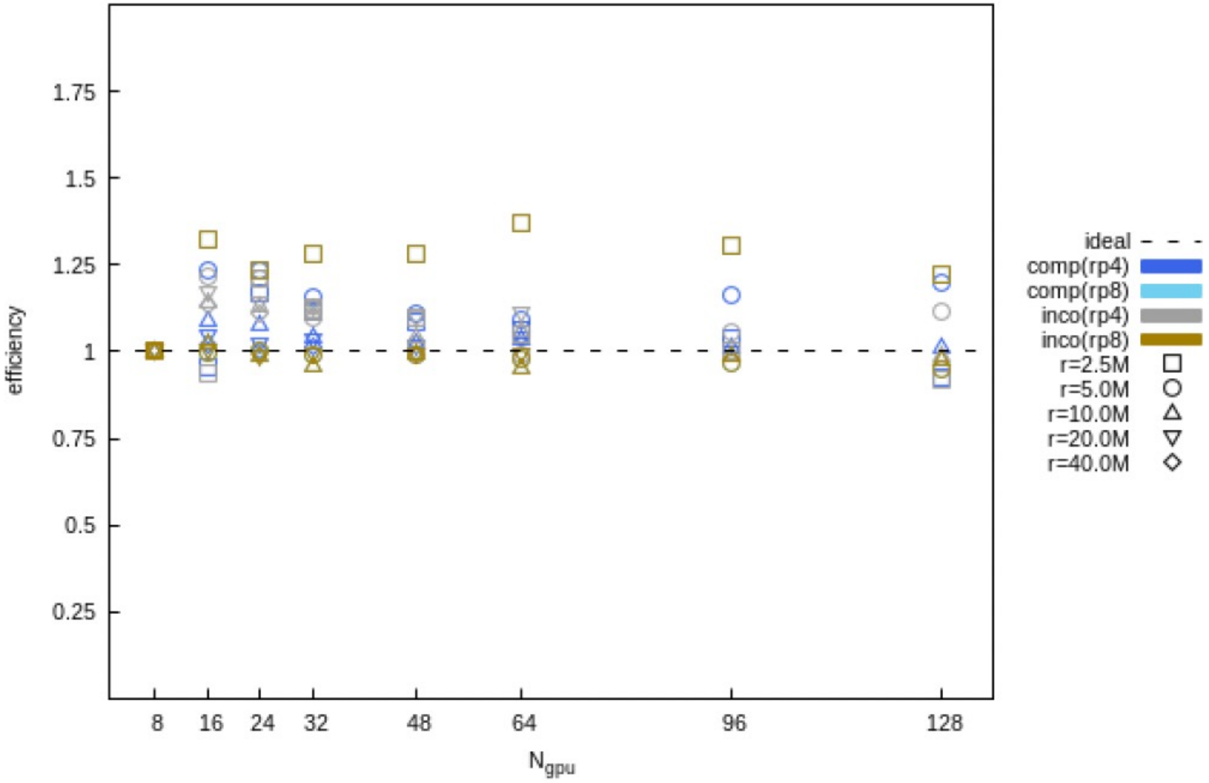
Figure 14: Visualization of instantaneous axial velocity (u_z/U_b) for pipe flow at $Re_\tau = 1000$; *(a,c,e)* Nek5000; *(b,d,f)* SOD2D; *(a,b)* Coarse resolution; *(c,d)* Medium resolution; *(e,f)* Fine resolution; mesh resolution are indicated in Table 3.

Table 2: Performance of NekRS and SOD2D on Polaris. Tests have been performed for Channel flow at a resolution of $384 \times 384 \times 384$ with $E = 64^3$, $N = 6$, and the total number of grid points of 56 millions. Δt = 1.5e-03 ($CFL = 1.7$) and Δt = 6.5e-05 ($CFL = 1.5$) are used for NekRS and SOD2D, respectively.

