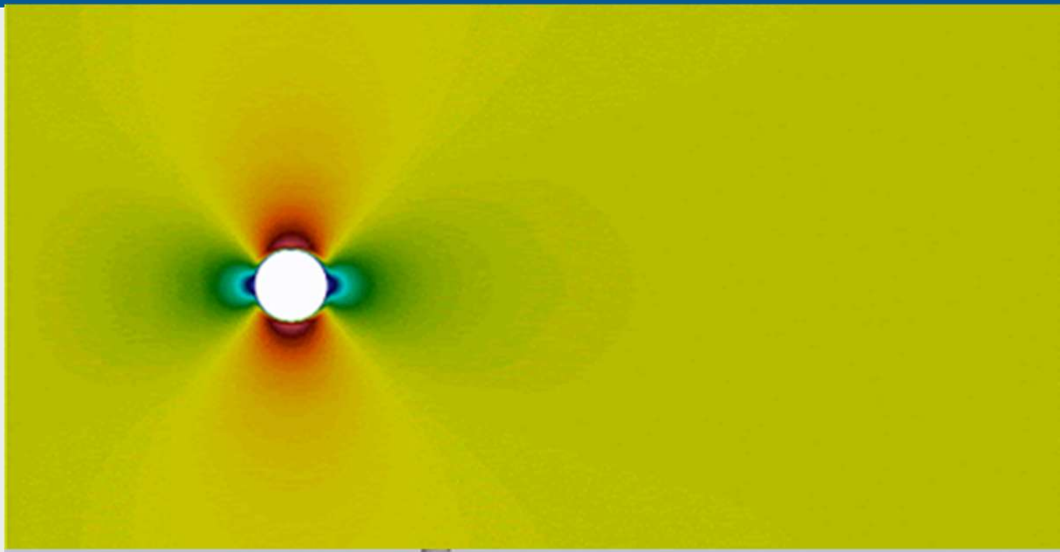
Enhancing computational fluid dynamics simulations by model reduction and scientific machine learning

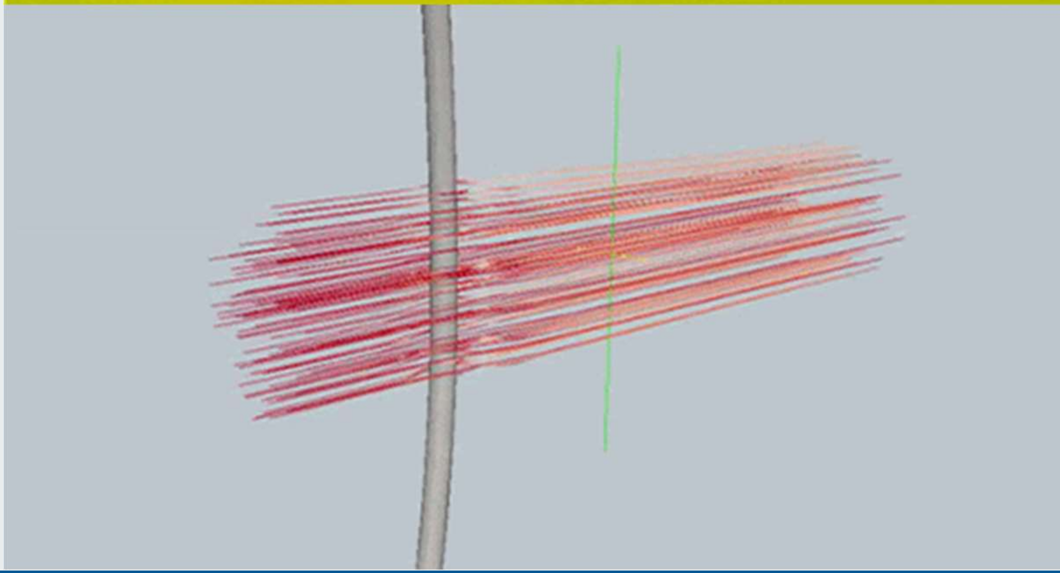
Gianluigi Rozza

mathLab, Mathematics Area, SISSA International School for Advanced Studies, Trieste, Italy
grozza@sissa.it

IMSI, University of Chicago
November 12, 2025

Introduction and Leading Motivations

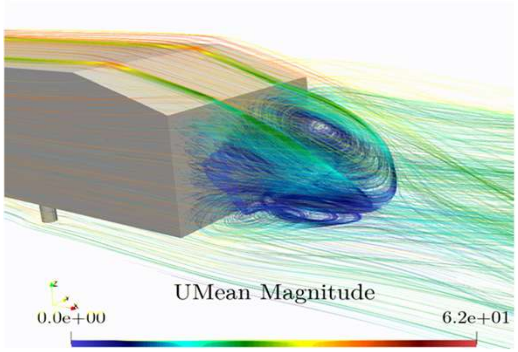




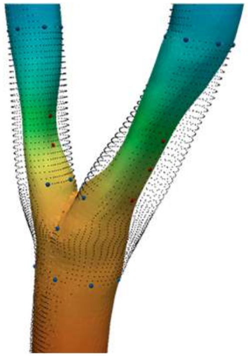
- Need of saving computational resources
- ***Offline-online coomputational*** procedures

Physical Parametric Differential Problems: Overview

Parametric Differential Problems are ubiquitous in many field of Natural and Applied Sciences from **naval** and **nautical** engineering, to **aeronautical** engineering, **bioengineering**, as well as **industrial** engineering.



automotive



biomedics

aeronautics

Rozza, Gianluigi, Giovanni Stabile, and Francesco Ballarin (2022) eds. Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics. Society for Industrial and Applied Mathematics., CSE series.

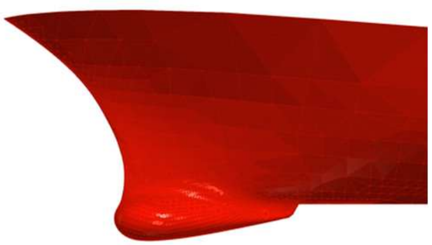
Leading Motivation: CFD challenges

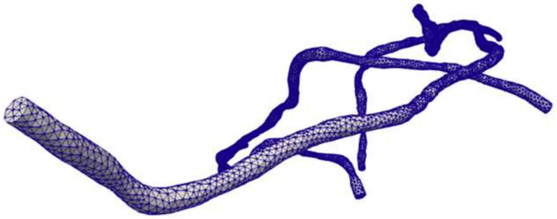
Growing demand of:

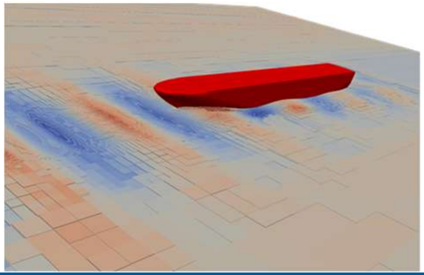
efficient tools	**uncertainty quantification**
parameterized formulations	**real-time** computations

Quickly emerging field of
Model Order Reduction
to efficiently **parametrize** and **accelerate** computations

Need of computational collaboration between **Full Order Model (FOM)+HPC** and **Reduced Order Model (ROM)**



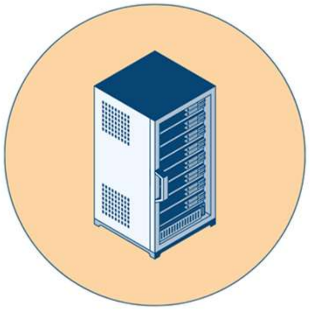




Towards real-time computing

Offline stage

The Full Order Model (FOM)



- Requires *super-computers (HPC)*
- *Expensive* computational resources
- *Several* degrees of freedom
- ***Extremely time-demanding***

Online stage

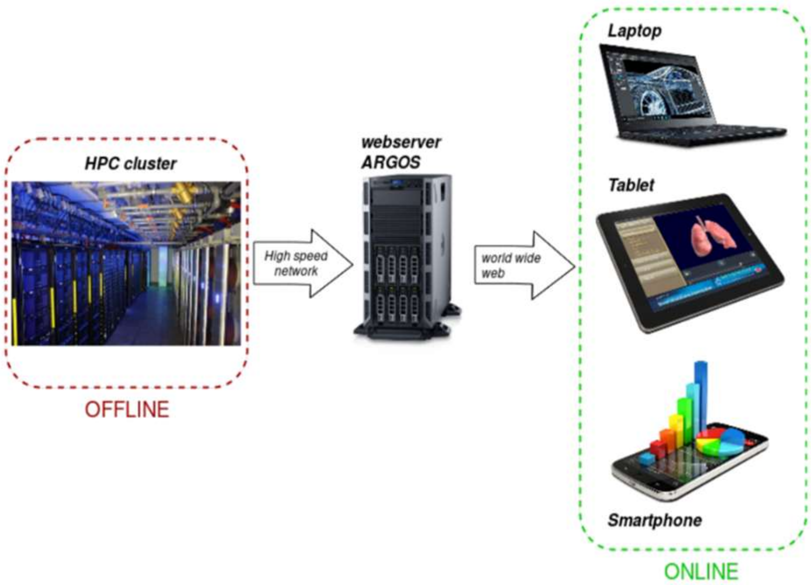
Reduced Order Model (ROM) techniques

- Needs a *laptop*
- *Small* computational resources
- *Few* degrees of freedom
- ***Fast, real-time computing***

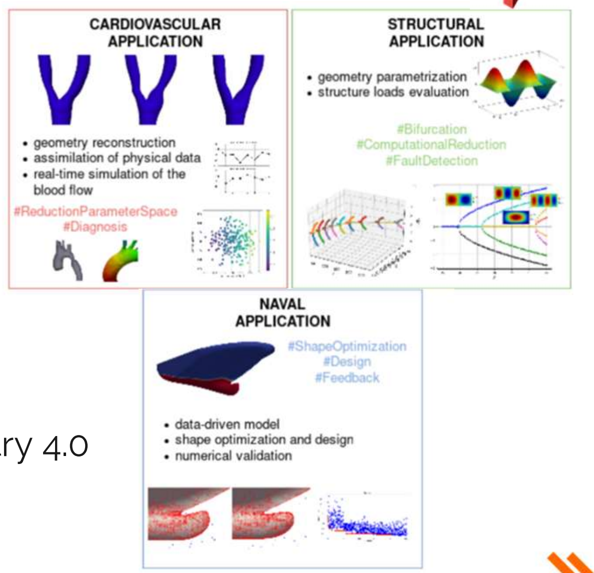
Technology perspective: computational webserver

Model order reduction for computational web server: to real world applications (ERC PoC ARGOS):

- `argos.sissa.it`
- `atlas.sissa.it`



- HPC
- data science
- Digital twin
- SMACT Industry 4.0
- 3D Printing



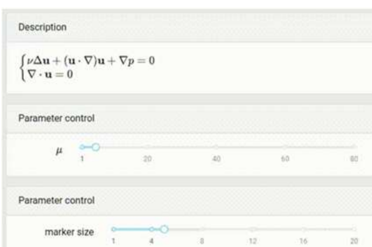
FAST COMPUTING

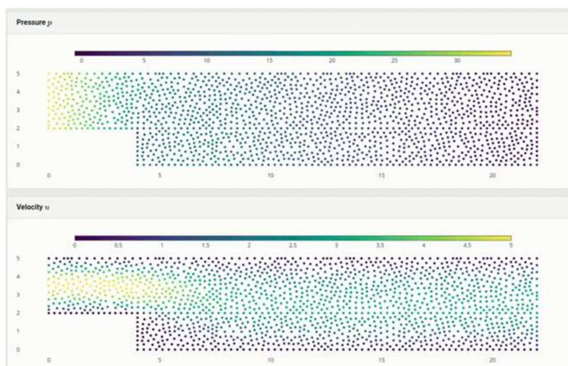
ARGOS - Computational Webserver

Model order reduction for computational web server: from academic to real world applications

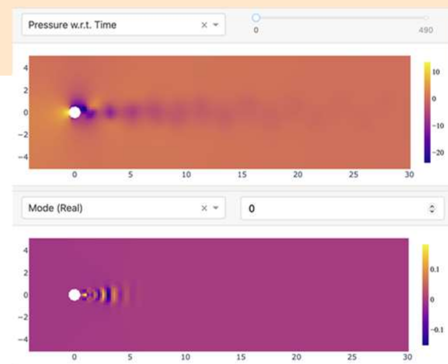
https://argos.sissa.it https://argos-edu.sissa.it

Benchmark applications and **worked out** problems for academia and beyond

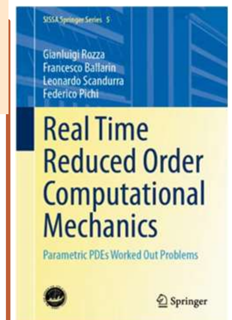




POD+Interpolation for Navier-Stokes



DMD



- **real-time** computation
- visualization tool

... and much more!

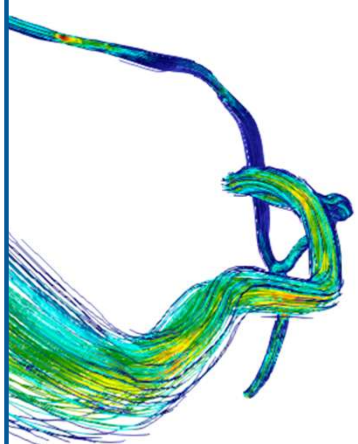
powered by FAST COMPUTING

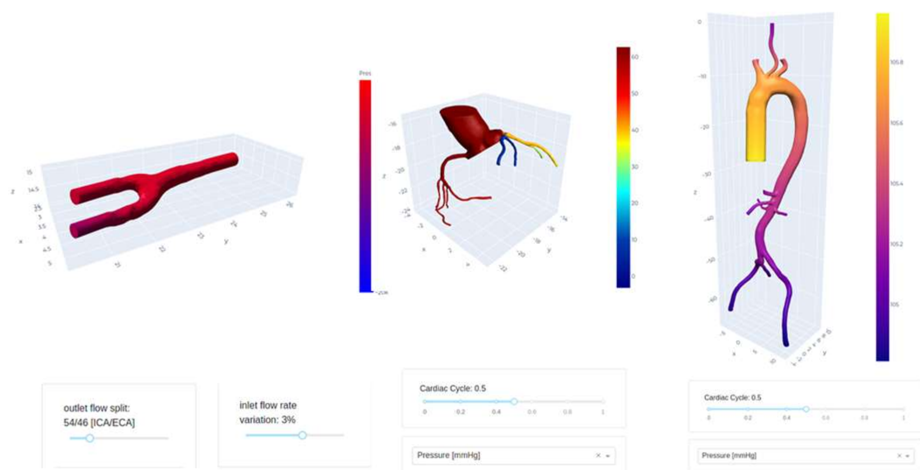
ATLAS - Computational Webserver

Model order reduction for computational web server: from academic to real world applications

https://atlas.sissa.it

A special focus on **cardiovascular** applications

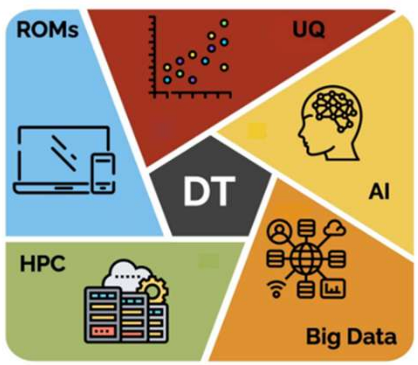




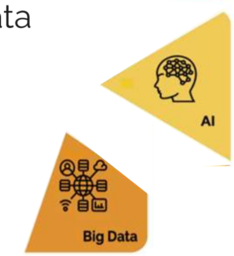
Carotid artery *Coronary arteries* *Aortic model*

powered by

Digital Twin (DT): integration of emerging fields

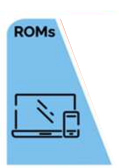


A large amount of data (**Big data**) can be collected



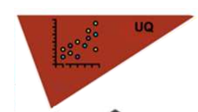
Artificial Intelligence can help to store and organize data.

- By using **black-box models,** AI techniques are able to find **fitting functions**
- It does not require knowledge of the physics of the problem, even if we do prefer integrated **"Big Models"** physics-informed approaches



The development of **High Performance Computing** (**HPC**) and its integration with **ROMs** allowed to reach better performances for:

- building **Digital Twins** (**DT**) of products and processes;
- **Uncertainty Quantification** (**UQ**);
- Data analytics.



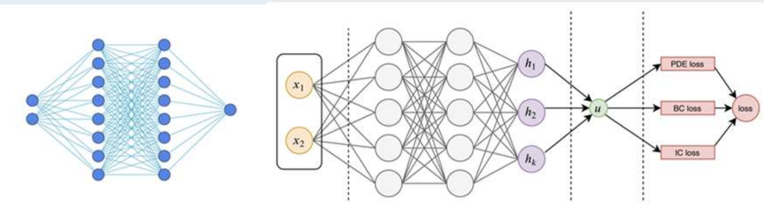
A sustainable perspective (reducing energy consumption, recycling computational works)

SISSA mathLab: our current efforts and perspectives

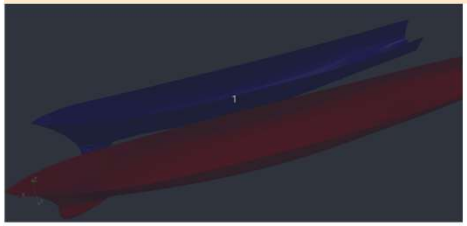
A team developing **Advanced Reduced Order Methods** for parametric PDEs!

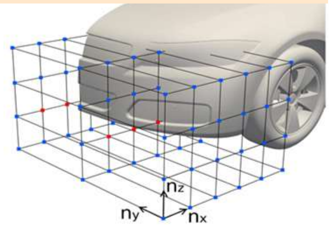
SISSA mathLab: our current efforts and perspectives

Face and overcome **some limitations** of classic parametric ROM also by means of **Machine Learning**

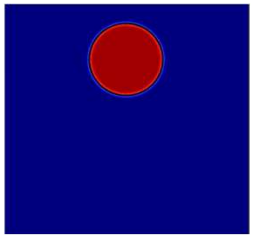


Improve capabilities of ROMs for **more demanding applications** in industrial, medical and applied sciences

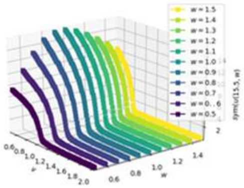


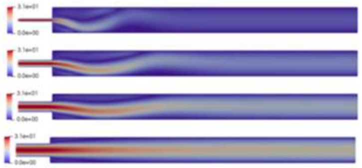


CFD as a central topic to enhance broader applications in **multiphysics** and **coupled** settings



Carry out important **methodological developments** with special emphasis on **mathematical modelling**





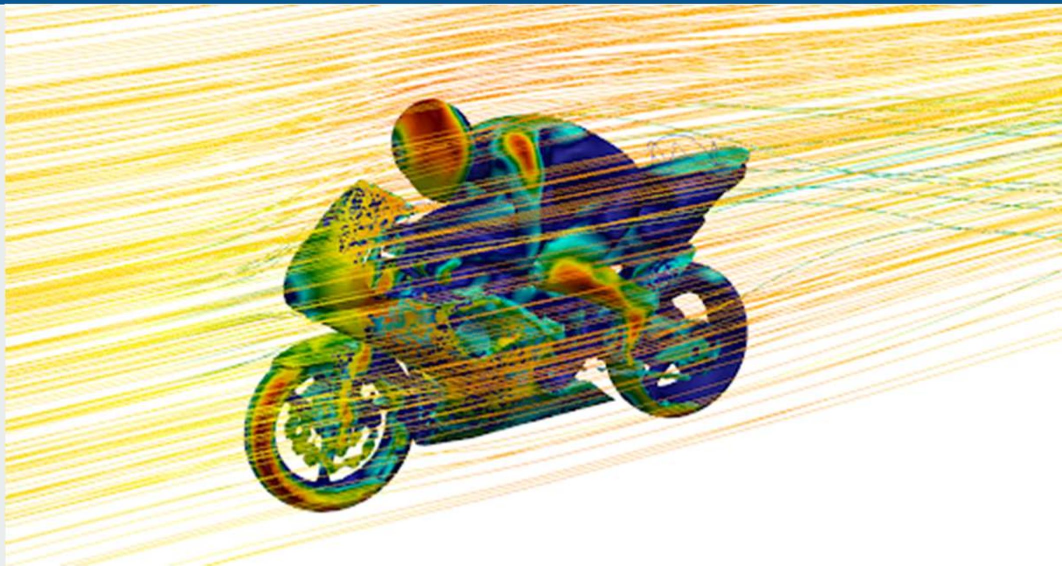
SISSA mathLab: our current efforts and perspectives
Open source libraries: mathlab.sissa.it/cse-software

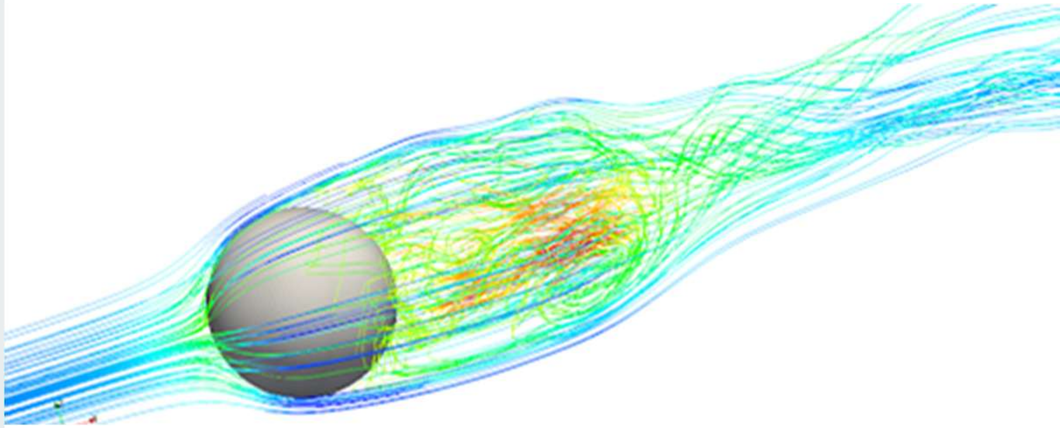
Development of **open-source tools based on surrogate modeling**:

