Towards data driven reduced order models for the automotive industry

B. Eiximeno, S. Gómez, H. Owen, O. Lehmkuhl

23rd September, 2022

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MareNostrum5: a new EuroHPC world-class supercomputer in Spain

16 June 2022

The procurement contract of MareNostrum 5, a new EuroHPC pre-exascale supercomputer, has been signed by the European High Performance Computing Joint Undertaking (EuroHPC JU) and the company Atos, the selected vendor.

Outline

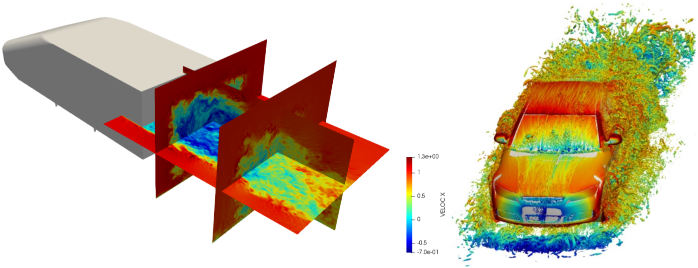
Outline

Case statistics

Case configuration

Wall modelled LES of the Windsor body and the Drivaer

- FEM with fractional step method solver.
- Vreman LES model.
- Reichardt wall law (exchange location at the 4th node).
- 4th order Runge-Kutta time integration.



Windsor body and Drivaer results

Windsor body

$C_D = 0.34$ - $C_{D_{EXP}} = 0.3298$

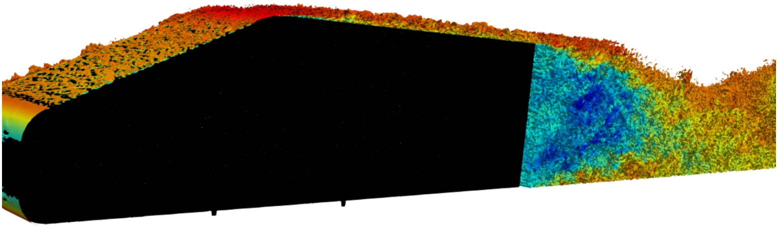
$C_L = -0.16$ - $C_{L_{EXP}} = -0.033$

$C_S = 0.15$ - $C_{S_{EXP}} = 0.1345$

Drivaer

$C_D = 0.25$ - $C_{D_{EXP}} = 0.2546$

$C_L = 0.028$ - $C_{L_{EXP}} = 0.0874$



Outline

Reduced order models

Definition

What are they?

- Data driven decompositions
- Any physical background
- Useful in complex datasets

Why should we use them?

- Flow field analysis
- Data storage and reconstruction
- Time interpolation
- Interpolation between different conditions (i.e. yaw angle)

Reduced order models

Definition

Proper orthogonal decomposition (POD):

Decompose a field in:

- Temporal coefficient
- Spatial distribution - POD modes

$$F(X, t) = \sum_{i=1}^{i=N} a_i(t)\Phi_i(X) \tag{1}$$

Decomposition ranked by energy:

$E_1 > E_2 > ... > E_{N-1} > E_N$

Reduced order models

Definition

Dynamic mode decomposition (DMD):

Linearize a dataset $\mathcal{D}$ of N flow fields:

$$\mathcal{D}_1^N = [d_1, d_2, d_3, ..., d_N] \tag{2}$$

$$\mathcal{D}_1^N = \left[d_1, Ad_1, A^2 d_1, ..., A^{N-1} d_1\right] \tag{3}$$

Compute spatial correlations which have a:

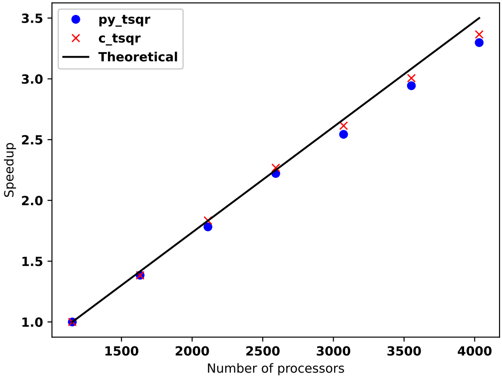
- Frequency
- Damping ratio
- Amplitude

Reduced order models

Performance

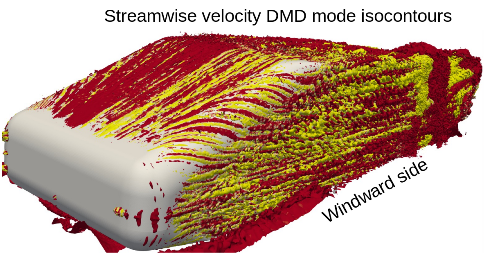
Implementation based on single value decomposition (SVD):

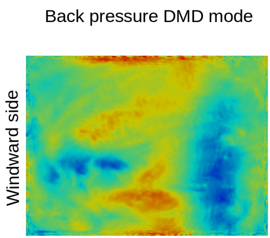
Algorithm which computes it at 70M points in 3.7 seconds



How does velocity affect the back pressure?