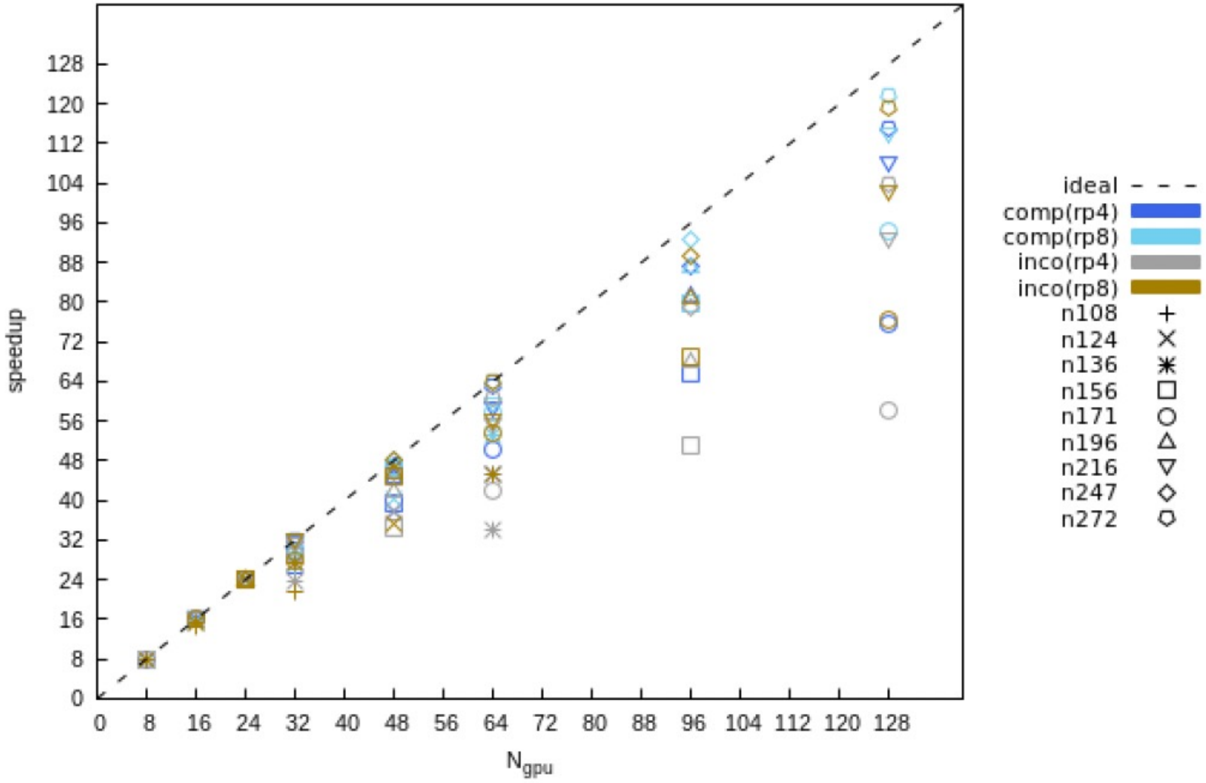
	Performance on Polaris						
Code	nodes	GPUs	dofs/GPU	v_i	p_i	t_{step} (s)	P_{eff}
Nek5000	2	8	7.0779e+06	2.97	2.13	2.3512e-01	100
	3	12	4.7186e+06	2.97	2.14	1.7690e-01	88.6
	4	16	3.5389e+06	2.97	2.18	1.4138e-01	83.1
	5	20	2.8312e+06	2.97	2.09	1.2734e-01	73.8
	6	24	2.3593e+06	2.97	2.16	1.1305e-01	69.3
	8	32	1.7695e+06	2.97	5.16	1.3139e-01	44.7
	9	36	1.5729e+06	2.97	2.13	9.4724e-02	55.1
SOD2D	2	8	7.0779e+06	-	-	3.4603e-01	100
	3	12	4.7186e+06	-	-	2.4769e-01	93.1
	4	16	3.5389e+06	-	-	1.9941e-01	86.7
	5	20	2.8312e+06	-	-	1.6249e-01	85.1
	6	24	2.3593e+06	-	-	1.3526e-01	85.2
	8	32	1.7695e+06	-	-	1.0868e-01	79.5
	9	36	1.5729e+06	-	-	9.6119e-02	80.0

5. Scalability analysis

- Weak speedup:



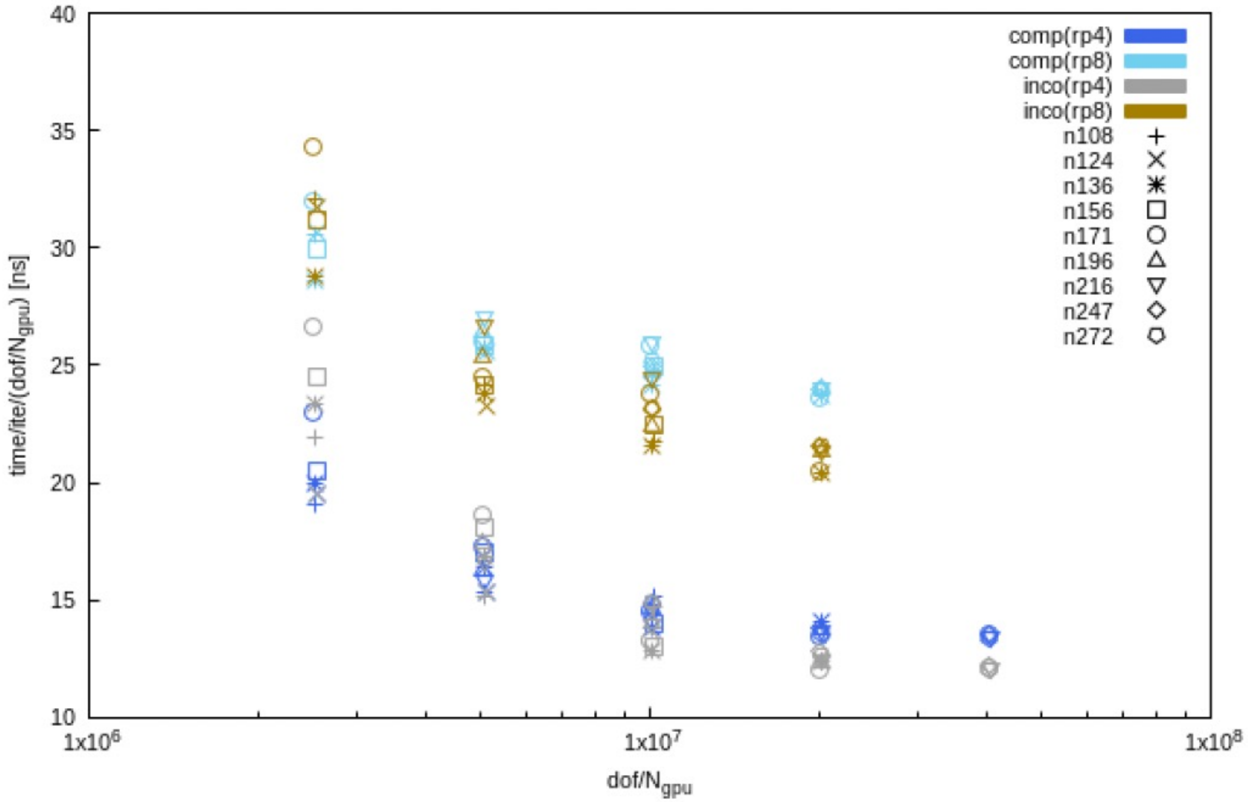
- Strong speedup:



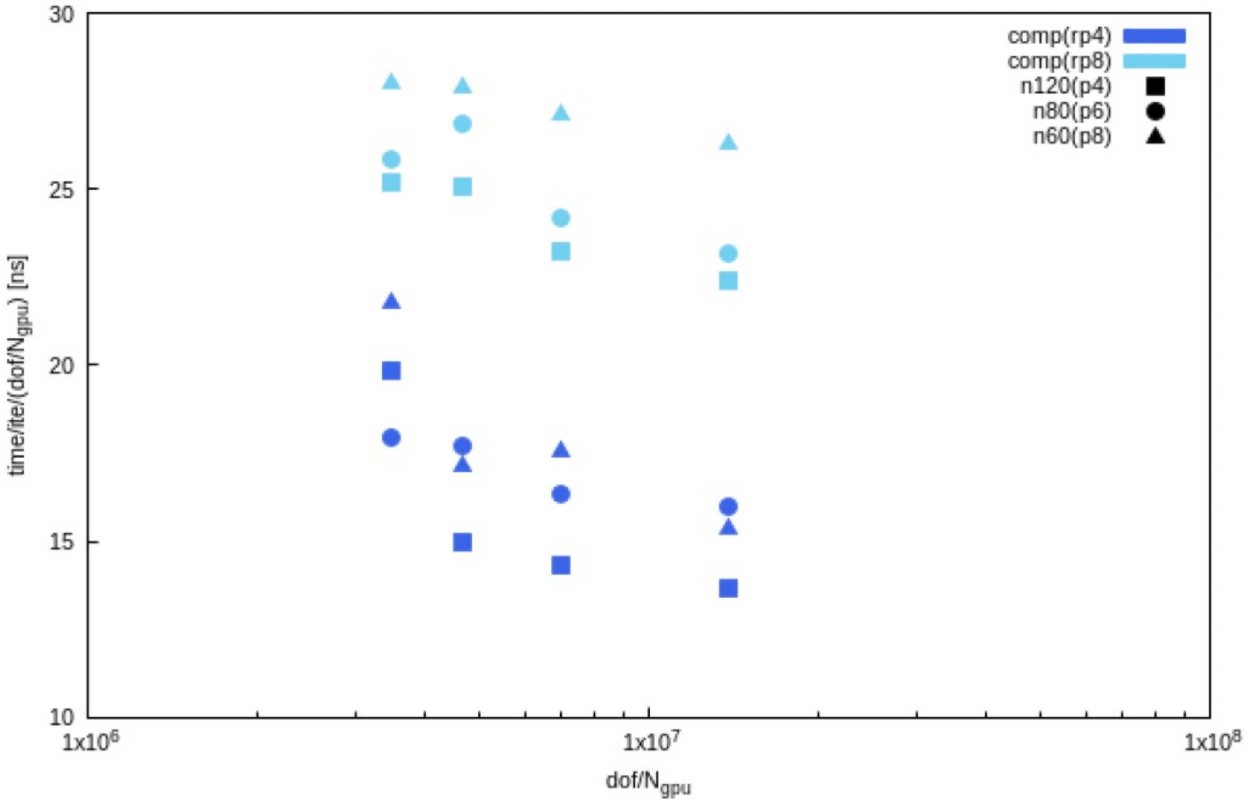
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