- **ITHACA**, In real Time Highly Advanced Computational Applications, as an add-on to integrate already well established CSE/CFD open-source software
- **RBniCS** as educational initiative (FEM) for newcomer ROM users (training).
- **EzyRB,** data-driven model order reduction for parametrized problems
- **PyDMD,** a Python package designed for Dynamic Mode Decomposition (in collaboration with University of Texas, CERN, and University of Washington)
- **ARGOS** **A**dvanced **R**educed order modellin**G** **O**nline computational web server for parametric **S**ystems
- **PINA,** a deep learning library to solve differential equations

PINA ITHACA ITHACA - FV PyDMD ITHACA-DG RBniCS multiphenics ARGOS ATLAS EzyRB

Challenges for ROM in CFD





- Strategic fields of development of ROMs
- Review and limitations of classical high-fidelity approaches

Full Order Models: a qualitative comparison

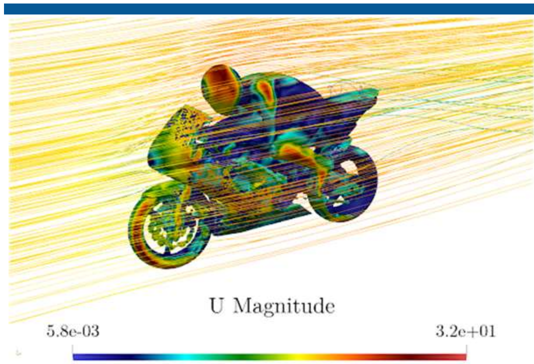
	Complexity	Computational cost	Conservation	Flexibility	Accuracy
Finite Element Method (FEM)	Moderately complex, especially in 3D problems	Moderate to high cost due to matrix operations	Depends on formulation	Very flexible on complex geometries	High accuracy with higher-order elements
Finite Volume Method (FVM)	Less complex, straightforward formulation	Moderate cost, depends on grid quality	Inherently conservative	Less flexible compared to FEM	Moderate accuracy
Spectral Element Method (SEM)	Very high complexity, particularly in implementation	Very high cost, especially for high accuracy	Conservative	Flexible but requires high-quality mesh	Very high accuracy with exponential convergence
Discontinuous Galerkin (DG)	High complexity, especially in dealing with interfaces	High cost due to local problem solving and flux calculation	Naturally conservative	Very flexible, suitable for complex geometries	High accuracy with appropriate polynomial order

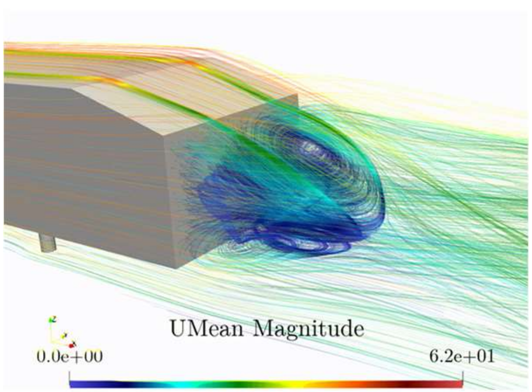
Full Order Models: a qualitative comparison

	Complexity	Computational cost	Conservation	Flexibility	Accuracy
Finite Element Method (FEM)	Moderately complex, especially in 3D problems	Moderate to high cost due to matrix operations	Depends on formulation	Very flexible on complex geometries	High accuracy with higher-order elements
Finite Volume Method (FVM)	Less complex, straightforward formulation	Moderate cost, depends on grid quality			Moderate accuracy
Spectral Element Method (SEM)	Very high complexity, particularly in implementation	Very high cost, especially for high accuracy	Conservative	Flexible but requires high-quality mesh	Very high accuracy with exponential convergence
Discontinuous Galerkin (DG)	High complexity, especially in dealing with interfaces	High cost due to local problem solving and flux calculation	Naturally conservative	Very flexible, suitable for complex geometries	High accuracy with appropriate polynomial order

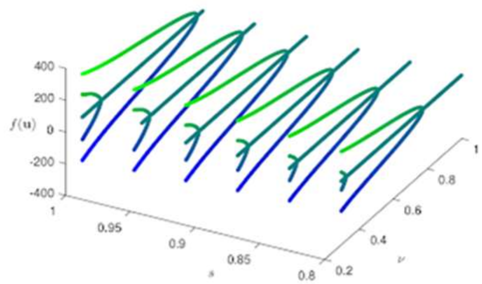
Motivations for ***ROM development***

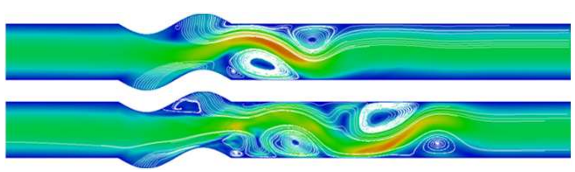
Strategic fields of development of ROMs in CFD



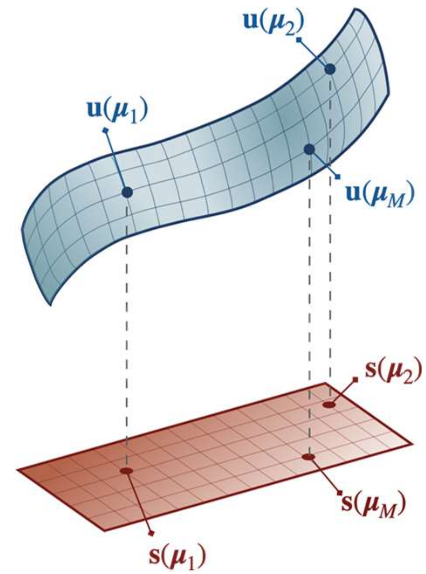


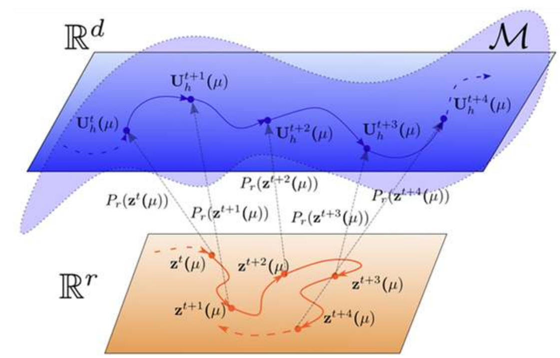
- **Geometric parameterization** and shape design (appendix)
- High Reynolds number (**turbulence**)
- **Automatic learning** developments
- **Parameter space reduction**
- Flow control and **data assimilation**
- **Bifurcations (loss of uniqueness of the solution)** (appendix)

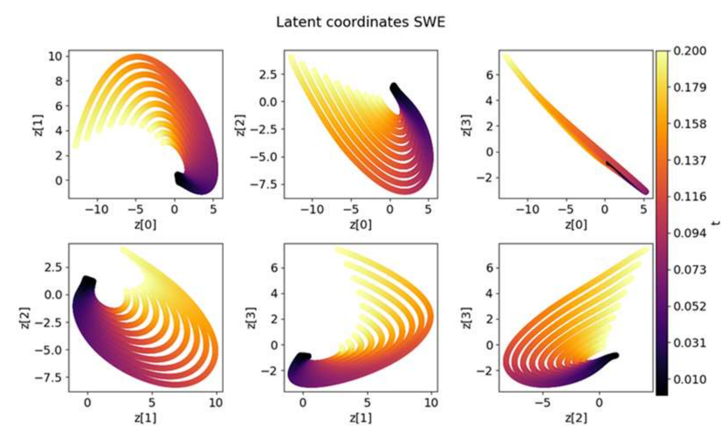




Reduced Order Models (ROMs)







- ➔ Equation-based or fully data-driven
- ➔ Machine-learning enhanced ROMs
- ➔ Fast Online Phase

Reduced Order Model - Accelerating Numerics

Problem: to find the approximation for an unseen (test) parameter $\boldsymbol{\mu}^\star$

Two macro-types of ROM approach:

Non-Intrusive ROM

- ***purely data-driven*** approach

$$\mathbf{u}(\boldsymbol{\mu}) \xrightarrow{\textit{reduce, then approximate}} \mathbf{u}_r(\boldsymbol{\mu}^\star)$$

- no knowledge of the mathematical model needed

Intrusive ROM

- ***equation-based*** approach

$$\mathscr{A}(\mathbf{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0 \xrightarrow{\textit{reduce, then evolve}} \mathscr{A}_r(\mathbf{u}_r(\boldsymbol{\mu}^\star), \boldsymbol{\mu}^\star) = 0$$

- consolidated mathematical theory

Hesthaven, J. S., Rozza, G., & Stamm, B. (2016). Certified reduced basis methods for parametrized partial differential equations (Vol. 590, pp. 1-131).

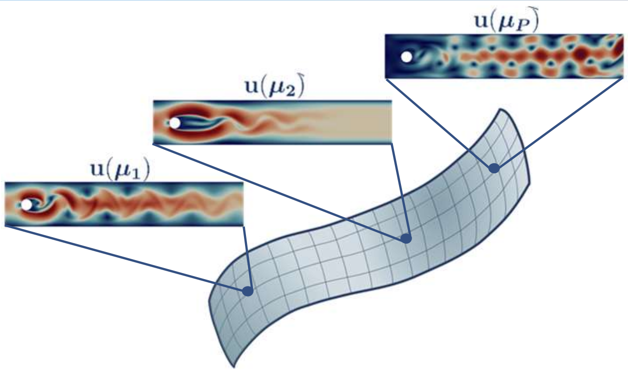
Rozza, Gianluigi, Giovanni Stabile, and Francesco Ballarin (2022) eds. Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics. Society for Industrial and Applied Mathematics., CSE series.

Benner, P., Schilders, W., Grivet-Talocia, S., Quarteroni, A., Rozza, G., & Miguel Silveira, L. (2020). Model Order Reduction: Volume 1, 2, 3. De Gruyter.

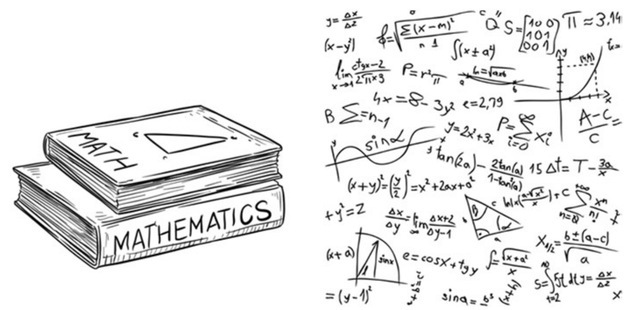
Reduced Order Model - Accelerating Numerics

Recent research goal: integrate data and physics' knowledge

Hybrid ROMs



Data collection and integration



Knowledge of the ***equations*** and the ***physical model*** of interest

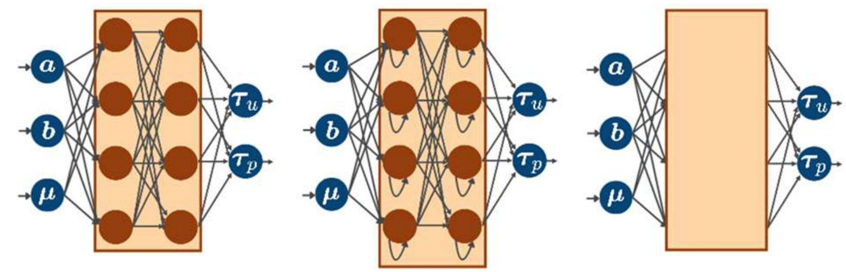
Rozza, Gianluigi, Giovanni Stabile, and Francesco Ballarin (2022) eds. Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics. Society for Industrial and Applied Mathematics., CSE series.

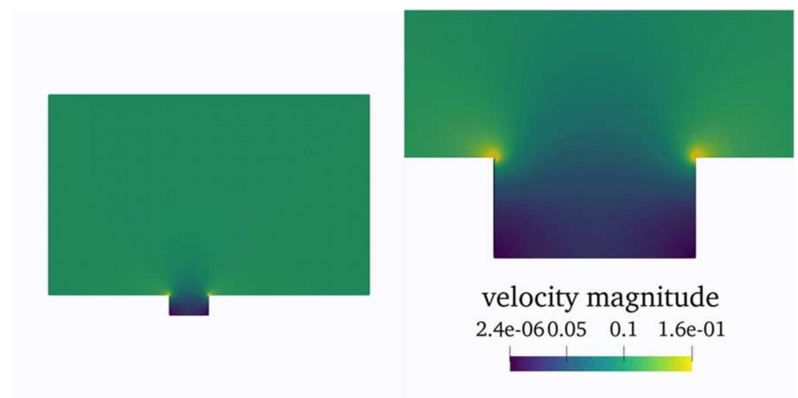
Benner, P., Schilders, W., Grivet-Talocia, S., Quarteroni, A., Rozza, G., & Miguel Silveira, L. (2020). Model Order Reduction: Volume 1, 2, 3. De Gruyter.

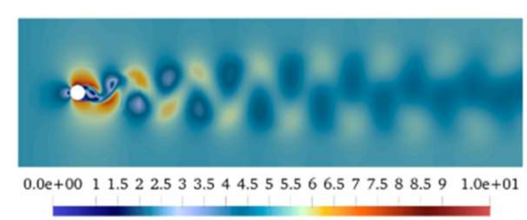
Hybrid data-driven intrusive ROMs for turbulent flows

- ➔ Hybrid approaches for reduced order models
- ➔ How to stabilize and enhance the flows?
- ➔ How to integrate ROMs with machine learning?

Joint work with:
Anna Ivagnes, Giovanni Stabile

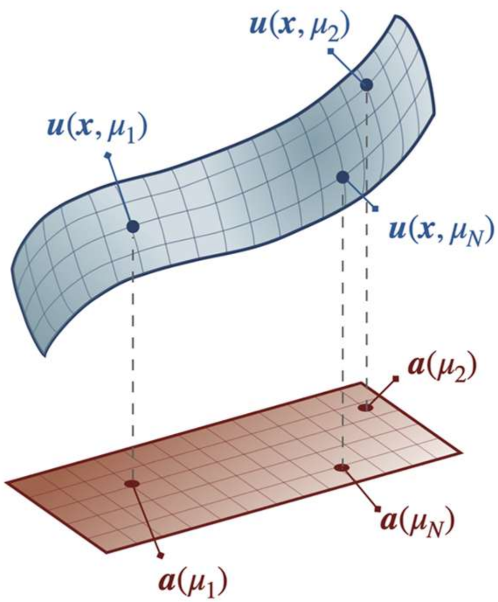






Intrusive ROMs - POD-Galerkin approach

POD principles



Linearity hypothesis

$$\boldsymbol{u}(\boldsymbol{x},\mu) \sim \sum_{i=1}^{N_u} a_i(\mu)\,\boldsymbol{\varphi}_i(\boldsymbol{x}) \qquad p(\boldsymbol{x},\mu) \sim \sum_{i=1}^{N_p} b_i(\mu)\,\chi_i(\boldsymbol{x})$$

$\boxed{\boldsymbol{x} \in \mathbb{R}^{d \times N_{dof}}}$

- ***d:*** dimension
- N_{dof} : degrees of freedom <u>*(several)*</u>

$\boxed{N_u, N_p \ll N_{dof}}$: reduced dimensions for velocity and pressure, chosen ***a priori***

$\boldsymbol{a} = (a_i)_{i=1}^{N_u}$

$\boldsymbol{b} = (b_i)_{i=1}^{N_p}$

vectors of coefficients (**parameter-dependent**)

$(\boldsymbol{\varphi}_i)_{i=1}^{N_u} \quad (\chi_i)_{i=1}^{N_p}$

POD modes (**space-dependent**)

Intrusive ROMs - POD-Galerkin approach

The ***Galerkin*** approach:

- momentum equation projected into the **velocity modes** $$\left(\boldsymbol{\varphi}_i, \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T\right) + \boldsymbol{\nabla} p\right)_{L^2(\Omega)} = 0.$$
- continuity equation projected into the **pressure modes** $$(\chi_i, \boldsymbol{\nabla} \cdot \boldsymbol{u})_{L^2(\Omega)} = 0.$$

Reduced ODEs system *(compact form)*

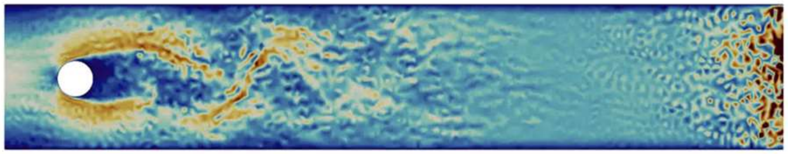
$$\begin{cases} \dot{\boldsymbol{a}}(t^n) = \boldsymbol{f}\left(\boldsymbol{a}(t^n), \boldsymbol{b}(t^n); \boldsymbol{\mu}^*\right), \\ \boldsymbol{h}\left(\boldsymbol{a}(t^n)\right) = \boldsymbol{0}. \end{cases} \longrightarrow$$

dynamical (*cheap*) system to be solved at each *n*-th iteration

Stabilized POD-Galerkin ROMs

Stabilization issues in standard ROMs:

- **spurious oscillations**
- ***reduced* inf-sup condition** not fulfilled



Supremizer enrichment

- Enrichment of the velocity POD space with additional N_{sup} modes
- Fulfillment of the inf-sup condition

$$\boldsymbol{a} = (a_i)_{i=1}^{N_u+N_{sup}}$$
$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\mu}) = \sum_{i=1}^{N_u+N_{sup}} a_i(\boldsymbol{\mu})\boldsymbol{\varphi}_i(\boldsymbol{x})$$

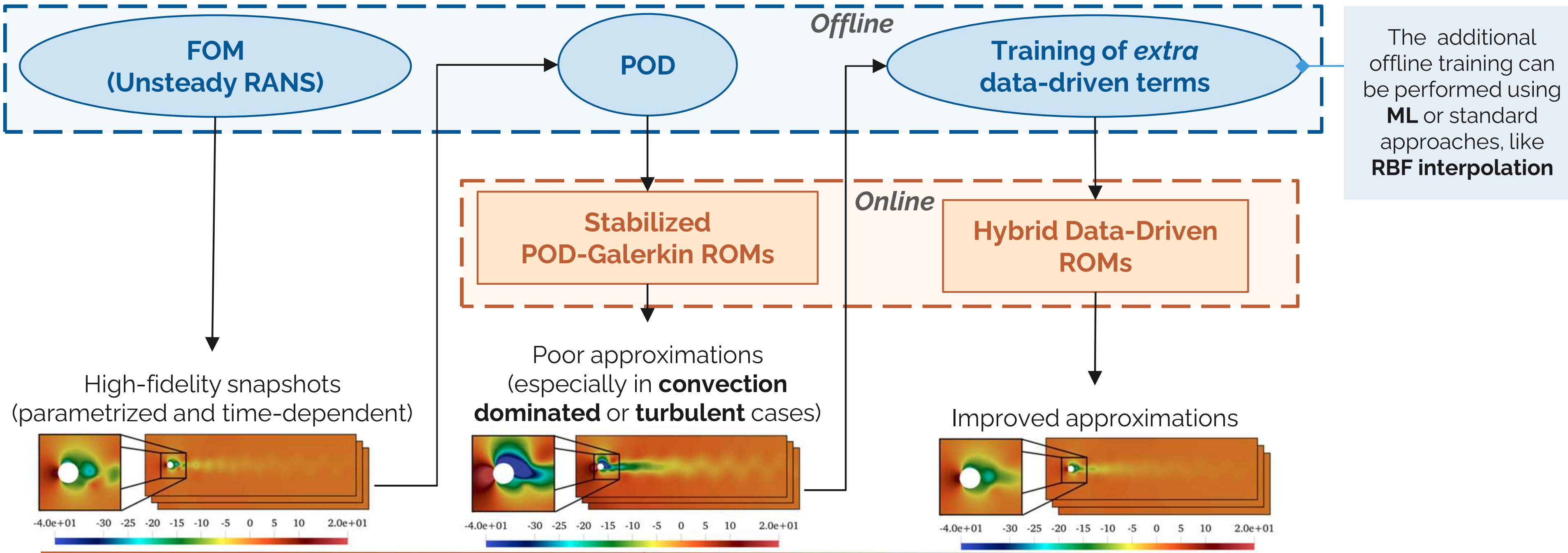
Pressure Poisson Equation

Replacement of the continuity equation with PPE

- at the ***FOM*** level:
$$\nabla \cdot \boldsymbol{u} = 0 \rightarrow \Delta p = -\nabla \cdot (\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}))$$
- at the ***ROM*** level *(at each time step)*:
$$\begin{cases} \dot{\boldsymbol{a}}(t^n) = \boldsymbol{f}\big(\boldsymbol{a}(t^n), \boldsymbol{b}(t^n); \boldsymbol{\mu}^*\big), \\ \boxed{\boldsymbol{h}_{\mathrm{PPE}}\big(\boldsymbol{a}(t^n), \boldsymbol{b}(t^n); \boldsymbol{\mu}^*\big) = \boldsymbol{0}.} \end{cases}$$

Stabile, G., & Rozza, G. (2018). Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. Computers & Fluids, 173, 273-284.

Stabilized ROMs enhanced with data



Ivagnes, A., Stabile, G., & Rozza, G. (2024). Parametric Intrusive Reduced Order Models enhanced with Machine Learning Correction Terms. arXiv preprint arXiv:2406.04169.

Purely DD-ROMs

Purely data-driven approach

WHY

Reintroduce the contribution of the neglected modes in a <u>*LES*</u> fashion

HOW

The procedure to build the *extra-correction* terms

- Choose a reduced dimension ***r*** and a bigger dimension ***d>r***
- Select a stabilization $\mathcal{C}$ operator
- Compute the *exact correction* $\boldsymbol{\tau}^{exact} = \overline{\mathcal{C}(\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r, \boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_d)}^r - \mathcal{C}(\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r)$
- Create a map for the approximated correction $\boldsymbol{\tau}^{approx} = \boldsymbol{\tau}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}) = \boxed{\mathcal{M}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}; \boldsymbol{\theta}_{\mathcal{M}})}$
- Train the map: $\min_{\boldsymbol{\theta}_{\mathcal{M}}} ||\mathcal{M}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}; \boldsymbol{\theta}_{\mathcal{M}}) - \boldsymbol{\tau}^{exact}||_{L^2}$

PURELY DD-ROM

$$\begin{cases} \dot{\boldsymbol{a}} = \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{\mu}^*) + \boldsymbol{\tau}_u(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}^*), \\ \boldsymbol{h}_{\text{PPE}}(\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{\mu}^*) + \boldsymbol{\tau}_p(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}^*) = \boldsymbol{0}. \end{cases}$$

Ivagnes, A., Stabile, G., Mola, A., Iliescu, T., & Rozza, G. (2023). Pressure data-driven variational multiscale reduced order models. Journal of Computational Physics, 476, 111904.