- Fluctuations at $St = 0.35$ (highest amplitude)





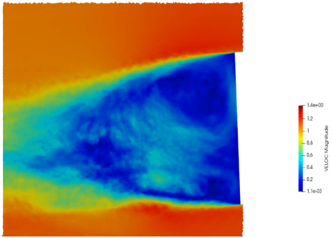
Reduced order models

Data storage and reconstruction

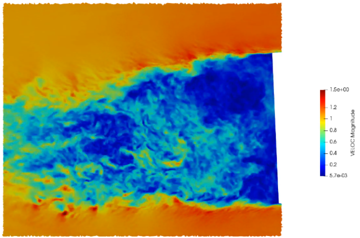
Reduction to the most meaningful modes

- Less space needed for storage (from 275GB to 6.9GB)
- Flow well captured
- Easy to reconstruct

10 modes

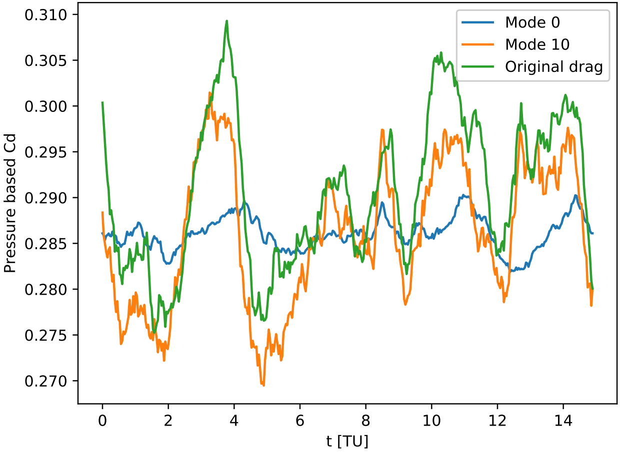


Original flow



Reduced order models

Data storage and reconstruction



Reduced order models

Yaw angle interpolation

Can we predict the back pressure at another yaw angle?

- Are there any modes related with the windward vortex?
- What happens if we give them an arbitrary contribution?

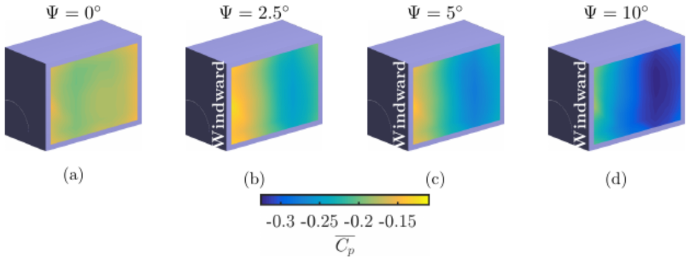


Figure: Extracted from Max Varney 2019

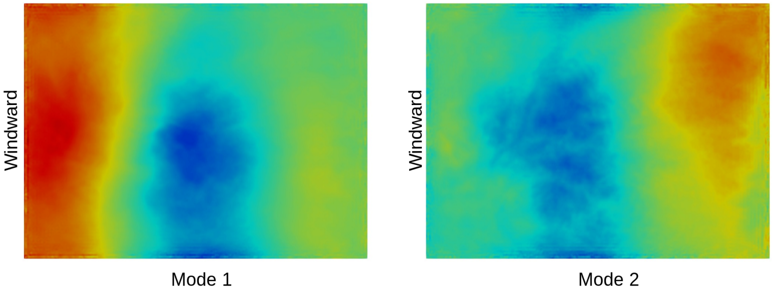
Reduced order models

Yaw angle interpolation

The two most energetic pressure POD modes:

Reproduce the windward vortex effect:

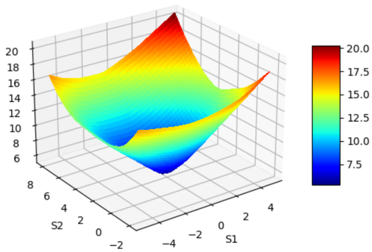
$$P(\delta) = P_{\delta=2.5^\circ} + S_1(\delta) M1_{\delta=2.5^\circ} + S_2(\delta) M2_{\delta=2.5^\circ} \qquad (4)$$



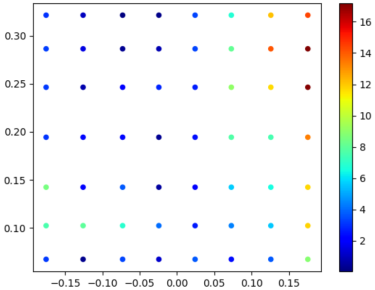
Computation of S_1 and S_2

- Error minimisation at $\delta = 5°$ and $\delta = 10°$
- Usage of experimental data
- Mean error of 5.3 % at $\delta = 5°$ and 14.64 % at $\delta = 10°$

Error function



Error in experimental probes

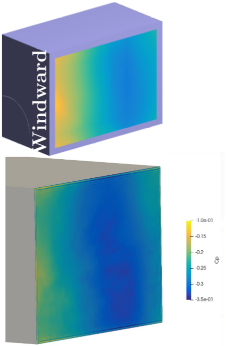


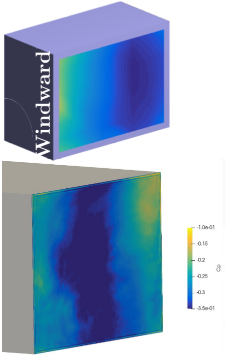
Reduced order models

Yaw angle interpolation

Interpolated back pressures

- Room for improvement at the top-leeward corner
- General agreement with experimental results



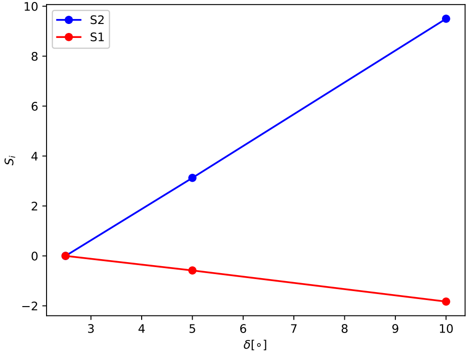


Reduced order models

Yaw angle interpolation

What about the rest of the angles?

$S_i(\delta)$ are linear (at least) between $\delta = 2.5^\circ$ and $\delta = 10^\circ$



Outline

Spectral elements code in GPU

Results and code performance

Numerical approach

- Full compressible SEM with RK4 for time integration
- Entropy stable arguments for strong gradients
- Skew symmetric splittings

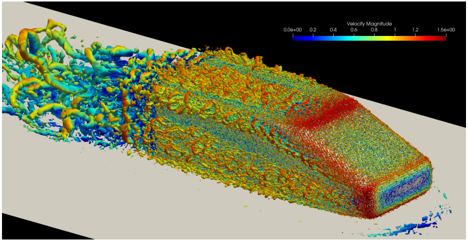


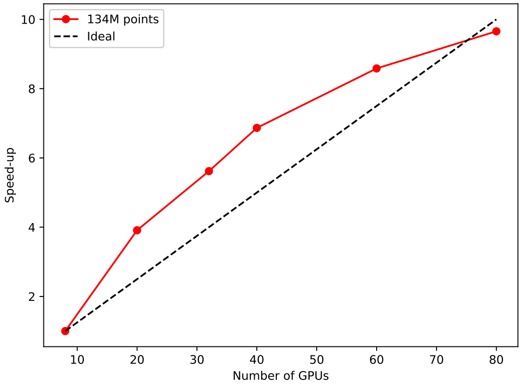
Figure: Third order 13M points mesh

Spectral elements code in GPU

Results and code performance

Performance and efficiency

- In GPU needs 4 times less of power than in CPU
- 0.032s/timestep with 134M points mesh in 80 GPU



Outline

Conclusions and future work

Conclusions

- Submitted results in agreement with experiments
- Reduced order models applications in CFD:
  - Flow field analysis
  - Data reduction
  - Results extrapolation
- Spectral elements code in GPU is more accurate and faster

Future work

- Perform better simulations with the new code
- Analyze deeply the results with ROMs
- Create a complete yaw angle interpolation model

Thank you very much for your attention