5. Scalability analysis

- Time per iteration vs GPU load:

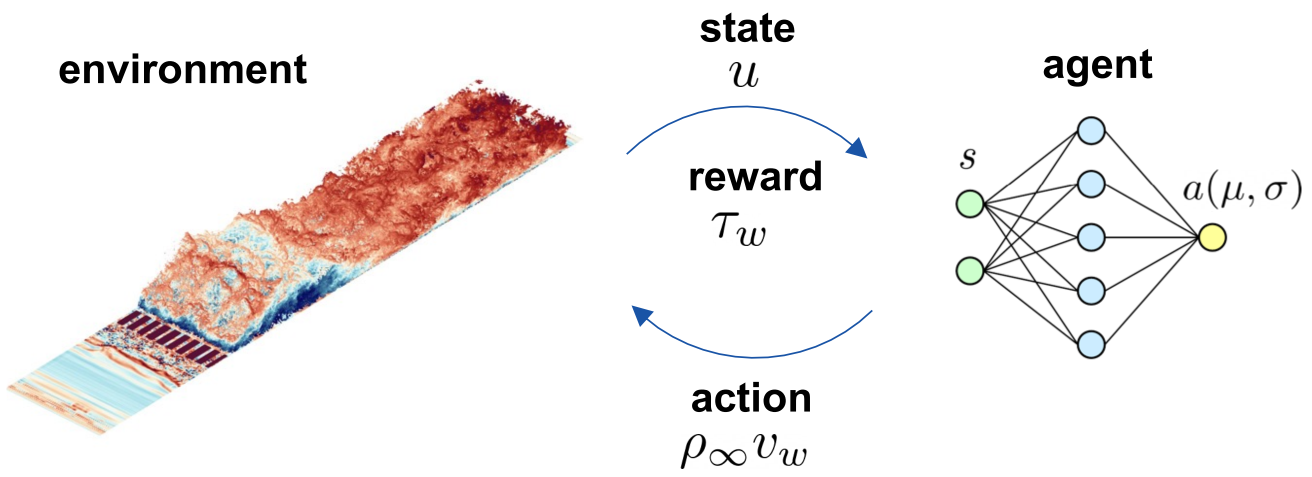


- Time per iteration vs GPU load (p-order):



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

6. ML integration



trajectory (or episode)

$$\{(s^0, r^0, a^0), (s^1, r^1, a^1), ..., (s^n, r^n, a^n)\}$$

General training approach:
1. Collect trajectory via noisy-sampled actions (exploration)
2. Optimize agents weights to maximize accumulated reward in time
3. Check agent performance with deterministic action (most probable, μ)
4. Repeat