Physics-based DD-ROMs

Physics-based data-driven approach

WHY

Reintroduce the turbulence modeling in ROMs in a *RANS* fashion

HOW

Modeling the reduced eddy viscosity

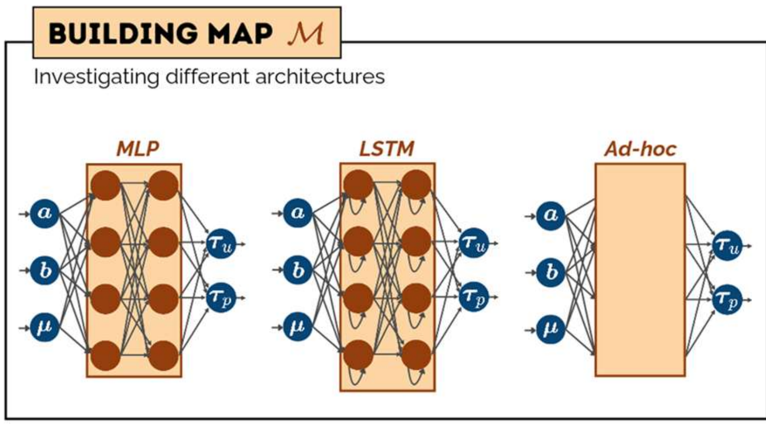
- Choose a reduced dimension for the eddy viscosity N_{ν_t}
- Extract the eddy viscosity modes $(\eta_i(\boldsymbol{x}))_{i=1}^{N_{\nu_t}}$ such that: $\nu_t(\boldsymbol{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_{\nu_t}} g_i(\boldsymbol{\mu})\eta_i(\boldsymbol{x})$
- Compute the *projected coefficients* $\boldsymbol{g}^{exact}$
- Create a map for the approximated correction $\boldsymbol{g}^{approx} = \boldsymbol{g}(\boldsymbol{a},\boldsymbol{\mu}) = \boxed{\mathcal{G}(\boldsymbol{a},\boldsymbol{\mu};\boldsymbol{\theta}_{\mathcal{G}})}$
- Train the map: $\min_{\boldsymbol{\theta}_{\mathcal{G}}} ||\mathcal{G}(\boldsymbol{a},\boldsymbol{\mu};\boldsymbol{\theta}_{\mathcal{G}}) - \boldsymbol{g}^{exact}||_{L^2}$

PHYSICS-BASED DD-ROM

$$\begin{cases} \dot{\boldsymbol{a}} = \boldsymbol{f}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{g}(\boldsymbol{a},\boldsymbol{\mu}^*);\boldsymbol{\mu}^*), \\ \boldsymbol{h}_{\mathrm{PPE}}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{g}(\boldsymbol{a},\boldsymbol{\mu}^*);\boldsymbol{\mu}^*) = \boldsymbol{0}. \end{cases}$$

Hijazi, S., Stabile, G., Mola, A., & Rozza, G. (2020). Data-driven POD-Galerkin reduced order model for turbulent flows. Journal of Computational Physics, 416, 109513.

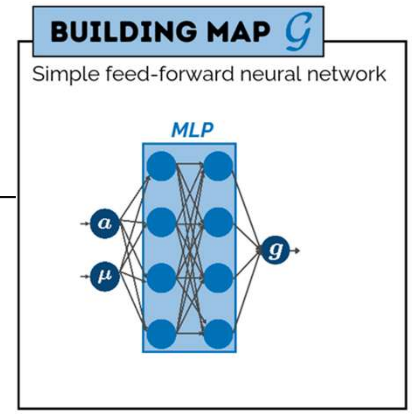
Machine learning maps



HYBRID DD-ROM

$$\begin{cases} \dot{\boldsymbol{a}} = \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{g}(\boldsymbol{a}, \boldsymbol{\mu}^*); \boldsymbol{\mu}^*) + \boldsymbol{\tau_u}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}^*), \\ \boldsymbol{h}_{\mathrm{PPE}}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{g}(\boldsymbol{a}, \boldsymbol{\mu}^*); \boldsymbol{\mu}^*) + \boldsymbol{\tau_p}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}^*) = \boldsymbol{0}. \end{cases}$$

Mixed online formulation



Purely data-driven training
(aimed to reintegrate missing modes)

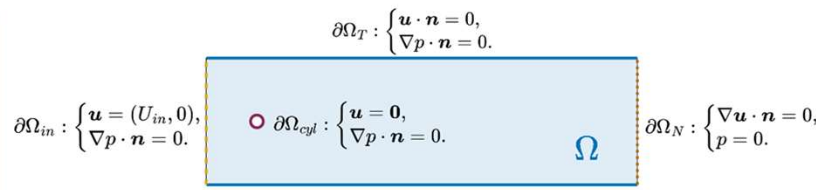
Physics-based data-driven training
(aimed to reintegrate turbulence)

Ivagnes, A., Stabile, G., & Rozza, G. (2024). Parametric Intrusive Reduced Order Models enhanced with Machine Learning Correction Terms. arXiv preprint arXiv:2406.04169.

Ivagnes, A., Stabile, G., Mola, A., Iliescu, T., & Rozza, G. (2023). Hybrid data-driven closure strategies for reduced order modeling. Applied Mathematics and Computation, 448, 127920.

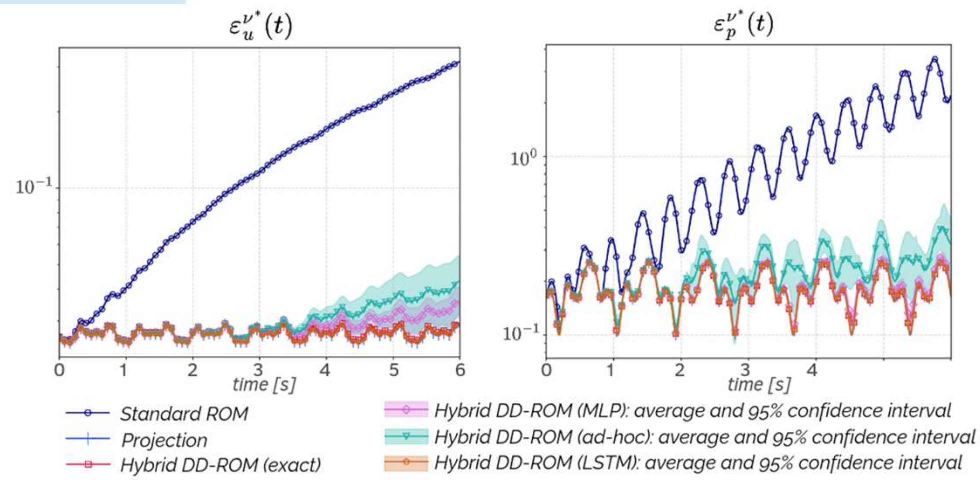
Numerical results

Test case: *periodic flow past a cylinder*
Parameters: *time and Reynolds number*
Number of modes: ***3*** *for velocity, pressure and eddy viscosity*



ERROR ANALYSIS
(for a test parameter and in time extrapolation)

- $$\varepsilon_u^{\nu^*}(t) = \frac{\|\boldsymbol{u}_{\mathrm{FOM}}^{\nu^*}(t) - \boldsymbol{u}_{\mathrm{ROM}}^{\nu^*}(t)\|_{L^2(\Omega)}}{\|\boldsymbol{u}_{\mathrm{FOM}}^{\nu^*}(t)\|_{L^2(\Omega)}}$$
- $$\varepsilon_p^{\nu^*}(t) = \frac{\|p_{\mathrm{FOM}}^{\nu^*}(t) - p_{\mathrm{ROM}}^{\nu^*}(t)\|_{L^2(\Omega)}}{\|p_{\mathrm{FOM}}^{\nu^*}(t)\|_{L^2(\Omega)}}$$

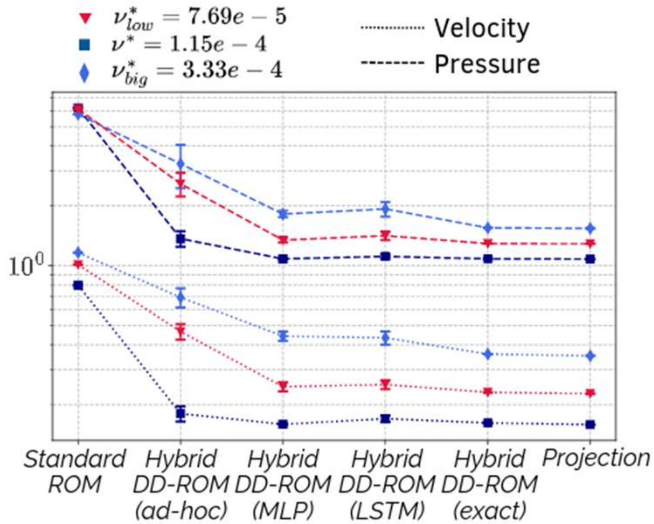


Ivagnes, A., Stabile, G., & Rozza, G. (2024). Parametric Intrusive Reduced Order Models enhanced with Machine Learning Correction Terms. arXiv preprint arXiv:2406.04169.

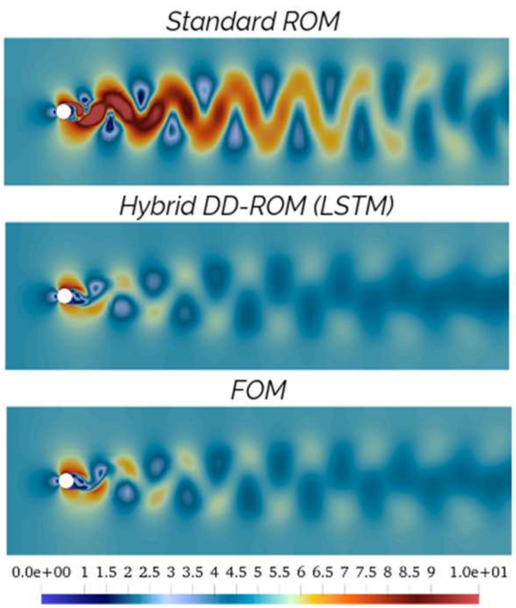
Graphical results

ANALYSIS OF GLOBAL PERFORMANCE

- *Computation of the errors' **time integrals***
- ***Graphical** velocity fields at the final instance of the online ROM simulation*



Time integrals of relative errors

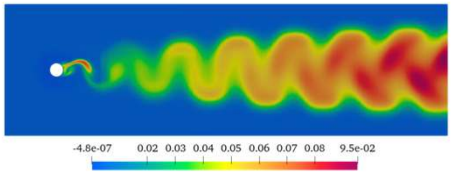


Ivagnes, A., Stabile, G., & Rozza, G. (2024). Parametric Intrusive Reduced Order Models enhanced with Machine Learning Correction Terms. arXiv preprint arXiv:2406.04169.

Turbulence modeling: another approach

Eddy viscosity model: EV-ROM *(state-of-the-art)*

$$\begin{cases} \dot{\mathbf{a}} = \boldsymbol{f}(\mathbf{a}, \mathbf{b}, \mathbf{g}) \\ \boldsymbol{c}(\mathbf{a}, \mathbf{b}, \mathbf{g}) = 0 \end{cases}$$



Example of eddy viscosity field for the flow past a cylinder

- At the **full** order level:

$$\begin{cases} \dfrac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}} \otimes \overline{\mathbf{u}}) = \nabla \cdot \left[-\overline{p}\boldsymbol{I} + (\nu + \nu_t)\left(\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T \right) \right], \\ \Delta \overline{p} = -\nabla \cdot (\nabla \cdot (\overline{\mathbf{u}} \otimes \overline{\mathbf{u}})) + \nabla \cdot \left[\nabla \cdot \left(\nu_t \left(\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T \right) \right) \right]. \end{cases}$$

RANS equations with eddy viscosity modeling

- At the **reduced** order level:

$$\nu_t \sim \sum_{i=1}^{N_{\nu_t}} g_i(\boldsymbol{\mu})\, \eta_i(\mathbf{x})$$

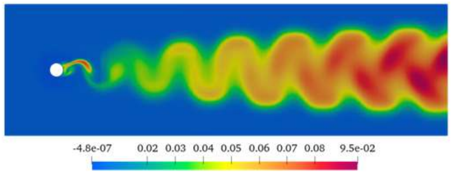
Reduced eddy viscosity

Hijazi, S., Stabile, G., Mola, A., & Rozza, G. (2020). Data-driven POD-Galerkin reduced order model for turbulent flows. Journal of Computational Physics, 416, 109513.

Turbulence modeling: another approach

Eddy viscosity model: EV-ROM *(state-of-the-art)*

$$\begin{cases} \dot{\mathbf{a}} = \boldsymbol{f}(\mathbf{a}, \mathbf{b}, \mathbf{g}) \\ \boldsymbol{c}(\mathbf{a}, \mathbf{b}, \mathbf{g}) = 0 \end{cases}$$



Example of eddy viscosity field for the flow past a cylinder

- The system is not closed: we have $N_u + N_p$ equations, but $N_u + N_p + N_{\nu_t}$ unknowns
- A ***regression map*** is used to compute the eddy viscosity coefficients

$$\begin{bmatrix} \mathbf{a} \\ \boldsymbol{\mu} \end{bmatrix} \longrightarrow \boxed{\mathscr{G}} \longrightarrow \boxed{\mathbf{g}}$$

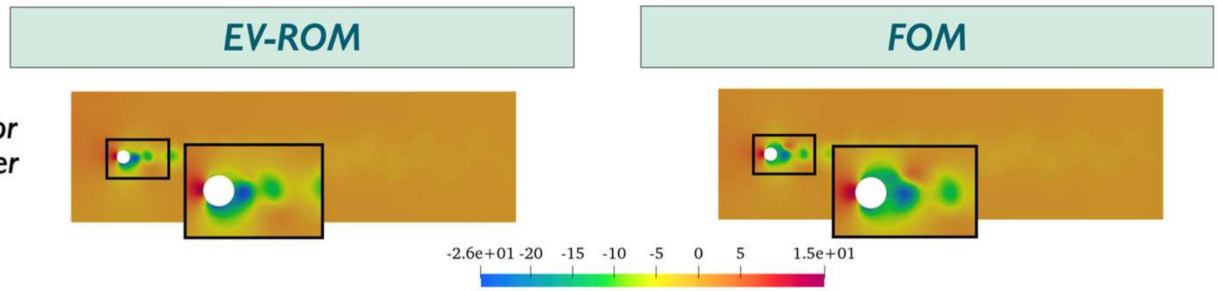
Hijazi, S., Stabile, G., Mola, A., & Rozza, G. (2020). Data-driven POD-Galerkin reduced order model for turbulent flows. Journal of Computational Physics, 416, 109513.

Turbulence modeling: another approach

Eddy viscosity model: EV-ROM *(state-of-the-art)*

$$\begin{cases} \dot{\mathbf{a}} = f(\mathbf{a}, \mathbf{b}, \mathbf{g}) \\ c(\mathbf{a}, \mathbf{b}, \mathbf{g}) = 0 \end{cases}$$

Pressure field for a test parameter

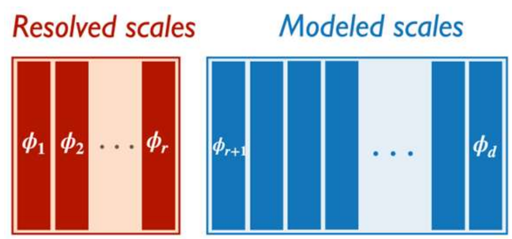


- ***Poor reconstruction*** of the fields of interest
- We need further improvement introducing a ***closure modeling***

Sequential closure modeling

Data-driven ROM: DD-EV-ROM

$$\begin{cases} \dot{\mathbf{a}} = f(\mathbf{a}, \mathbf{b}, \mathbf{g}) + \tau_u \\ c(\mathbf{a}, \mathbf{b}, \mathbf{g}) + \tau_p = 0 \end{cases}$$

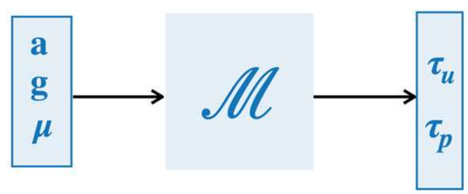


Model the contribution of the neglected scales:

- Choose a reduced dimension ***r*** and a bigger dimension ***d>r***
- Select the nonlinear operator $\mathscr{C}$
- Compute the *exact correction*

  $$\tau_{exact} = \overline{\mathscr{C}(\phi_1, \ldots, \phi_d)}^r - \mathscr{C}(\phi_1, \ldots, \phi_r)$$

- Train a map for the *approximated correction*

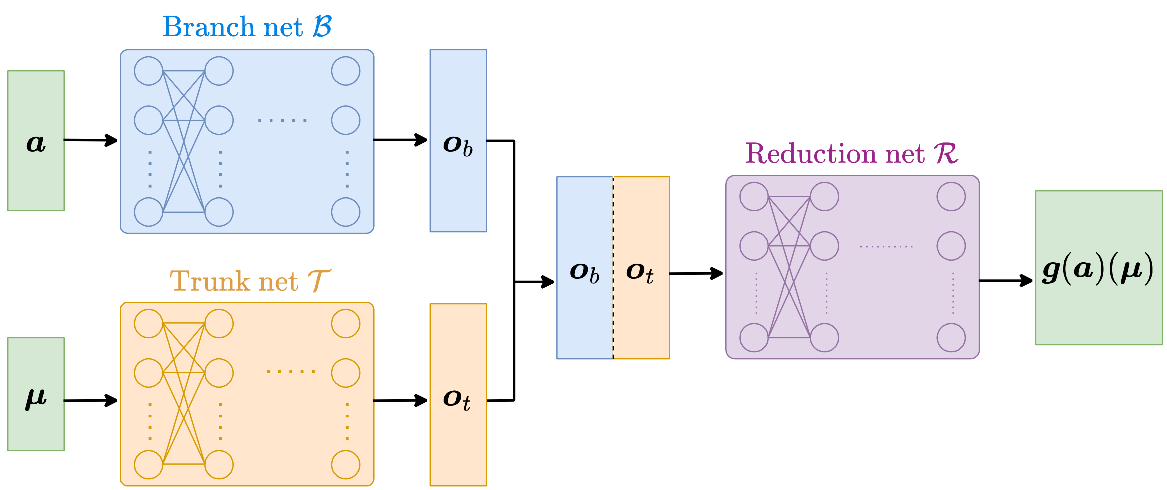


Ivagnes, A., Stabile, G., Mola, A., Iliescu, T., & Rozza, G. (2023). Pressure data-driven variational multiscale reduced order models. Journal of Computational Physics, 476, 111904.

Neural operators: the turbulence map

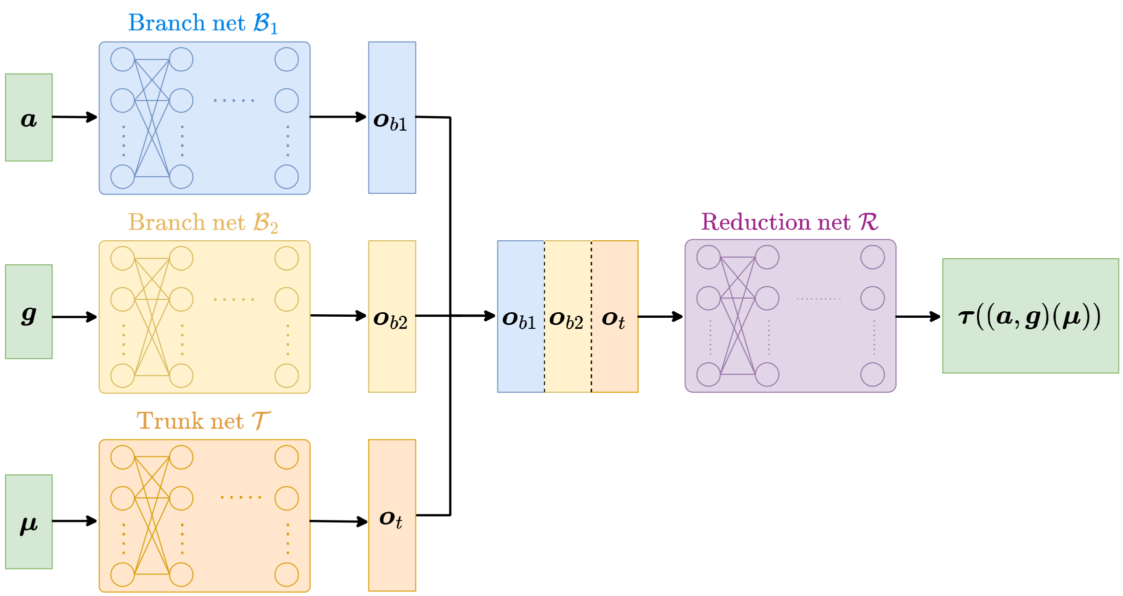
$\mathcal{G}$

- Mapping modeled through a ***DeepONet*** (deep operator network)
- Specifically *designed to learn operators* due to the sub-networks structure that separately handle the different inputs
- Minimize the discrepancy with respect to the projected eddy viscosity



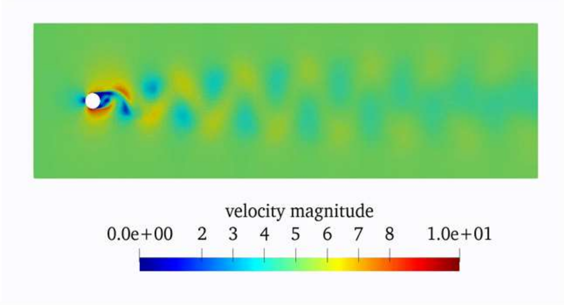
Neural operators: the closure map

- Mapping modeled through a ***MIONet*** (multi-input operator network)
- Minimize the discrepancy with respect to the exact correction

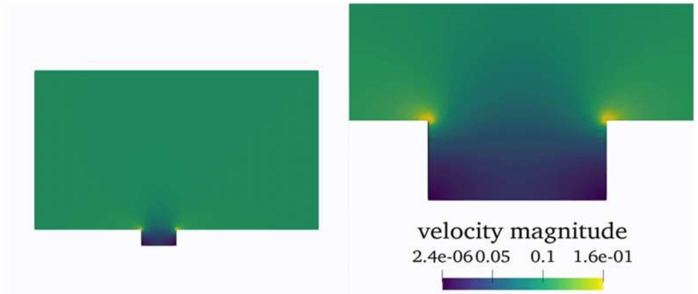


Numerical results: the test cases

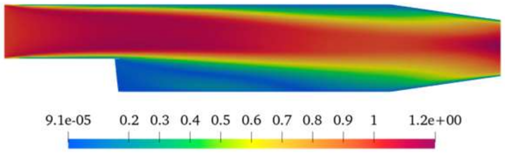
Unsteady flow past a cylinder with parameters (ν, t)

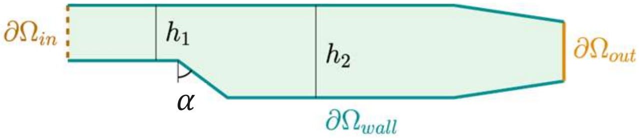


Unsteady channel-driven cavity flow with parameters (ν, t)



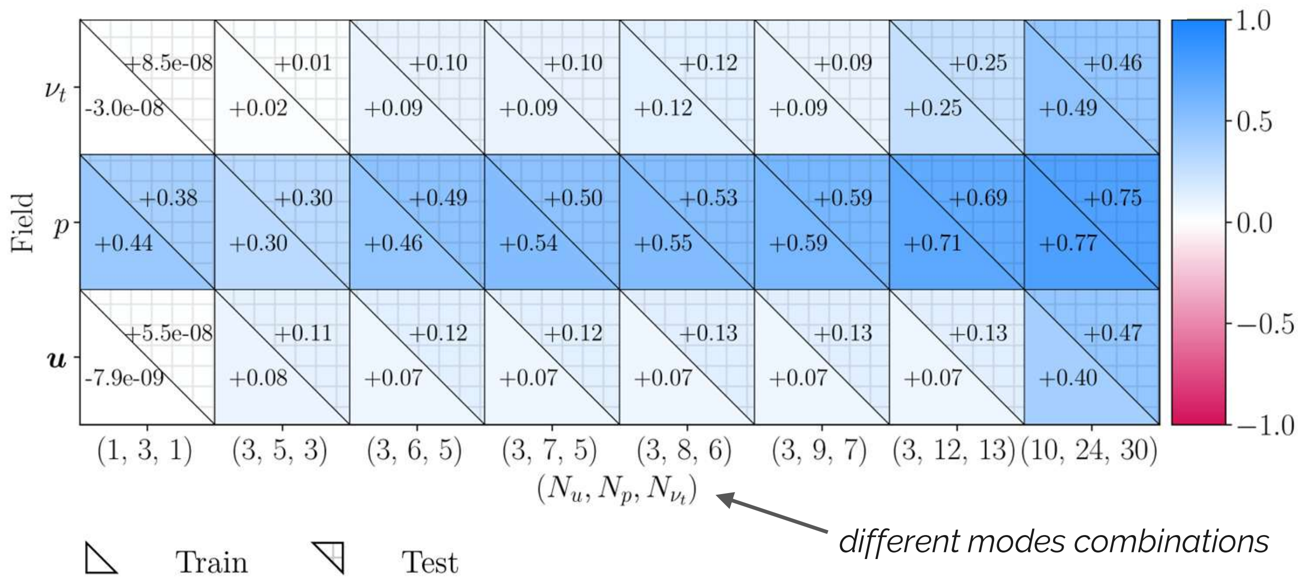
Steady backward-facing step flow with geometrical parameters (α, h_1, h_2)





Numerical results: the flow past a cylinder

Average gain in the relative error of ***DD-EV-ROM*** with respect to the state-of-the-art baseline ***EV-ROM***

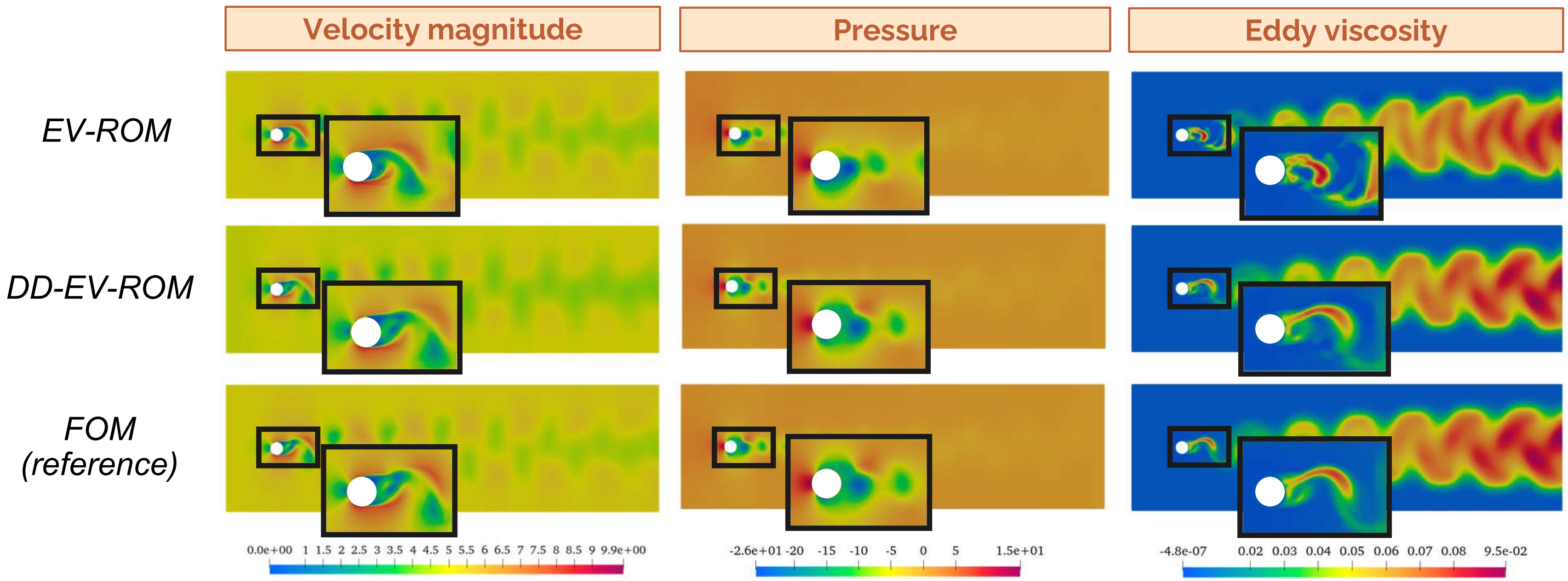


different modes combinations

- Improvement of the accuracy especially for the ***pressure***: we introduce a ***dedicated pressure closure***
- Good predictive performance in all modal regimes

Numerical results: the flow past a cylinder

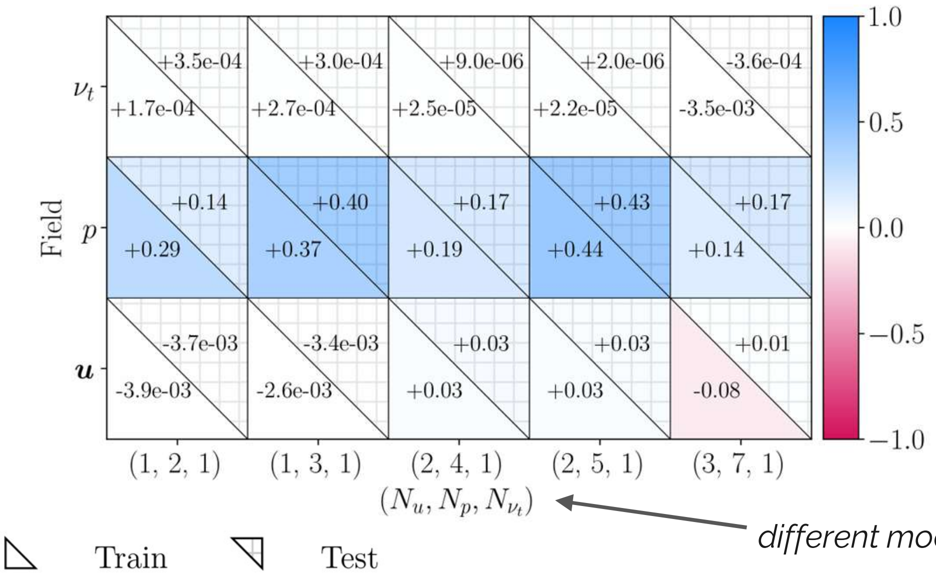
Graphical results for $(N_u, N_p, N_{\nu_t}) = (10, 24, 30)$ for a test parameter



Improved accuracy and fields' reconstruction in the novel ***DD-EV-ROM***

Numerical results: the channel-driven cavity

Average gain in the relative error of ***DD-EV-ROM*** with respect to the state-of-the-art baseline ***EV-ROM***

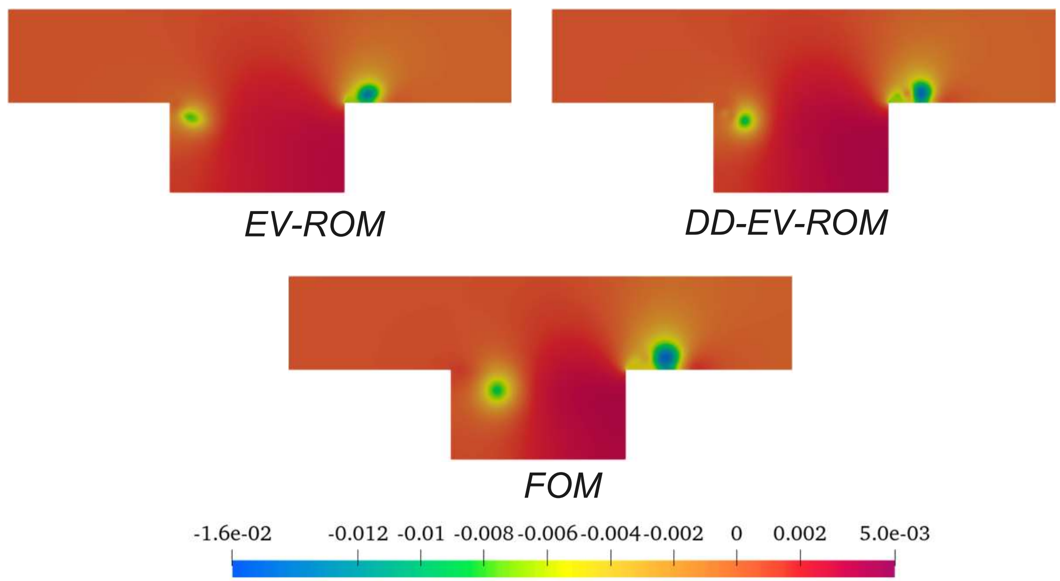


different modes combinations

- Improvement of the accuracy only for the ***pressure***: we introduce a ***dedicated pressure closure***
- Good predictive performance in all modal regimes

Numerical results: the channel-driven cavity

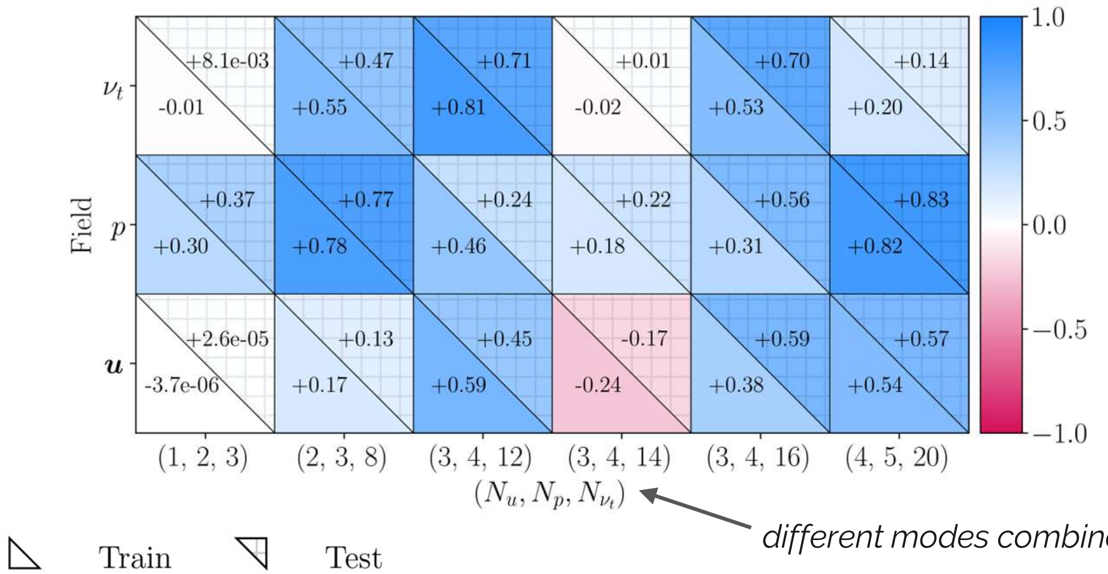
Graphical results for $(N_u, N_p, N_{\nu_t}) = (1, 3, 1)$ for a test parameter



Improved accuracy and pressure reconstruction in the novel ***DD-EV-ROM***

Numerical results: the backward-facing step

Average gain in the relative error of ***DD-EV-ROM*** with respect to the state-of-the-art baseline ***EV-ROM***

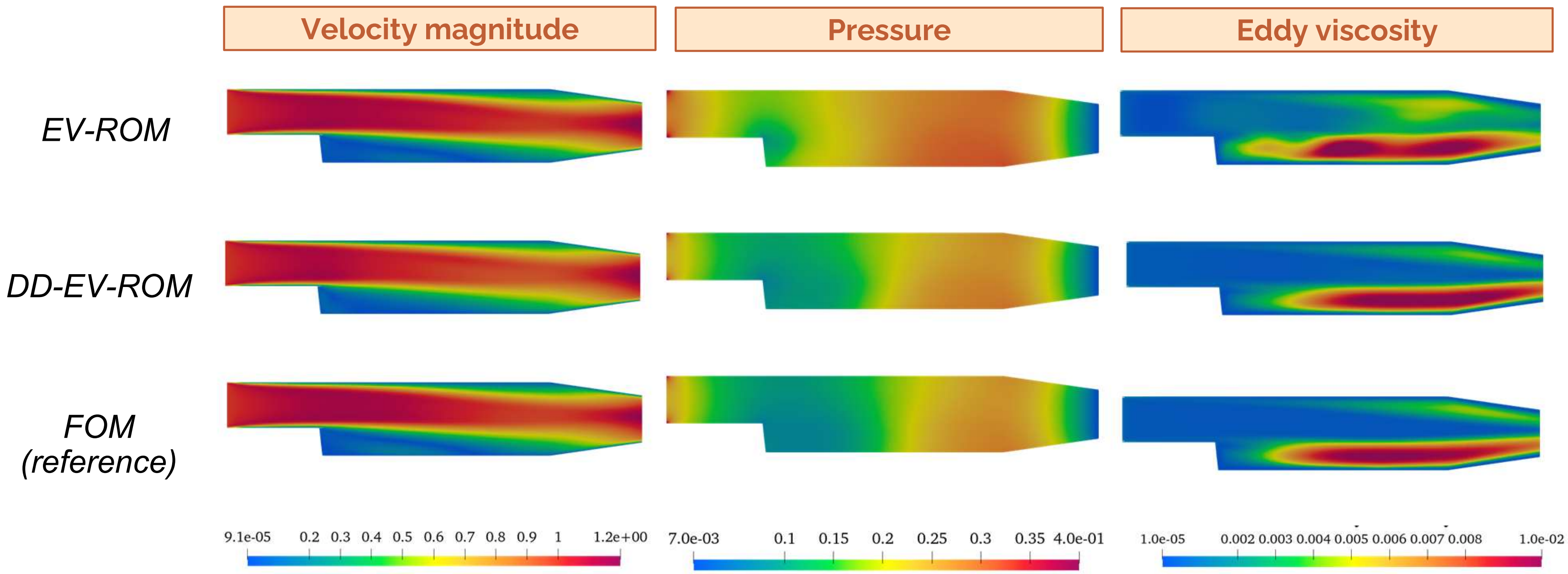


different modes combinations

- Improvement of the accuracy especially for the ***pressure*** and ***eddy viscosity*** fields
- Good predictive performance in all modal regimes

Numerical results: the backward-facing step

Graphical results for $(N_u, N_p, N_{\nu_t}) = (4, 5, 20)$ for a test parameter

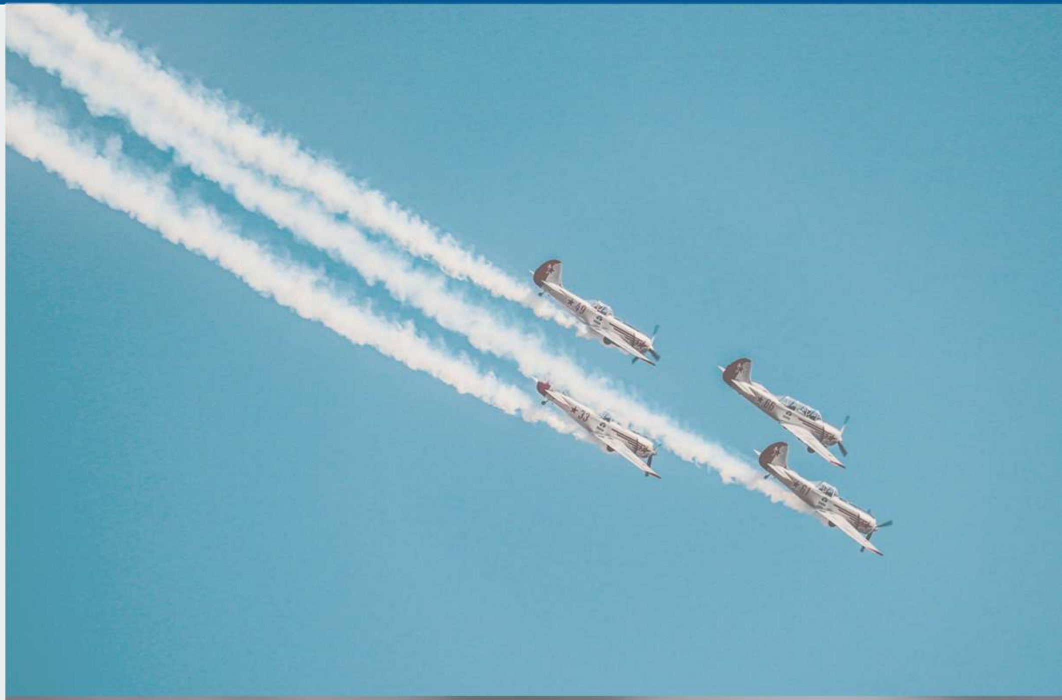


Improved accuracy and fields' reconstruction in the novel ***DD-EV-ROM***

Intrusive ROMs for turbulent and compressible problems

- How to improve ROMs in compressible flows
- ROM segregated methods
- FOM-ROM consistency

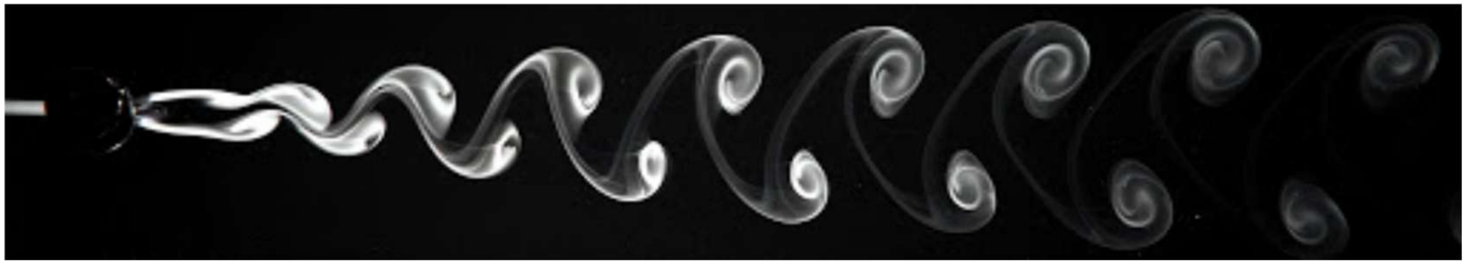
Joint work with:
Matteo Zancanaro, Giovanni Stabile



Overview of the physical problem of interest

- The scope of this work is the **resolution of parametric computational fluid dynamics problems** where an unaffordable computational cost is required to obtain accurate solutions;

- Applications of interest are spread over different fields and scales: **aerospace engineering**, **automotive industry**, **nautical studies** or **environmental fields**.



The analytical model for compressible flows

What is this problem characterized by?

- Mach number > 0.3
- **varying density field**
- **thermodynamics** for energy evolution
- no shocks
- high **turbulent** fluctuations

The Favre averaged Navier–Stokes Equations

$$\begin{cases} \nabla \cdot (\bar{\rho}\tilde{\boldsymbol{u}}) = 0 \\ \nabla \cdot [\bar{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}_{turb} - \tilde{\boldsymbol{\tau}} + \bar{p}\boldsymbol{I}] = 0 \\ \nabla \cdot \left[\bar{\rho}\tilde{\boldsymbol{u}}\left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2}\right) - \frac{C_p}{C_v}\frac{\mu}{Pr}\nabla\tilde{e} - \frac{C_p}{C_v}\frac{\mu_t}{Pr_t}\nabla\tilde{e} + \bar{p}\tilde{\boldsymbol{u}}\right] = 0 \end{cases}$$

The Favre averaging rule

$$\tilde{\Phi} = \frac{\overline{\rho\Phi}}{\bar{\rho}}, \qquad \Phi = \tilde{\Phi} + \Phi'' .$$

$$\boldsymbol{u} = \tilde{\boldsymbol{u}} + \boldsymbol{u}'' , \qquad e = \tilde{e} + e'' .$$

The Reduced SIMPLE algorithm

Why using a segregated approach at the full order level?