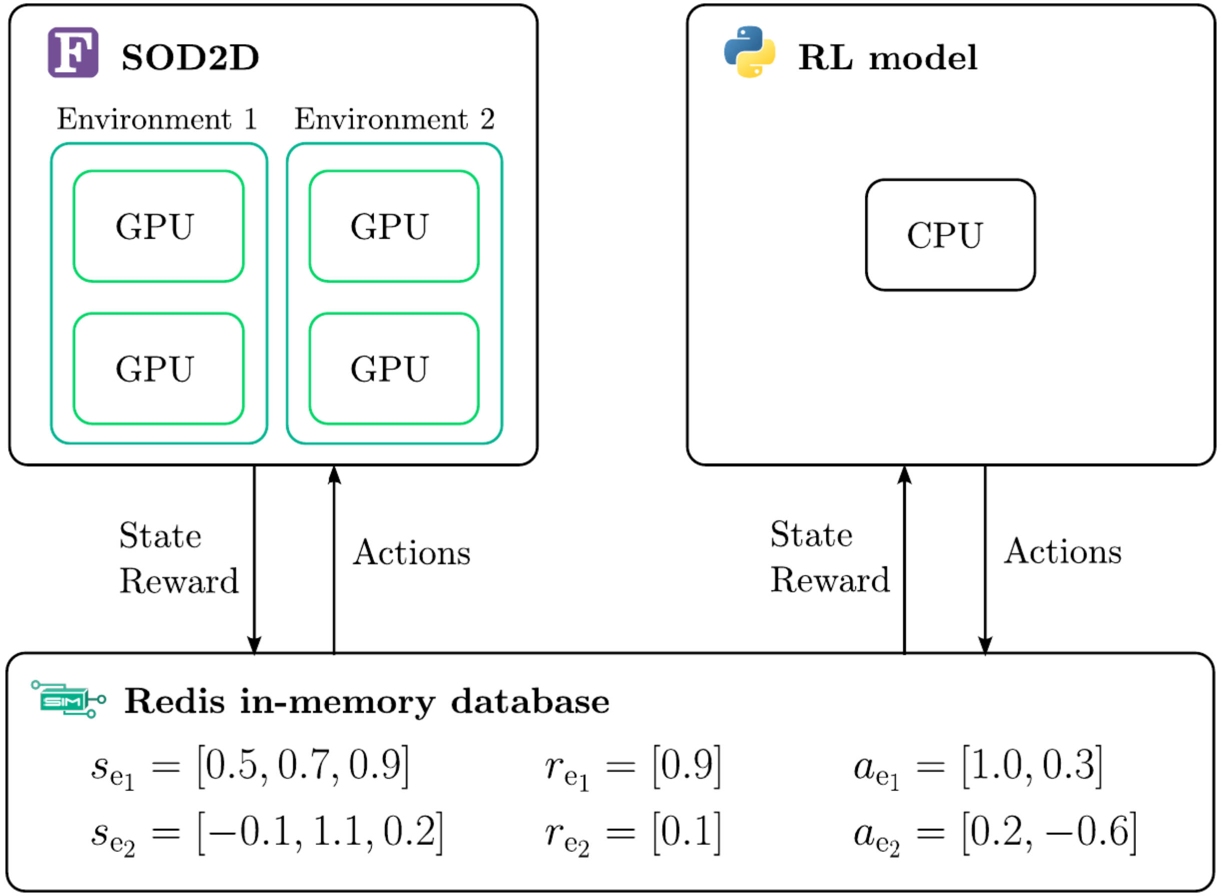
- *State* consists of a set of velocity probes within the bubble and surroundings
- *Reward*: wall shear stress as a proxy for recirculation length
- *Action*: Instantaneous zero-net-mass-flux
  - control half of the actuators (ρv), impose opposite ($-\rho v$) on the other half *
- Multi-environment
  - Multiple simulations run in parallel that share the controlling agent (RL model)
- Optimal control
  - Action sampled from an optimal distribution that maximizes the reward
  - Control signal can contain multiple frequencies

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

KTH VETENSKAP OCH KONST

6. ML integration



- Multiple parallel simulations interacting with the RL model
  - Accelerated training
- Execution scheme that fills both GPUs and CPUs
  - RL model trains on the CPU while CFD simulation runs on GPUs
- In-memory Redis database
  - Measured minimal communication overhead
- RL control strategy
  1. Environment sends state and reward to the agent
  2. Agent predicts actions and sends them to the environment
  3. Environment applies actions and proceeds to next time step

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

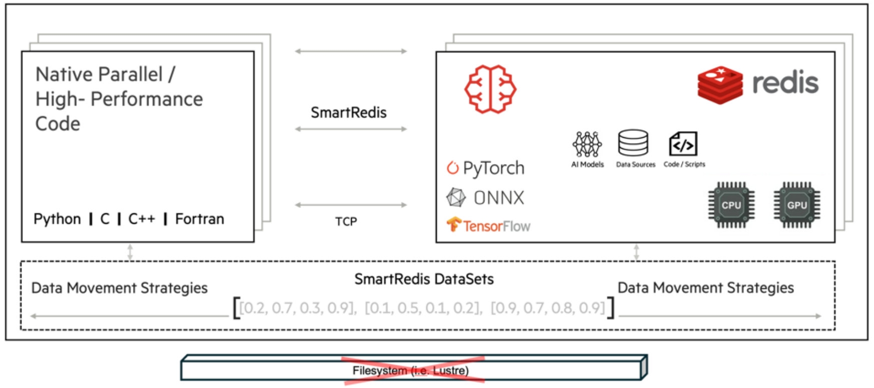
B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

KTH VETENSKAP OCH KONST

6. ML integration

SmartSim

- Python library to create databases (DBs) and managing workloads in HPC environments.
- Allows linking HPC applications with ML models written in Python.
- DB can be located in a single node, distributed across nodes, or fully duplicated across nodes.



https://www.craylabs.org/docs/overview.html

SmartRedis

- DB clients for different programming languages (Fortran, Python, C/C++) with consistent API.
- Recently updated to be compilable with NVIDIA HPC toolkit (required to link with apps compiled by nvhpc, like SOD2D).
- Read/write ”tensors” (named arrays) into DB.
- Allows syncronization across programs by waiting on arrays to be created into the DB.
- Allows the freedom to write data from each rank of each simulation (one client per rank), or gather data and then write it from a single rank (one client per MPI communicator).