Iterative block solvers	Iterative segregated solvers
✗ A very big matrix has to be stored	✓ It requires 1/4 of the storage needed by block solvers
✓ The system can be solved without being modified	✗ Two decoupled equations, to be solved iteratively are needed

Why using a segregated approach also at the reduced order level?

- **consistency** between offline and online equations ⟶ **improved accuracy**
- **no saddle point formulation** ⟶ **no need for stabilization** (e.g. supremizers, etc)

The Reduced SIMPLE algorithm - part 1

Reduced approximated fields: $\tilde{\boldsymbol{u}}_r = \sum_{i=1}^{N_u} \boldsymbol{a}_i \tilde{\psi}_i = \tilde{\boldsymbol{\Psi}} \boldsymbol{a}$, $\overline{p}_r = \sum_{i=1}^{N_p} \boldsymbol{b}_i \tilde{\varphi}_i = \tilde{\boldsymbol{\Phi}} \boldsymbol{b}$, $\tilde{e}_r = \sum_{i=1}^{N_e} \boldsymbol{c}_i \tilde{\theta}_i = \tilde{\boldsymbol{\Theta}} \boldsymbol{c}$

Input: first attempt reduced velocity, pressure and energy coefficients $\boldsymbol{a}^\star$, $\boldsymbol{b}^\star$ and $\boldsymbol{c}^\star$; modal basis functions matrices $\tilde{\Psi}$, $\tilde{\Phi}$ and $\tilde{\Theta}$

Output: reduced pressure, velocity and energy fields $\overline{p}_r$, $\tilde{\boldsymbol{u}}_r$, $\tilde{e}_r$

From $\boldsymbol{a}^\star$, $\boldsymbol{b}^\star$, $\boldsymbol{c}^\star$, reconstruct $\tilde{\boldsymbol{u}}^\star$, $\overline{p}^\star$, $\tilde{e}^\star$: $\tilde{\boldsymbol{u}}^\star = \tilde{\Psi} \boldsymbol{a}^\star$, $\overline{p}^\star = \tilde{\Phi} \boldsymbol{b}^\star$, $\tilde{e}^\star = \tilde{\Theta} \boldsymbol{c}^\star$;

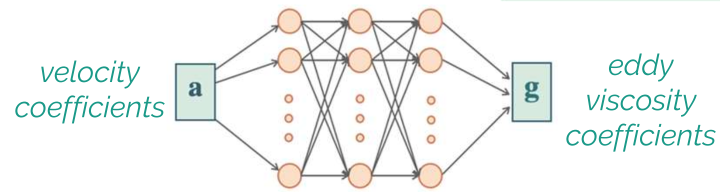
Evaluate the eddy viscosity field ν_t;

How to deal with turbulence in ROM?

$$\nu_t \sim \sum_{i=1}^{N_{\nu_t}} g_i(t, \nu)\, \eta_i(\mathbf{x})$$

Introduction of a **reduced eddy viscosity** field, computed with:

- **RBF** interpolation
- **neural network** (our choice)



Hijazi, S., Stabile, G., Mola, A., & Rozza, G. (2020). Data-driven POD-Galerkin reduced order model for turbulent flows. Journal of Computational Physics, 416, 109513.

The Reduced SIMPLE algorithm - part 2

Reduced approximated fields: $\tilde{\boldsymbol{u}}_r = \sum_{i=1}^{N_u} \boldsymbol{a}_i \tilde{\psi}_i = \tilde{\boldsymbol{\Psi}} \boldsymbol{a}$, $\overline{p}_r = \sum_{i=1}^{N_p} \boldsymbol{b}_i \tilde{\varphi}_i = \tilde{\boldsymbol{\Phi}} \boldsymbol{b}$, $\tilde{e}_r = \sum_{i=1}^{N_e} \boldsymbol{c}_i \tilde{\theta}_i = \tilde{\boldsymbol{\Theta}} \boldsymbol{c}$

. . .

3: Momentum predictor step assemble momentum equation, project it over $\tilde{\psi}_i$, solve it to obtain new reduced velocity coefficients $\boldsymbol{a}^{\star\star} \to$ reconstruct $\tilde{\boldsymbol{u}}^{\star\star}$;

4: **Energy estimation step**: assemble energy equation, project it over $\tilde{\theta}_i$, solve it to obtain new reduced energy coefficients $\boldsymbol{c}^{\star\star} \to$ reconstruct $\tilde{\boldsymbol{e}}^{\star\star}$;

5: Calculate ρ and T fields from $\overline{p}^\star$, $\tilde{\boldsymbol{u}}^{\star\star}$ and $\tilde{e}^{\star\star}$: **state equation**;

Pressure Poisson Equation (PPE)

$$\boxed{\Delta P = -\rho\left[\left(\frac{\partial u}{\partial x}\right)^2 + 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \left(\frac{\partial v}{\partial y}\right)^2\right]}$$

6: Pressure correction step: assemble pressure equation, project it over $\tilde{\varphi}_i$ to get new reduced pressure coefficients $\boldsymbol{b}^{\star\star} \to$ reconstruct $\overline{p}^{\star\star}$;

7: **if** convergence **then**

8: $\quad \tilde{\boldsymbol{u}}_r = \tilde{\boldsymbol{u}}^{\star\star}, \overline{p}_r = \overline{p}^{\star\star}, \tilde{e}_r = \tilde{e}^{\star\star}$

9: **else**

10: $\quad$ set $\tilde{\boldsymbol{u}}^\star = \tilde{\boldsymbol{u}}^{\star\star}$, $\overline{p}^\star = \overline{p}^{\star\star}, \tilde{e}^\star = \tilde{e}^{\star\star}$.

11: **end if**

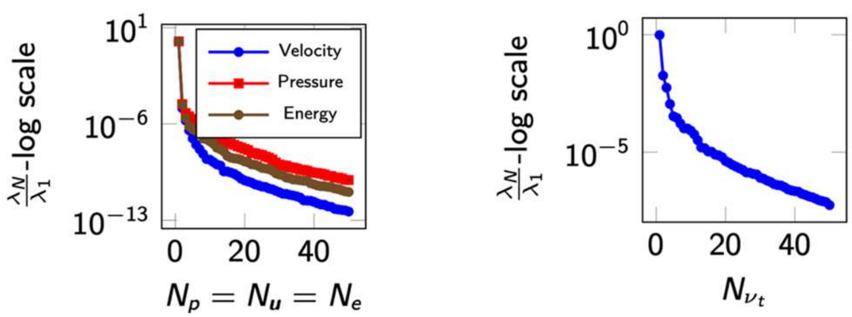
Zancanaro, M., Nkana V., Stabile, G., & Rozza, G. *Segregated methods for reduced order models, submitted 2024.*

Results - ROM with physical parameterization

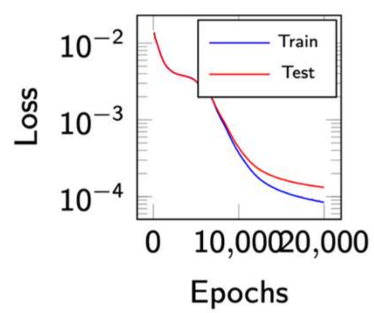
Test case: flow around a **NACA 0012** airfoil where the **viscosity** is parametrized.

Data of the problem:

- $\mu \in [10^{-5}, 10^{-2}]$, $\mu_{on} = 1.2 \times 10^{-3}$;
- Mach $= 0.73$;
- $Re \in 2.92 \times [10^4, 10^7]$;
- number of offline snapshots: $N_{off} = 50$;
- activation function of the neural network: Tanh ;
- number of epochs for training of the neural network: 2×10^3 epochs;
- reduced number of modes: $N_u = N_p = N_e = 20$;
- reduced number of modes for eddy viscosity: $N_{\nu_t} = 30$.



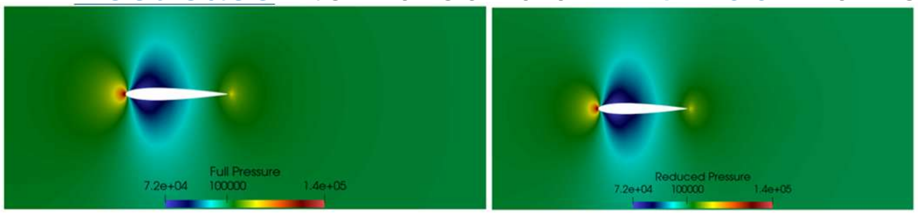
Eigenvalues decay for all the fields of interest



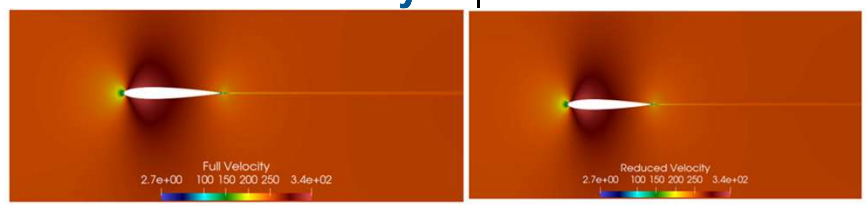
Loss of the neural network used for eddy viscosity coefficients

Results - ROM with physical parametrization

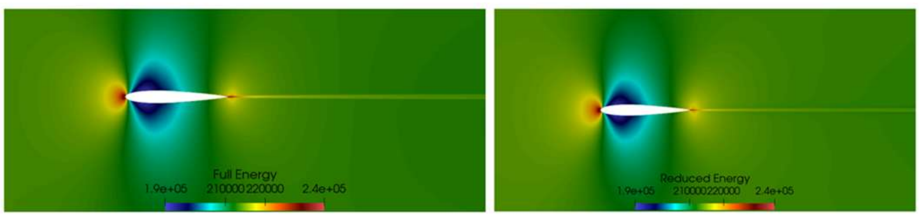
Test case: flow around a **NACA 0012** airfoil where the **viscosity** is parametrized.



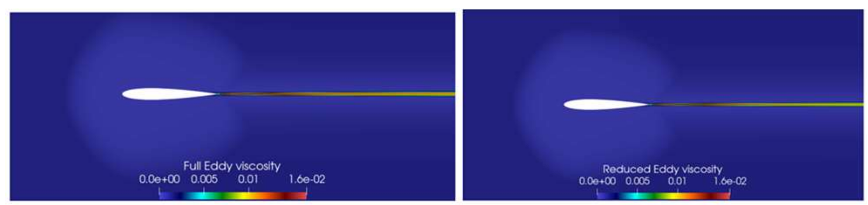
FOM-ROM (pressure)



FOM-ROM (velocity)



FOM-ROM (energy)

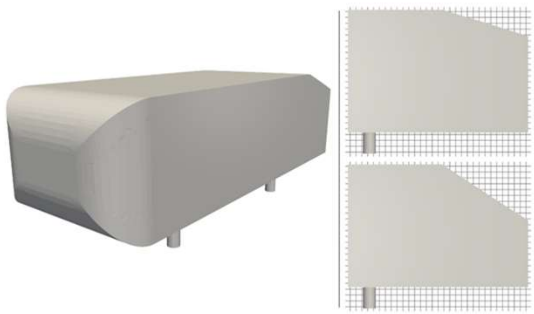


FOM-ROM (eddy viscosity)

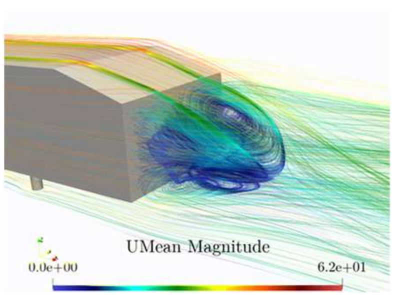
Zancanaro, M., Stabile, G., & Rozza, G. *Segregated methods for reduced order models, 2023.*

ROM with geometrical parameterization

Ahmed body test case with varying slant angles



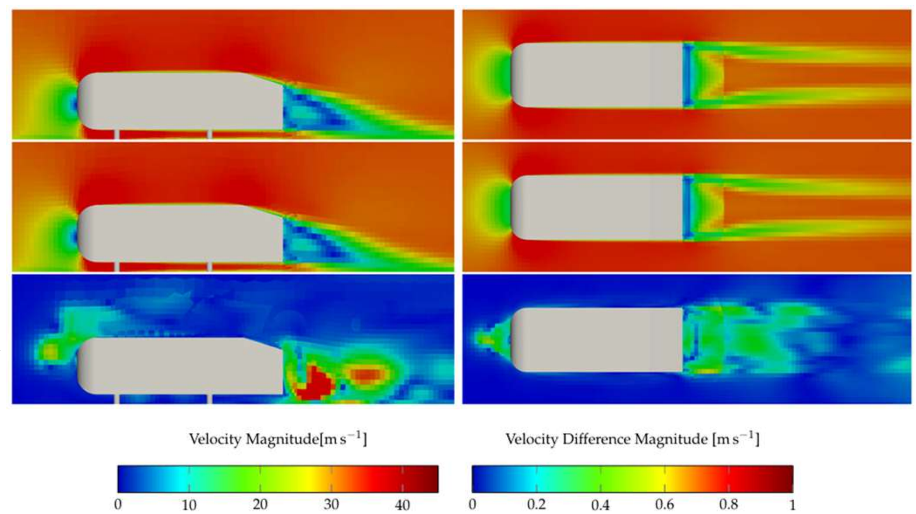
Isogeometric view of the Ahmed body and side views for minimum and maximum slant angles



FOM

ROM

Absolute error FOM-ROM



Zancanaro, M., Mrosek, M., Stabile, G., Othmer, C., & Rozza, G. (2021). *Hybrid neural network reduced order modelling for turbulent flows with geometric parameters.* Fluids, 6(8), 296.

ROM with geometrical parameterization

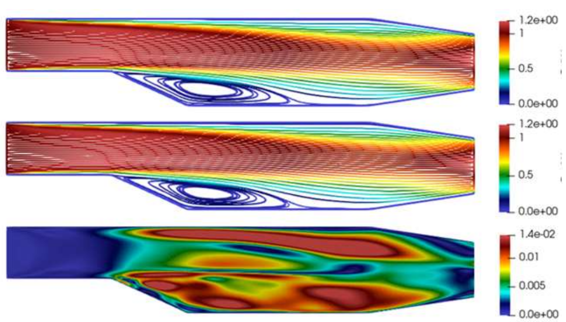
Backstep test case: the step is constructed as a **moving boundary** so that the slope β can varies

Geometry deformation in backstep channel

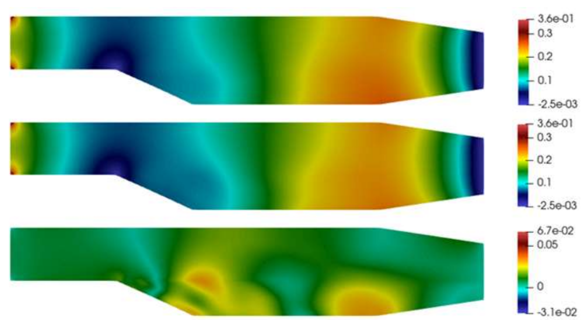
FOM

ROM

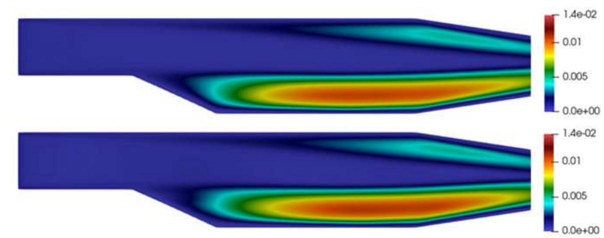
Error



***Velocity** field FOM-ROM comparison*



***Pressure** field FOM-ROM comparison*

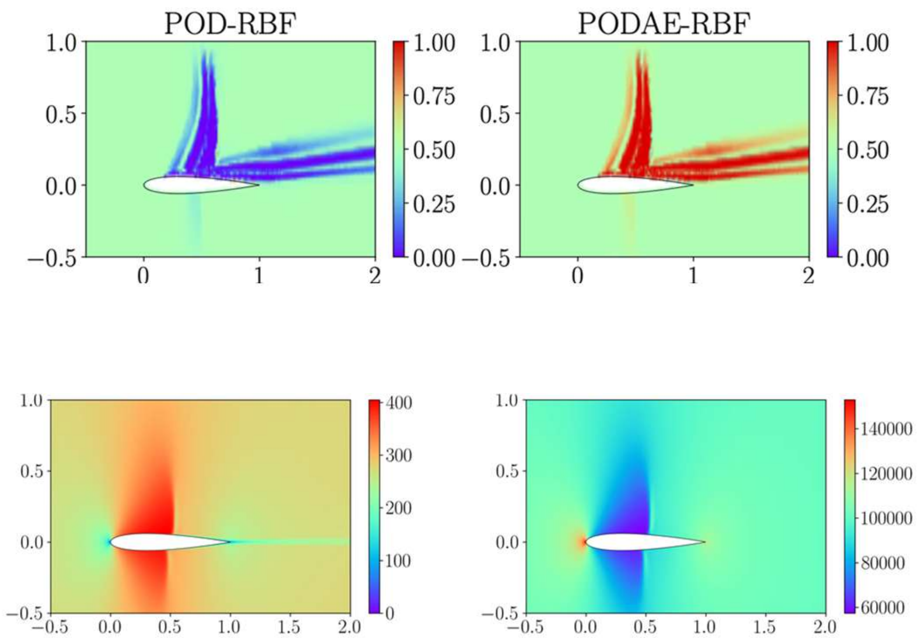


***Eddy viscosity** field FOM-ROM comparison*

Zancanaro, M., Mrosek, M., Stabile, G., Othmer, C., & Rozza, G. (2021). *Hybrid neural network reduced order modelling for turbulent flows with geometric parameters.* Fluids, 6(8), 296.

Non-Intrusive ROMs enhanced with aggregation models

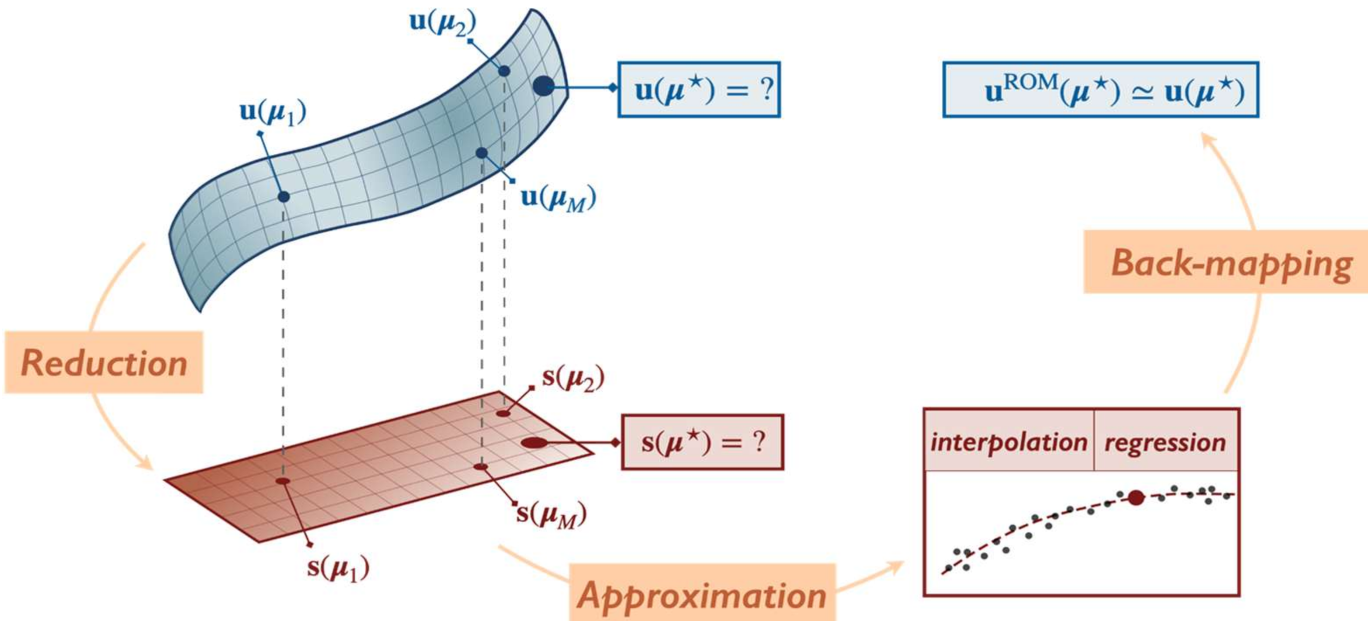
- ➔ Exploit predictions of different ROMs
- ➔ *Automatically* deduce the best model
- ➔ Associate space-dependent weights to every ROM in a *model mixture*



Joint work with:
Anna Ivagnes, Niccoló Tonicello,
Paola Cinnella *(Sorbonne University)*

Non-intrusive ROM

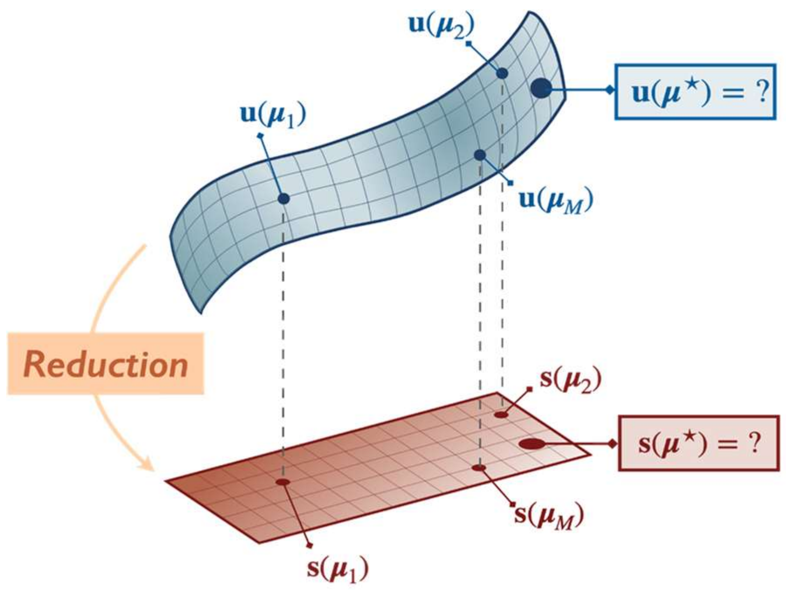
ROM approximate the high dimensional solution manifold by dimensionality reduction and perform interpolation to find the prediction for unseen parameters.



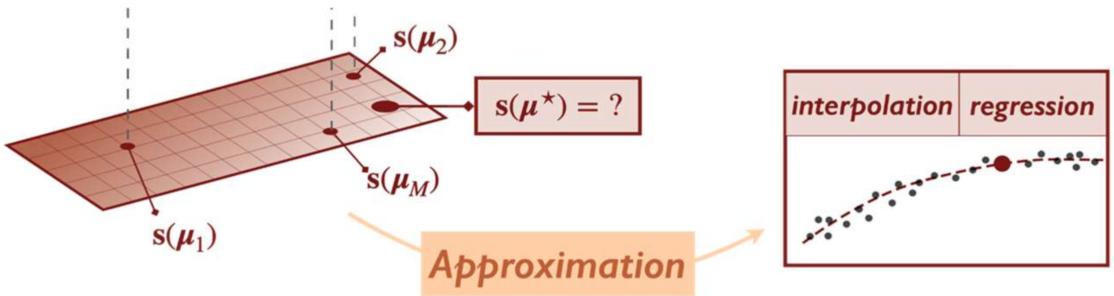
Leading motivation for *mixed-ROMs*

Individual ***reduction*** approaches are not always accurate:

- the **POD** as a **linear** reduction is inaccurate in ***advection-dominated problems*** (high Reynolds parameter) and ***nearby discontinuities*** (i.e. shocks)
- the **AE (AutoEncoder)** as a **nonlinear** approach is more accurate close to shocks but inaccurate in ***smooth regions***



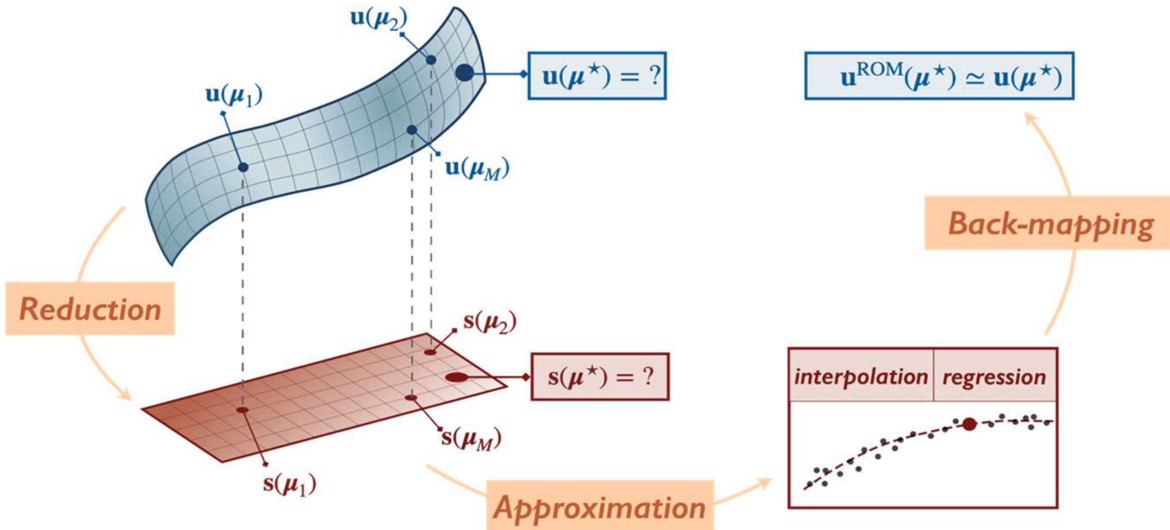
Leading motivation for *mixed-ROMs*



Individual ***approximation*** techniques are not always accurate:

- the **RBF (Radial Basis Function Interpolation)** is characterized by ***smooth interpolants***, but is ***sensitive to the basis function*** chosen;
- the **GPR (Gaussian Process Regression)** is characterized by ***automated hyperparameter tuning*** but it is ***sensitive to noisy data***;
- the **ANN (Artificial Neural Network)** can capture ***complex relationships*** in data but it is ***expensive and hard to train***.

Leading motivation for *mixed-ROMs*



Build a **database of ROMs**, combined in a ***mixed-ROM***, whose prediction is the convex combination of the individual ROMs in the database.

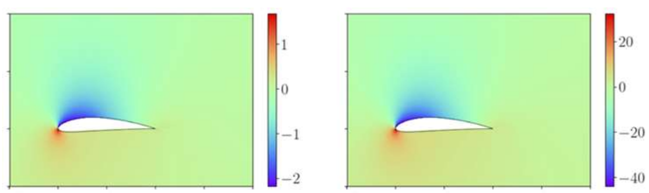
de Zordo-Banliat, M., Dergham, G., Merle, X., & Cinnella, P. (2024). *Space-dependent turbulence model aggregation using machine learning. Journal of Computational Physics*, *497*, 112628.

Cherroud, S., Merle, X., Cinnella, P., & Gloerfelt, X. (2023). *Space-dependent aggregation of data-driven turbulence models. arXiv preprint arXiv:2306.16996*.

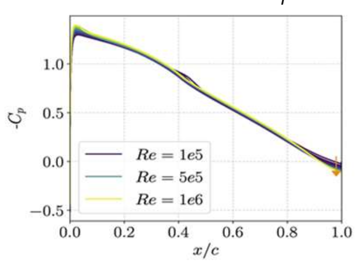
Pipeline of aggregation-ROMs

1. ***Run the FOM and build a database***
2. ***Divide the database into training, validation and test database***

Test case 1



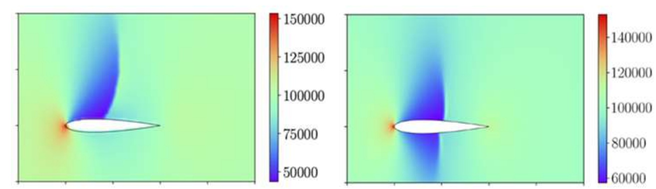
Two pressure snapshots



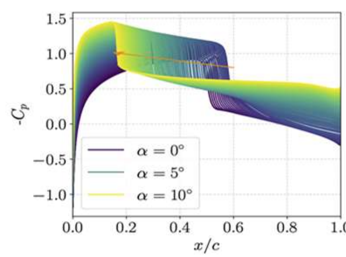
Pressure coefficient

Reynolds number varies in interval **[1e5, 1e6]**

Test case 2



Two pressure snapshots



Pressure coefficient

Angle of attack varies in interval **[0, 10] degrees**

More challenging test case: the shock position varies a lot!

Pipeline of aggregation-ROMs

Training ROMs

3. *Compute different ROMs in the training database*

Set of ROMs: $\mathscr{M} = \{M_1, M_2, \ldots, M_{n_M}\}$

- M_i is a non-intrusive ROM
- $\delta^{(i)}(\eta)$ is the prediction of M_i
- η is the set of parameters (spatial/physical)

Fields approximated with ROM:

- 1D *pressure*/*wall shear stress* on airfoil
- 2D *pressure*/*velocity magnitude* around airfoil

ROMs considered in $\mathscr{M}$:

- POD-RBF
- POD-GPR
- POD-ANN

- AE-RBF
- AE-GPR
- AE-ANN

In the case of 2D fields **AE** is replaced with **PODAE** to gain computational time

Pipeline of aggregation-ROMs

The model mixture

4. *Compute the weights associated to ROMs in the validation database*

Prediction of the aggregation model: $$\delta^{(mixed)}(\eta) = \sum_{i=1}^{n_M} w_i(\eta)\delta^{(i)}(\eta)$$

How to compute the weights?

$$\forall M_i : \quad w_i(\eta_d) = \frac{g_i(\eta_d)}{\sum_{j=1}^{n_M} g_j(\eta_d)}, \; g_i(\eta_d) = exp\left(-\frac{1}{2}\frac{(\underbrace{\delta^{(i)}(\eta_d)}_{\text{ROM prediction}} - \underbrace{\overline{\delta}_d}_{\text{FOM Snapshot}})^2}{\sigma^2}\right)$$

$(\overline{\delta}_d)_{i=1}^{N_\delta}$	Snapshots in ***validation set***

Pipeline of aggregation-ROMs

Testing the method

5. ***Predict the weights in the test database***

DATA (from validation set) $\xrightarrow[\text{Regression (RF)}]{}$ **UNKNOWN** (in test set)

$(g_i(\eta_d))_{d=1}^{N_\delta} \quad \forall M_i$ $\qquad$ $g_i(\eta^*) \quad \forall M_i$

6. ***Test the method for unseen configurations***

$$\delta^{(mixed)}(\eta^*) = \sum_{i=1}^{n_M} w_i(\eta^*)\, \delta^{(i)}(\eta^*)$$

UQ analysis

Consider the prediction as a random variable

Expected value:

$$E[\hat{\delta}(\eta)] = \delta^{(mixed)}(\eta) = \sum_{i=1}^{n_M} w_i(\eta)\delta^{(i)}(\eta)$$

Variance:

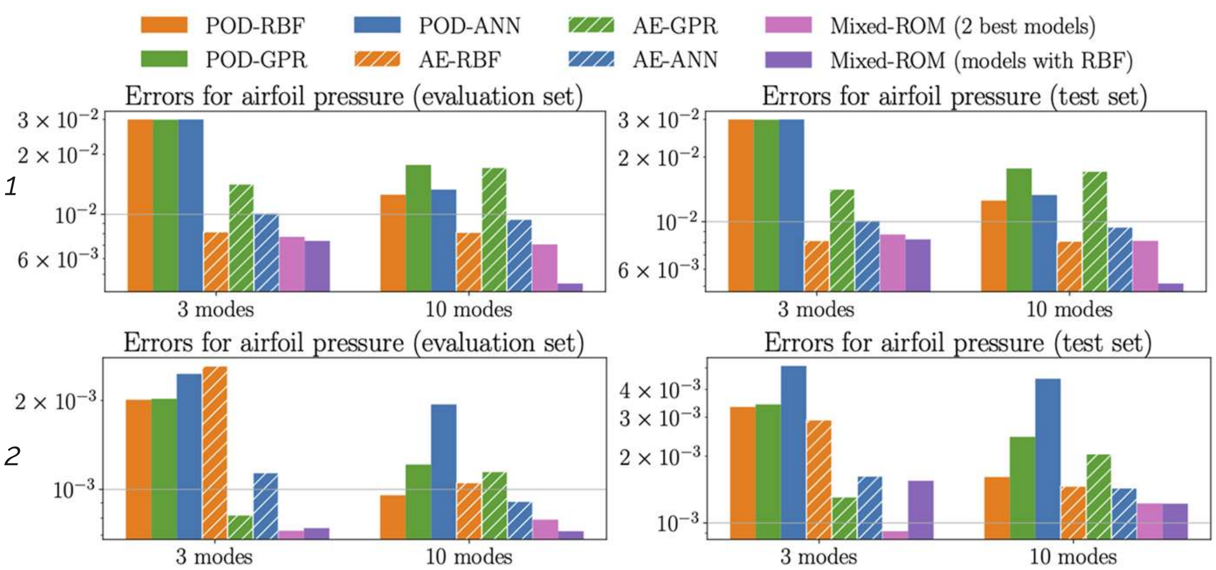
$$Var[\hat{\delta}(\eta)] = \sum_{i=1}^{n_M} w_i(\eta)\left(\delta^{(i)}(\eta) - E[\hat{\delta}(\eta)]\right)^2$$

Results of aggregation model

Relative errors for 1D airfoil pressure

Test case 1

Test case 2



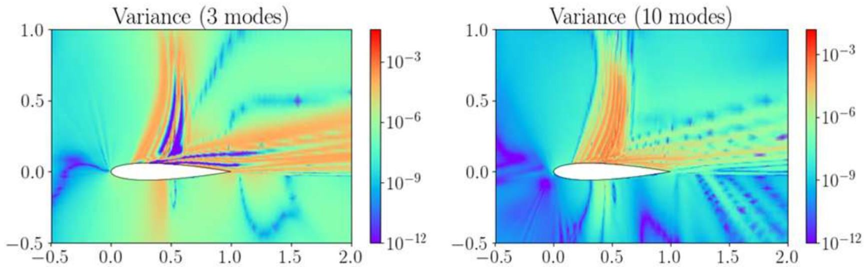
- ***Latent dimensions***: 3 and 10
- Improvement of accuracy in:
  - **validation set** (guaranteed by mathematical law)
  - **test set** (depends on the regression model)

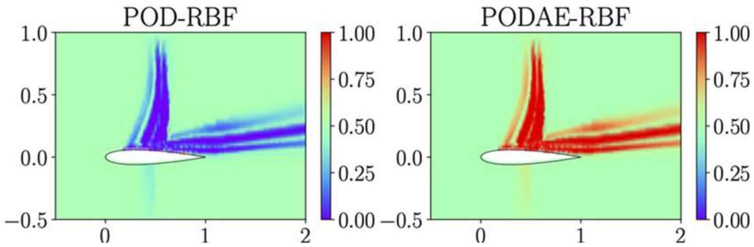
Ivagnes, A., Tonicello N., Cinnella P., and Rozza G., *Enhancing non-intrusive Reduced Order Models with space-dependent aggregation methods,* Acta Mechanica, 2024.

Results of aggregation model

Results for a test parameter (test case 2)

The weights are higher for the AE **nearby the shock position** and **nearby the wake**, where the nonlinear reduction is more accurate





Weights spatial distribution (3 modes)

The **variance** gives information on

- *consensus* among ROMs in space
- *deviation* of mixed-ROMs with respect to individual models

Ivagnes, A., Tonicello N., Cinnella P., and Rozza G., *Enhancing non-intrusive Reduced Order Models with space-dependent aggregation methods,* accepted in Acta Mechanica, 2024.

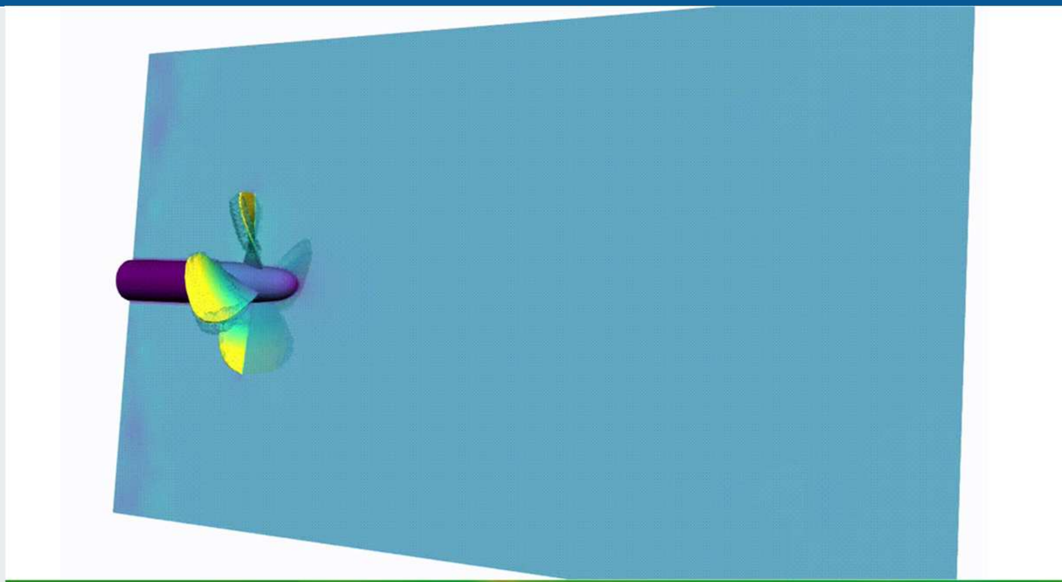
Conclusions

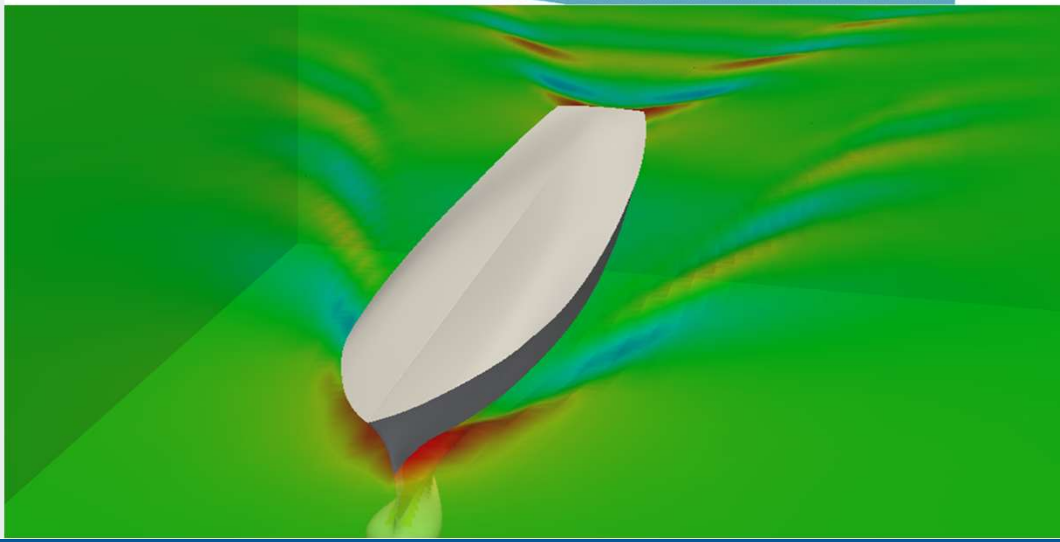
- We saw different techniques to enhance the results obtained in both **intrusive** and **non-intrusive** ROM frameworks;
- The accuracy is improved both with **machine learning** techniques (e.g. eddy viscosity coefficients) and stabilized **numerical algorithms** introducing **consistency FOM-ROM.**
- All techniques used in non-intrusive ROMs may have bottlenecks and can be improved by **automatically detect the approach with the best performance** (e.g. aggregation method)

Shape optimization in naval engineering

- Exploiting ROM in a shape optimization pipeline
- How to improve the efficiency in naval engineering applications?

Joint work with:
Anna Ivagnes, Nicola Demo



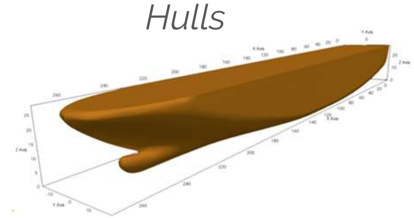


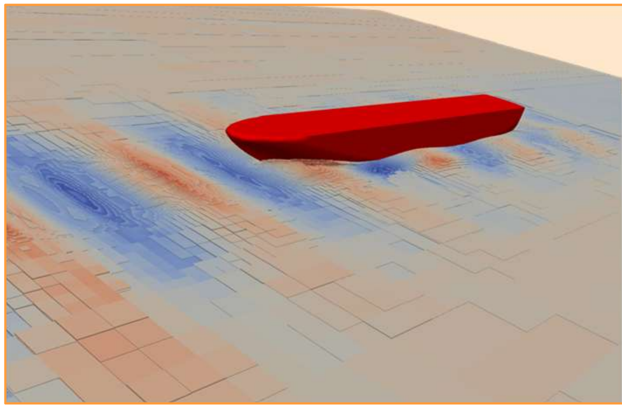
Motivation for naval design optimization

Goal:
optimize the design of a specific element of the ship to improve the performance

Propellers

Hulls

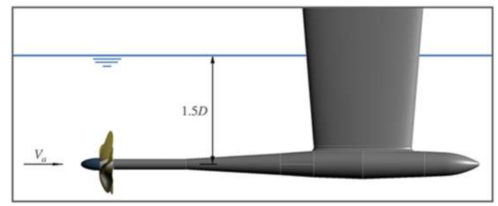




Optimization for different purposes

- Ensure comfort in yachts
- Avoid cavitation phenomena
- Increase efficiency
- Reduce vibrations

The propeller test case



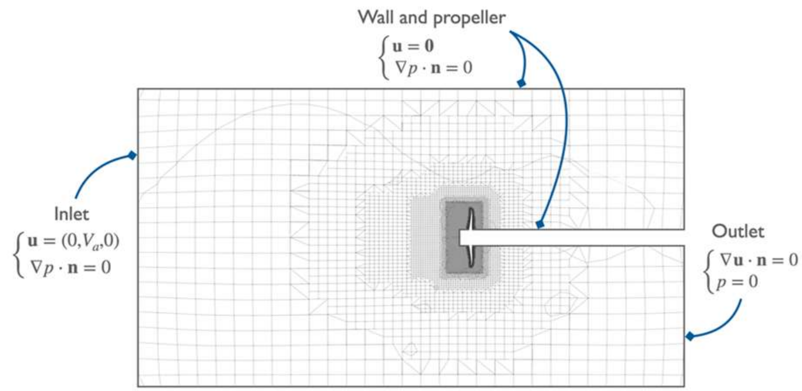
The test case: open-water tests

- Homogeneous inflow (velocity Va)
- Uniform and undisturbed flow conditions

The model: incompressible Navier-Stokes Equations

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla \cdot \nu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \nabla p \\ \nabla \cdot \mathbf{u} = \mathbf{0} \end{cases}$$

- ***Finite-Volume*** discretization
- ***RANS*** approach
- Turbulence model: κ-ω SST
- Mesh rotation: **Moving Reference Frame (MRF)**



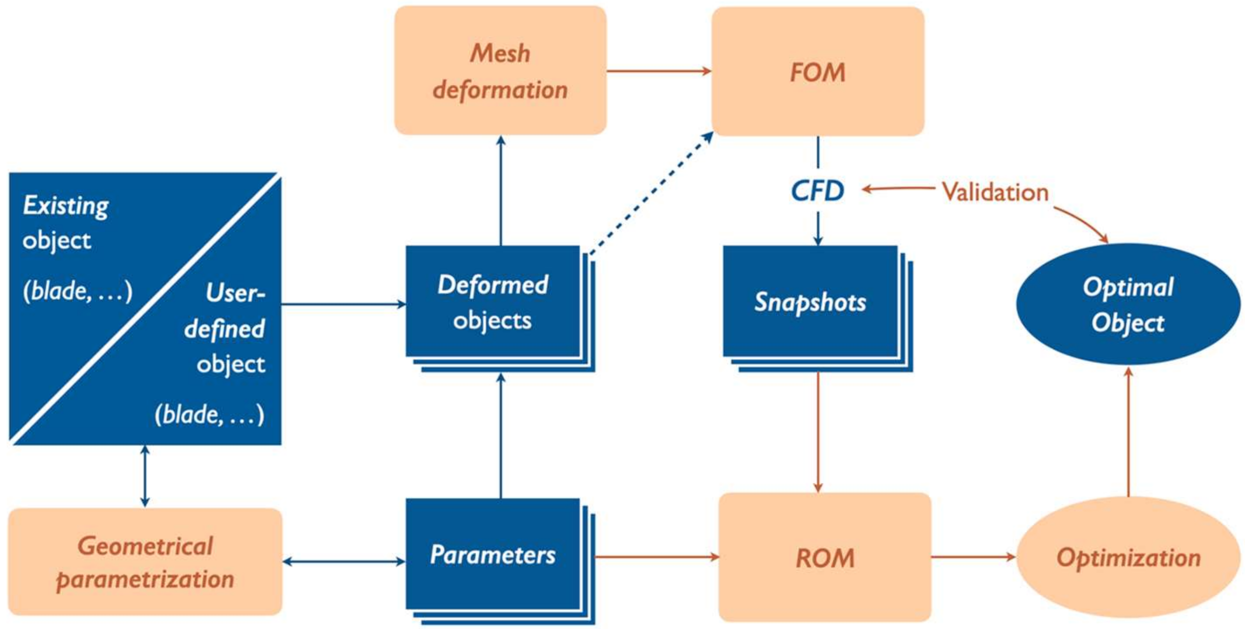
A slice of the mesh

Every simulation takes **24-48 hours** on our cluster in parallel on 55 cores

Unfeasible for optimization

A shape optimization pipeline using ROMs

A full pipeline exploiting **non-intrusive reduced order models**

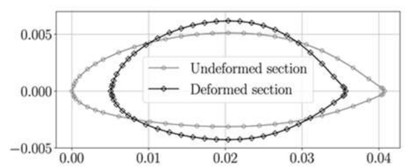


Ivagnes, Anna, Nicola Demo, and Gianluigi Rozza (2024). "A shape optimization pipeline for marine propellers by means of reduced order modeling techniques." International Journal for Numerical Methods in Engineering 125.7.