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

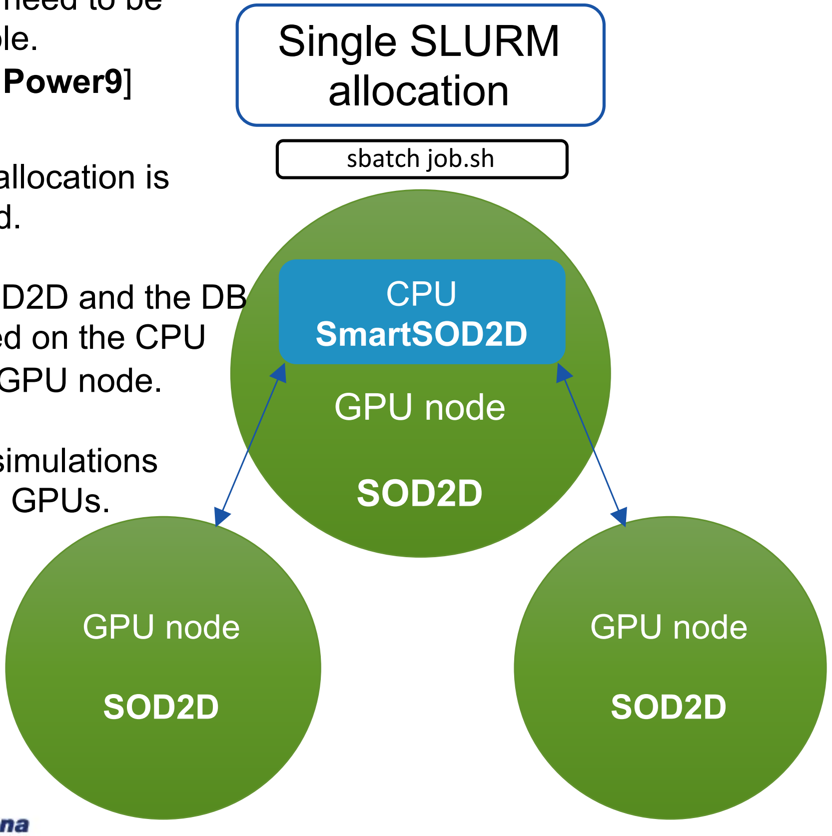
B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

6. ML integration

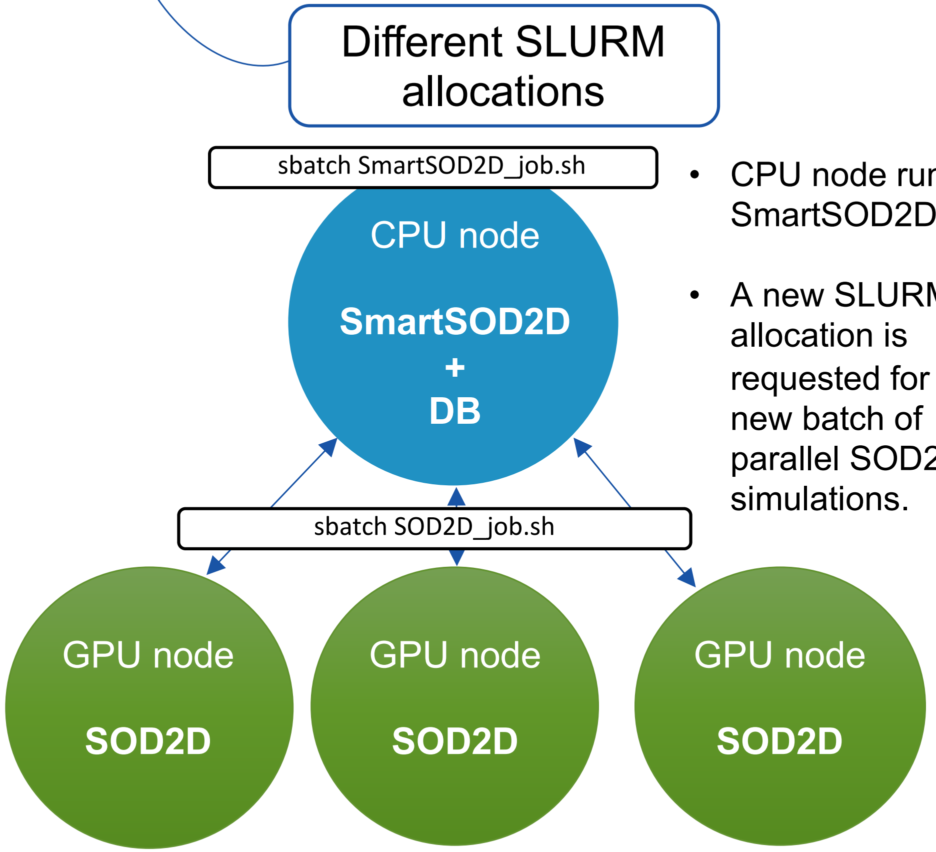
Framework management using SmartSim

Two different ways to run the framework in a cluster

- SmartSOD2D (Python) and SOD2D (Fortran) share the same module environment, so modules need to be compatible.
  [setup in **Power9**]
- A single allocation is requested.
- SmartSOD2D and the DB are pinned on the CPU part of a GPU node.
- SOD2D simulations run on all GPUs.



- SmartSOD2D (Python) and the SOD2D (Fortran) do not share the same module environment.
  [setup in **ALVIS**]



- CPU node running SmartSOD2D.
- A new SLURM allocation is requested for every new batch of parallel SOD2D simulations.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

KTH VETENSKAP OCH KONST

6. ML integration

Single SLURM allocation [Power9]

sbatch job.sh

```
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=40
#SBATCH --gres=gpu:4
. all_modules.sh
python train.py
```

SmartSim runs: $

```
mpirun –x SSDB=<IP-address:port> \
 -n 1 sod2d --args1 :
 -n 1 sod2d --args2 : …
 -n 1 sod2d --args8
```

Different SLURM allocations [Alvis]

sbatch SmartSOD2D_job.sh

```
#SBATCH -C NOGPU -n 1
. smartsod2d_modules.sh
python train.py
```

SmartSim runs: sbatch SOD2D_job.sh

```
#SBATCH --ntasks=8
#SBATCH --gpus-per-node=A100:4
. sod2d_modules.sh

mpirun –x SSDB=<IP-address:port> \
 -n 1 sod2d --args1 :
 -n 1 sod2d --args2 : …
 -n 1 sod2d --args8
```

(1 GPU per SOD2D simulation in these cases)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

KTH VETENSKAP OCH KONST

6. ML integration

Distributing simulations across available resources

Multiple-Program Multiple-Data (MPMD)

- The MPMD model distributes the requires MPI processes across the available resources (GPUs or CPUs).
- The MPI communicator is split within SOD2D, and each simulation gets its own ID (colouring) → Useful when reading/writing data into the DB.

```
! Get the unique app number (color)
call MPI_Comm_get_attr(world_comm, MPI_APPNUM, color_ptr, mpi_app_num_flag, mpi_err)
call MPI_Comm_split(world_comm, color_ptr, mpi_world_rank, app_comm, mpi_err)
call MPI_Comm_rank(app_comm, mpi_rank, mpi_err)
call MPI_Comm_size(app_comm, mpi_size, mpi_err)
```

- The global communicator across simulations `world_comm` is not used, only the local simulation communicator `app_comm`.
- Maybe use **pycompss** to handle processes and distribute workloads in the near future!

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

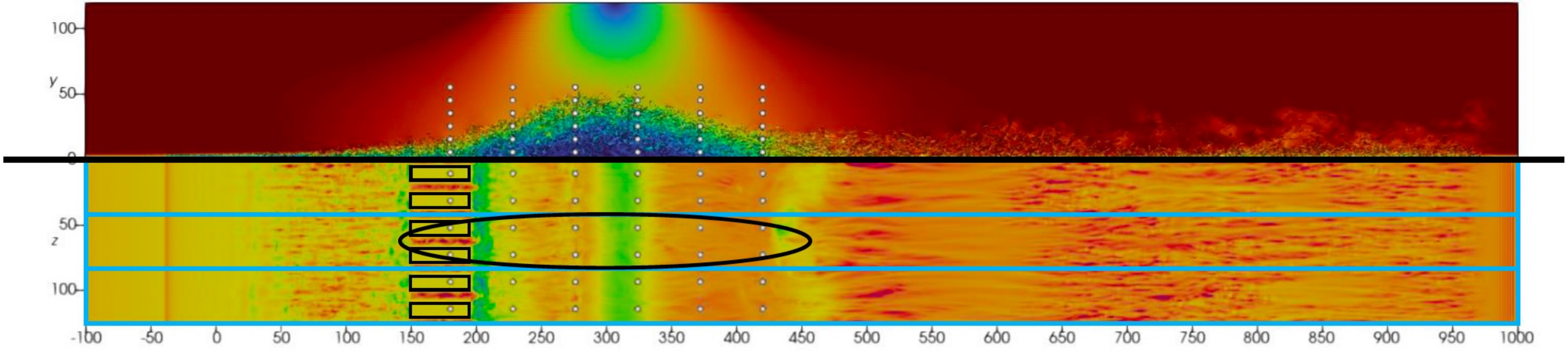
KTH VETENSKAP OCH KONST

6. ML integration

- MARL: Multi-agent reinforcement learning
  - Exploits **invariances → Pseudo-environment**
  - Reduces action dimensionality → **Curse of dimensionality**
  - References: **Guastoni et al. TEPJE, 46(4), 27** & **Vignon et al. PoF, 35, 6, 2023**
- State
  - 72 probes: Streamwise velocity, u