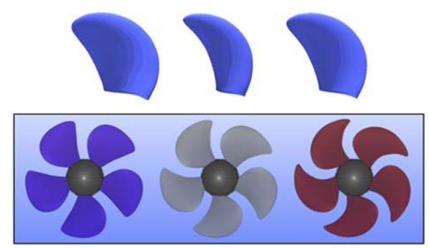
Geometric parametrization: two alternatives

Deformation through ***geometrical features*** (used for propellers)

- Select ***geometrical features*** (chord length, rake, thickness, ...)
- ***Deform the blades*** by modifying the parameters



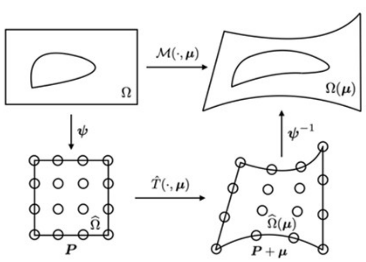
Example of section deformation


Example of blade/propeller deformation

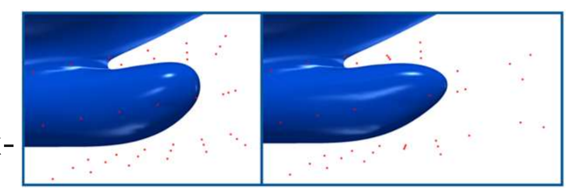
BladeX

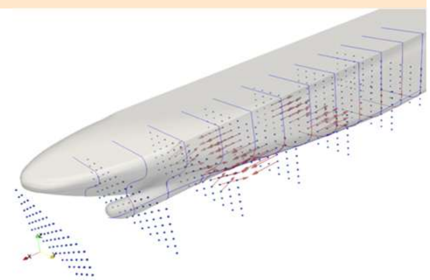
Library used

Deformation through ***Free Form Deformation*** (used for hulls)



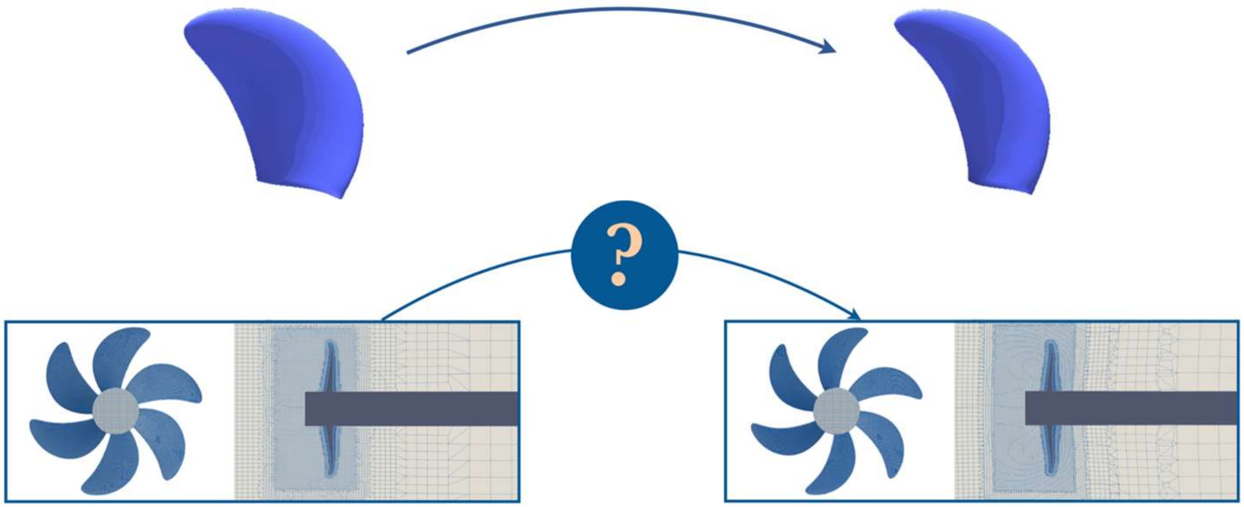
Strategy:
enclose the object in a cube, deform the cube, then back-map





Mesh deformation

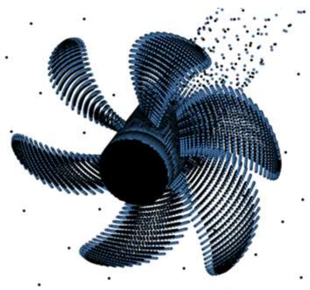
Problem: deform the mesh ***preserving the number of degrees of freedom*** in all simulations

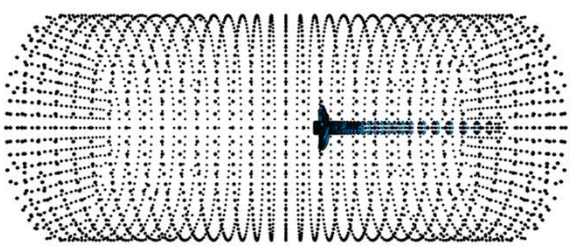


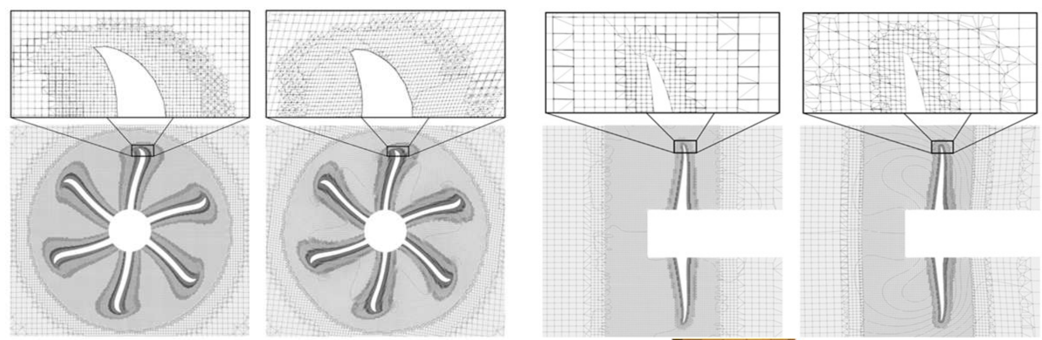
Mesh deformation

Solution: ***RBF interpolation technique***, using as ***control points*** the ***boundaries***

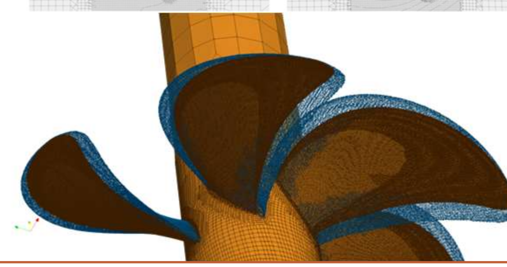
A look at the undeformed and deformed control points: blades (<u>right</u>), all boundaries (<u>below</u>).







Different deformed mesh slices (<u>above</u>); an example of mesh deformation on the propeller surface (<u>right</u>).



Ivagnes, Anna, Nicola Demo, and Gianluigi Rozza (2024). "A shape optimization pipeline for marine propellers by means of reduced order modeling techniques." International Journal for Numerical Methods in Engineering 125.7.

Non-intrusive ROM performance

Two alternative ROM approaches in optimization

Standard ROM

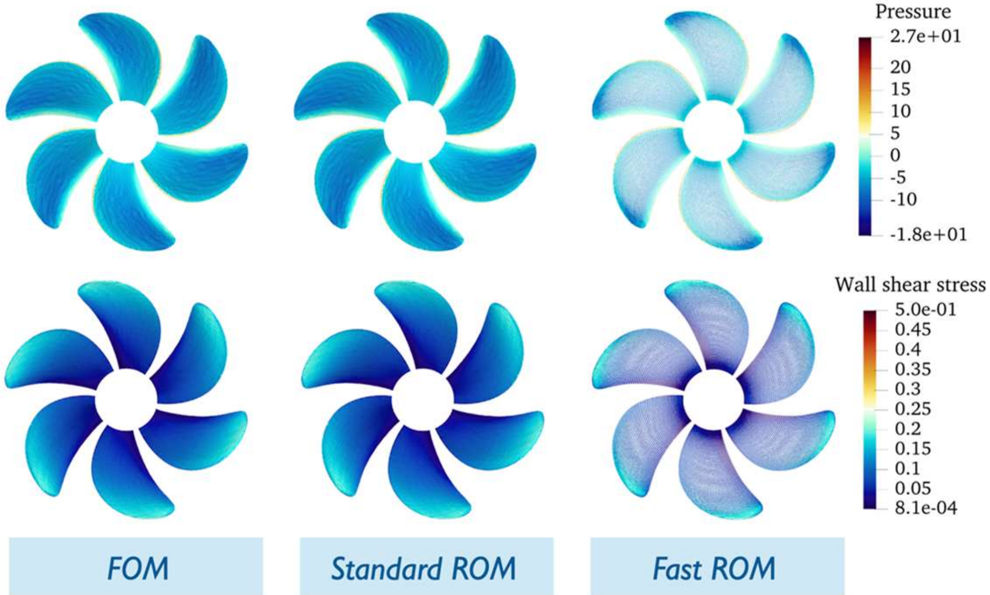
- fields evaluated at ***all blades points***
- needs to ***deform all blades points*** to compute the efficiency

5-6 minutes for each efficiency evaluation
Speed-up: ~ ***10^2***

Fast ROM

- fields evaluated at ***quadrature points***
- efficiency computed via ***quadrature formulas***

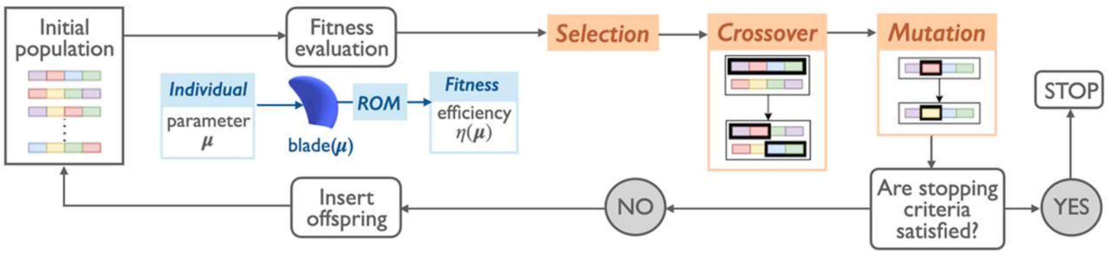
10-15 seconds for each efficiency evaluation
Speed-up: ~ ***10^5***



Ivagnes, Anna, Nicola Demo, and Gianluigi Rozza (2024). "A shape optimization pipeline for marine propellers by means of reduced order modeling techniques." International Journal for Numerical Methods in Engineering 125.7.

Optimization algorithm: results

The genetic algorithm: an evolution-inspired algorithm



Why genetic?

- Not stuck in *local minima*
- Not influenced on *initial guess*
- *Many* fitness evaluations

	Standard ROM	*Fast ROM*
Unconstrained optimization	$+5.13\,\%$	$+3.24\,\%$
Constrained optimization *(physical/geometrical constraints)*	$+0.81\,\%$	$+0.80\,\%$

Optimal blades (standard ROM)

Optimal blades (fast ROM)

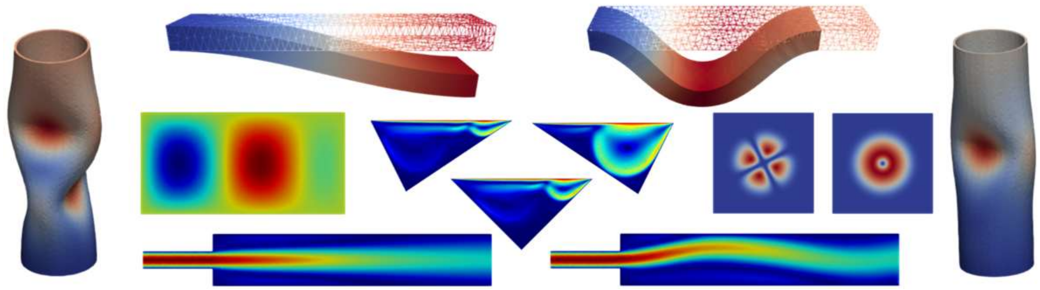
Ivagnes, Anna, Nicola Demo, and Gianluigi Rozza (2024). "A shape optimization pipeline for marine propellers by means of reduced order modeling techniques." International Journal for Numerical Methods in Engineering 125.7.

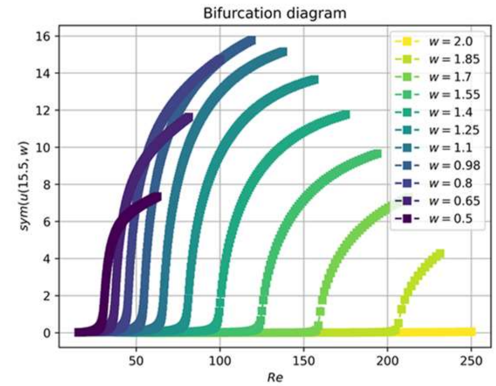
Conclusions

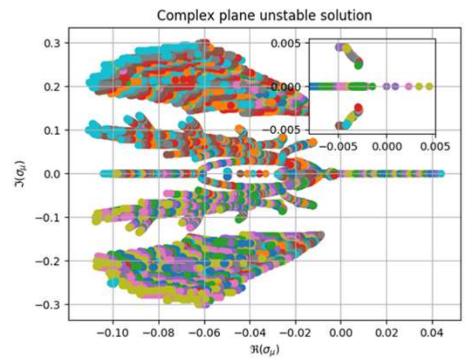
- We saw different techniques to enhance the results obtained in both **intrusive** and **non-intrusive** ROM frameworks;
- The accuracy is improved both with **machine learning** techniques (e.g. eddy viscosity coefficients) and stabilized **numerical algorithms** introducing **consistency FOM-ROM.**
- All techniques used in non-intrusive ROMs may have bottlenecks and can be improved by **automatically detect the approach with the best performance** (e.g. aggregation method)

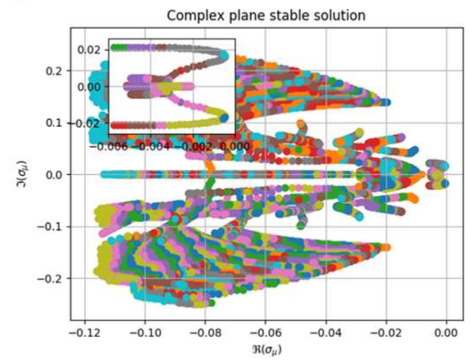
Driving bifurcating flows via optimal control problems

- ➔ Non-unique flow behavior
- ➔ Control to a desired state



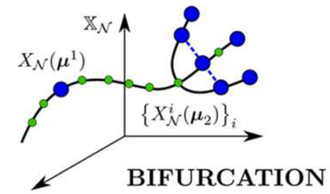




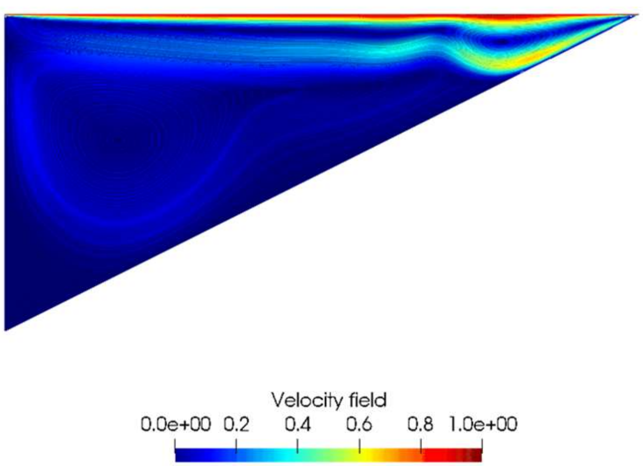


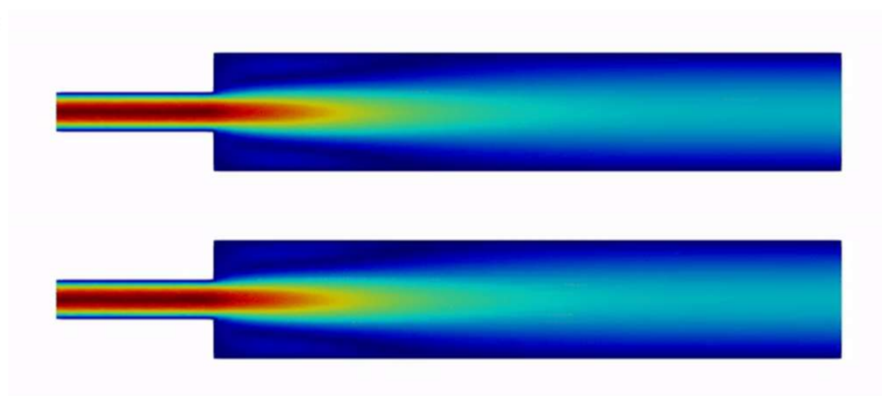
Joint work with:
Federico Pichi, Maria Strazzullo, Francesco Ballarin

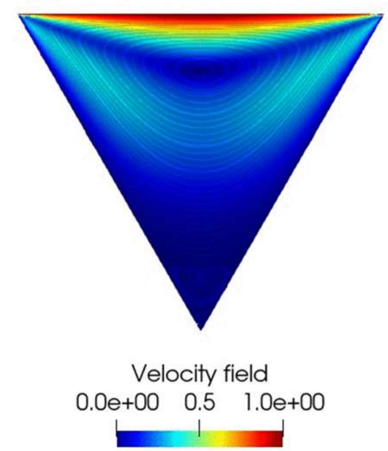
Bifurcating models in CFD



Issue: **Navier-Stokes model exhibits co-existing admissible states for relatively small Re**
e.g. Coanda Effect, Fluid-Dynamic Pinball, Triangular Cavity

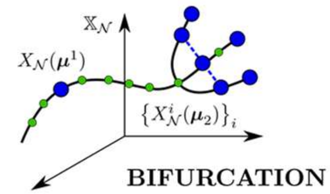




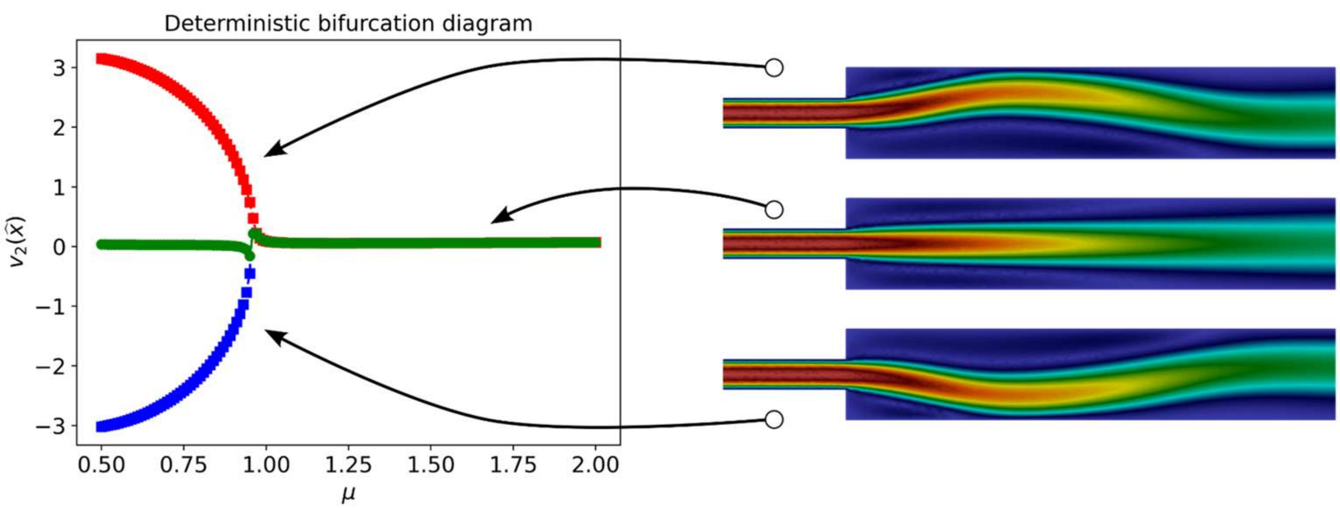


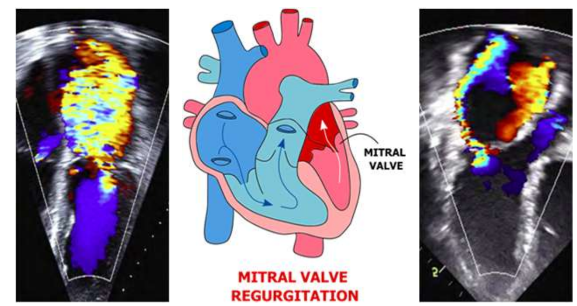
Pichi, F. et al. (2022) 'Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction', ESAIM: Mathematical Modelling and Numerical Analysis, 56(4), pp. 1361–1400. https://doi.org/10.1051/m2an/2022044