- Action
  - Jet pairs → ZNMF
  - $|v_{\mathrm{ac,max}}| = 0.3$
- Reward
  - Reduce bubble of recirculation
  - Proxy: Area of $\tau_w < 0$
  - $r = (r_{\mathrm{global}} + r_{\mathrm{local}})/2$

- Other parameters
  - 552 MARL episodes
  - 184 CFD episodes
  - 1600 time units/episode
  - 40 actions/episode
  - 35 hours training

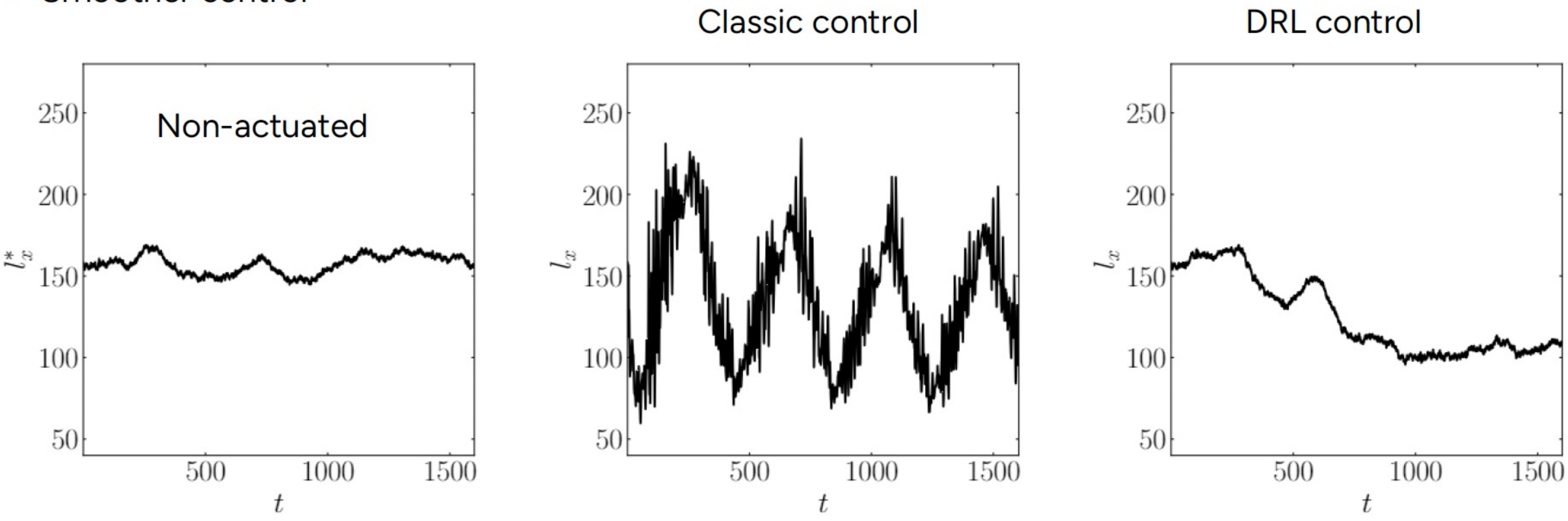


B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

6. ML integration

Results: DRL control

- Bubble reduction of **28.9%** in coarse grid
  - 12% more than classic control
- Smoother control



B. Font, F. Alcantara-Avia, J. Rabault., R. Vinuesa, O. Lehmkuhl (2024) Active flow control of a turbulent separation bubble through deep reinforcement learning, Journal of Physics: Conference Series

7. Conclusions

To wrap up

1. The present work has assessed and analysed the parallel performance of a new Continuous Galerkin High-Order Spectral Element Code aimed to solve simulations of turbulent compressible in the context of the CEEC project.
2. The obtained preliminary results are very promising: the code presents very good scalability
   - for both **strong** and **weak** speed-ups.
3. Validation efforts on-going however good results observed so far in a relevant benchmark cases.
4. First efforts on integration of ML algorithms promising

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you!

Any questions?

oriol.lehmkuhl@bsc.es