Bifurcation diagram for the Coanda effect

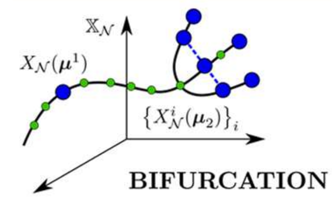


The flow exhibits a wall-hugging phenomena for viscosity values $\mu \le \mu^* \approx 0.96$





Pichi, F. et al. (2022) ‘Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction’, ESAIM: Mathematical Modelling and Numerical Analysis, 56(4), pp. 1361–1400. https://doi.org/10.1051/m2an/2022044



Steering the bifurcating behavior

Goal: Obtain a laminar flow towards the end of the channel via Optimal Control Problem

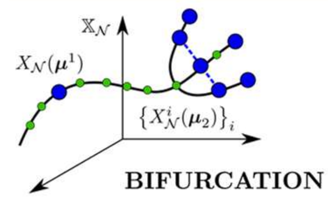
Given $\boldsymbol{\mu} \in \mathcal{P}$, $y_d \in \mathbb{Y}_{obs}(\Omega_{obs})$, $f \in \mathbb{Y}^*$, find state-control pair $(y, u) \in \mathbb{Y} \times \mathbb{U}(\Omega_u)$

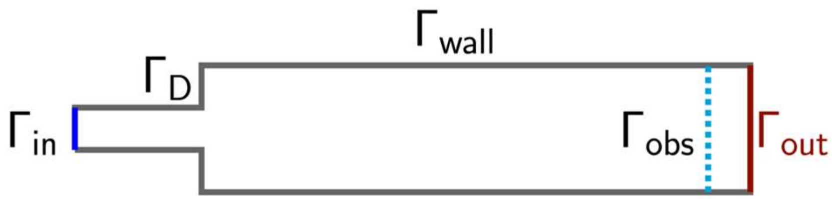
$$\min_{y \in \mathbb{Y},\, u \in \mathbb{U}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{y}_{\mathrm{d}}\|^2_{\mathbb{Y}_{\mathrm{obs}}} + \frac{\alpha}{2} \|\boldsymbol{u}\|^2_{\mathbb{U}} \quad \text{subject to} \quad \boldsymbol{G}(\boldsymbol{y};\mu) - \boldsymbol{C}(\boldsymbol{u}) - \boldsymbol{f} = \boldsymbol{0},$$

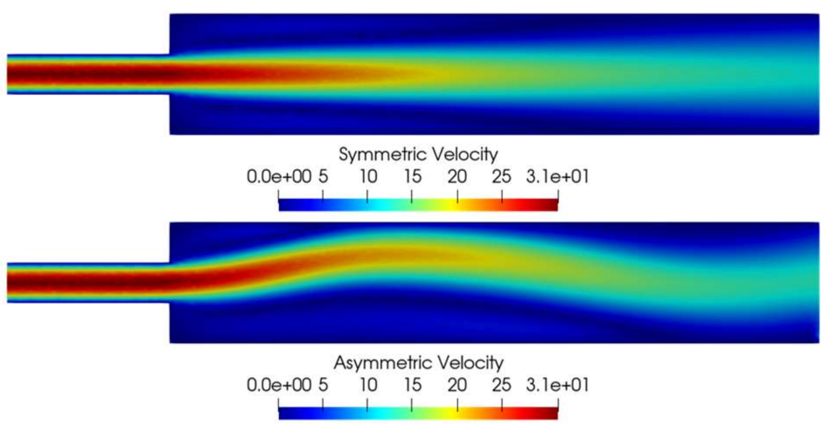
- $C \in \mathcal{L}(\mathbb{U}, \mathbb{Y}^*)$ control operator, and $G(y; \boldsymbol{\mu})$ state operator,
- $\alpha \in (0, 1]$ penalization parameter (the greater is α, the lower is the control),

Pichi, F. et al. (2022) 'Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction', ESAIM: Mathematical Modelling and Numerical Analysis, 56(4), pp. 1361–1400. https://doi.org/10.1051/m2an/2022044

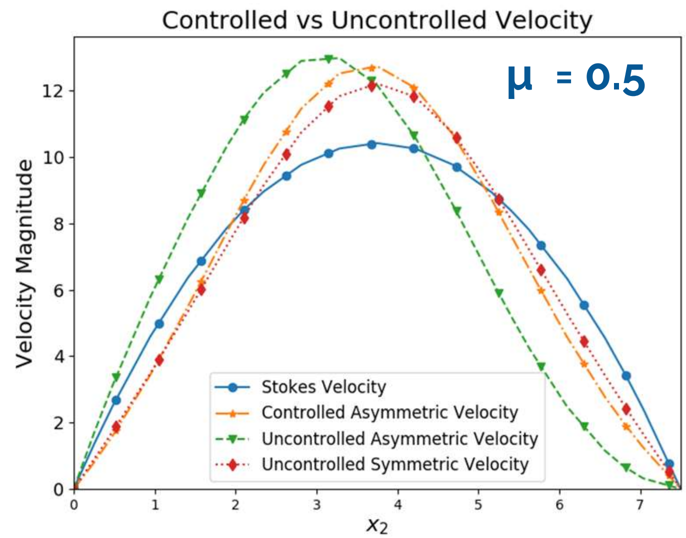
Neumann boundary control - weak action



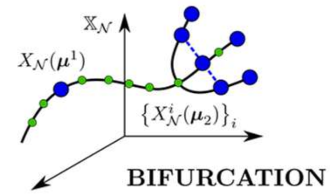




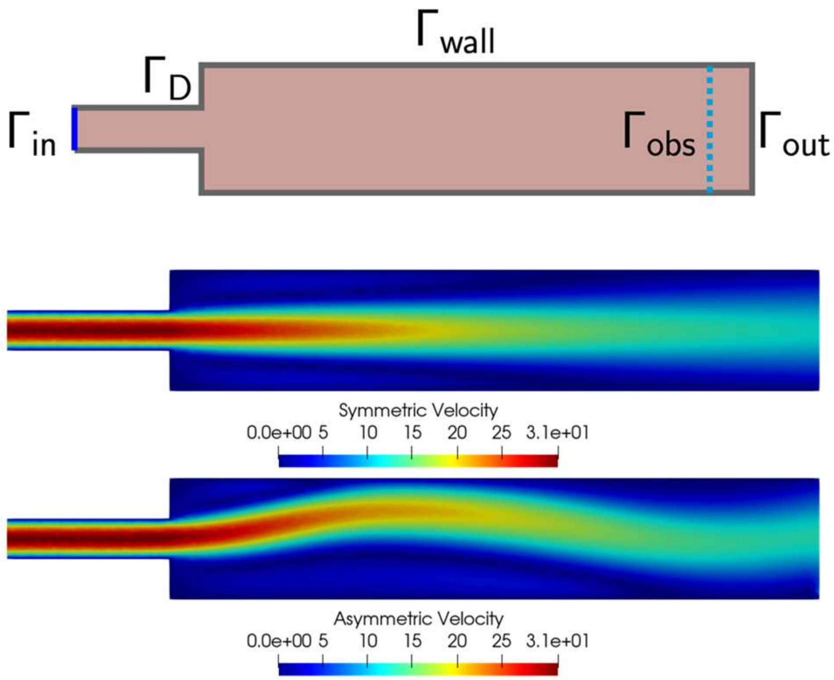
Slight effect on the bifurcating nature of the uncontrolled equation



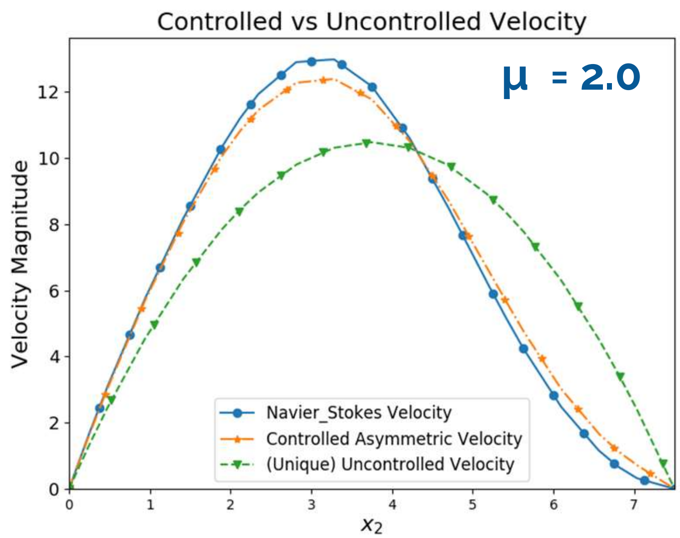
Pichi, F. et al. (2022) 'Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction', ESAIM: Mathematical Modelling and Numerical Analysis, 56(4), pp. 1361–1400. https://doi.org/10.1051/m2an/2022044



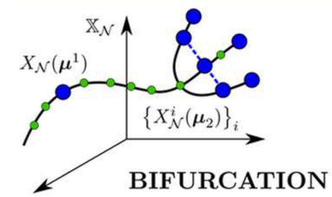
Distributed control - strong action



<u>The uncontrolled solution is heavily affected by acting on the forcing term</u>

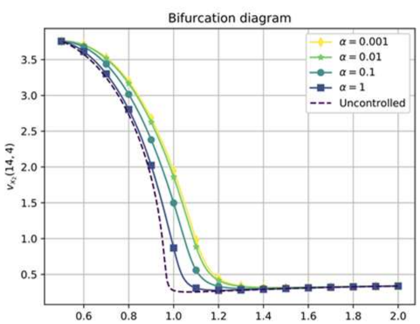


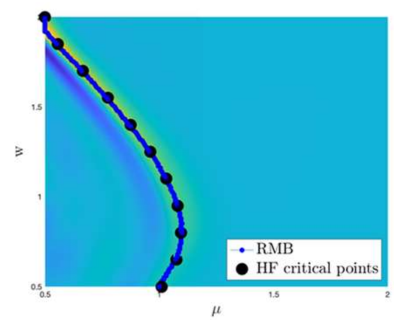
Pichi, F. et al. (2022) 'Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction', ESAIM: Mathematical Modelling and Numerical Analysis, 56(4), pp. 1361–1400. https://doi.org/10.1051/m2an/2022044

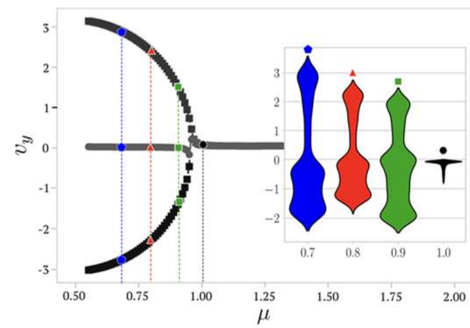


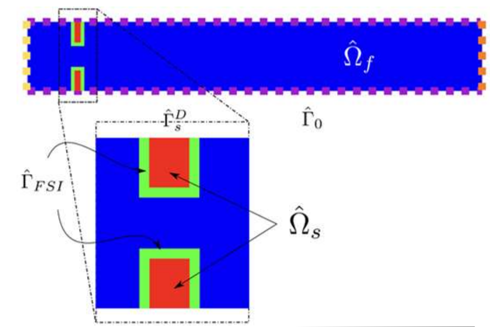
Further implications and extensions

1. Influencing the position of the **bifurcation point** via the penalization parameter α
2. Exploit **ML-enhanced ROMs** for the detection of the **bifurcating curve**
3. **Stochastic perturbation** to approximate unknown states via **Polynomial Chaos**
4. Interaction with **buckling** of elastic structures in the **FSI** scenario







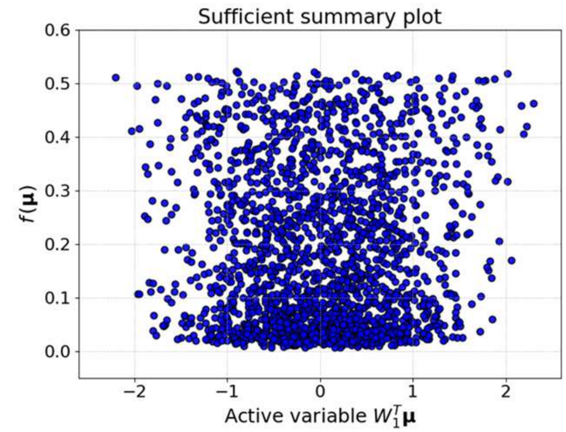


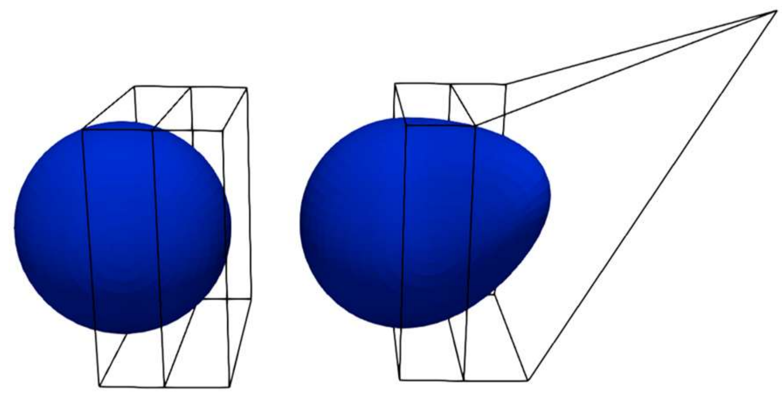
1. [Pichi 2022] '<u>Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction</u>', **M2AN**
2. [Pichi 2023] '<u>An artificial neural network approach to bifurcating phenomena in computational fluid dynamics</u>', **Computers & Fluids**
3. [Pichi 2024] '<u>A graph convolutional autoencoder approach to model order reduction for parametrized PDEs</u>', **Journal of Computational Physics**
4. [Pintore 2021] '<u>Efficient computation of bifurcation diagrams with a deflated approach to reduced basis spectral element method</u>', **Advances in Computational Mathematics**
5. [Gonnella 2024] '<u>A stochastic perturbation approach to nonlinear bifurcating problems</u>'**. arXiv 2402.16803**
6. [Khamlich 2022] '<u>Model order reduction for bifurcating phenomena in fluid-structure interaction problems</u>', **International Journal for Numerical Methods in Fluids**

Generative models for shape deformation

- ➔ Constrained generative models
- ➔ Free Form Deformation strategy
- ➔ Quantify model uncertainty

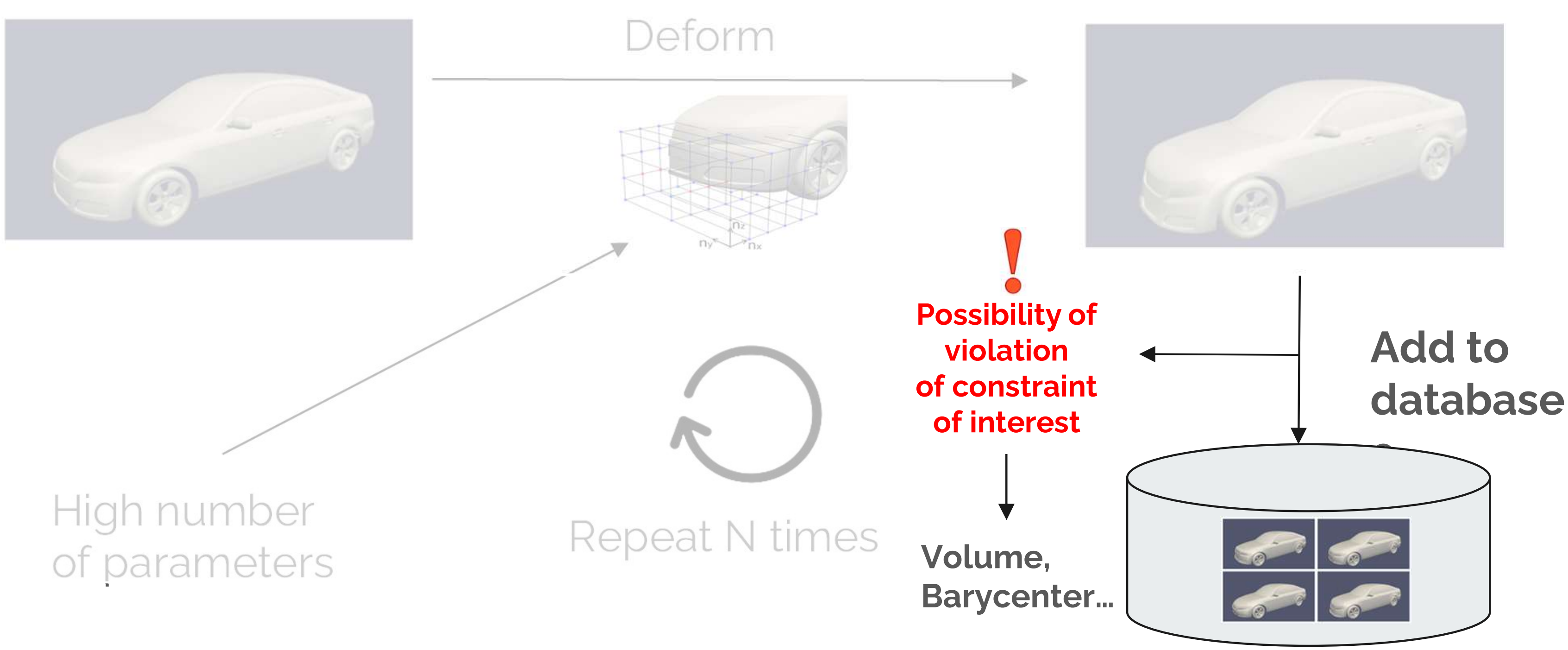
Joint work with:
Guglielmo Padula, Francesco Romor, Giovanni Stabile





Examples:

Free Form Deformation



Constrained Free Form Deformation

Constraint by Optimization

- Iterative solution
- Universally applicable
- Satisfied within certain bounds
- Slow

Examples:

- Genetic Algorithms (GA)
- Gradient Descent
- **Free Form Deformation + GA**

Constraint by Design

- One shot solution
- Problem dependent
- Is satisfied within numerical errors
- Fast

Examples:

- Least Squares
- **Constrained Free Form Deformation**

But still high number of parameters → Curse of Dimensionality

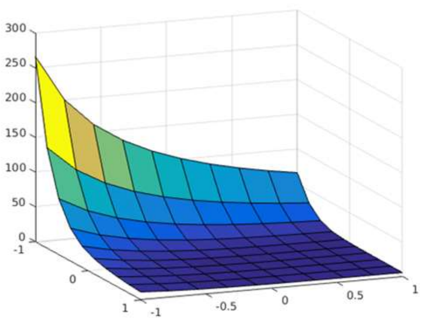
References:

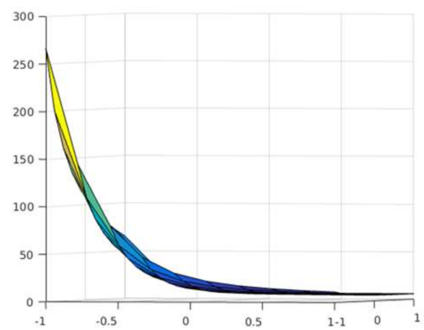
1. Hanmann, G. Bonneau PG., Barbier S., Elber G., Hagen H. (2012). Volume Preserving FFD for Programmable Graphics Hardware. The Visual. Computer.

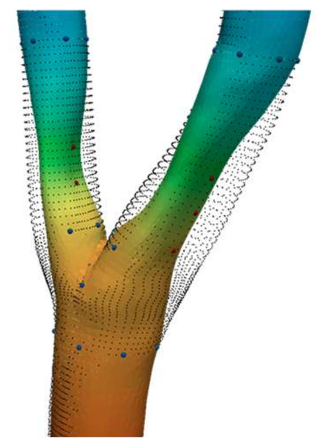
#Shape parametrization #Active Subspaces #POD-Galerkin

Combined Parameter and model Reduction

with Marco Tezzele and Francesco Ballarin







Active subspaces property

In many cases the dimension of the parametrised problem is only artificially high

- Active subspaces property identifies a **set of important directions** in the space of all inputs

$$f : \mathbb{R}^m \to \mathbb{R} \qquad \mathbf{x} \in \mathbb{R}^m$$

$$\mathbf{C} = \mathbb{E}\left[\nabla_{\mathbf{x}} f \, \nabla_{\mathbf{x}} f^T\right] = \int (\nabla_{\mathbf{x}} f)(\nabla_{\mathbf{x}} f)^T \rho \, d\mathbf{x}$$

$$\mathbf{C} = \mathbf{W} \mathbf{\Lambda} \mathbf{W}^T$$

f is a **scalar function** that takes as arguments the parameters x

C is the **uncentered covariance matrix** of the gradients of f, **symmetric, positive semidefinite**

E is the expected value and rho a probability density function

- We define the active subspace to be the range of the **first n eigenvectors** of W

$$\mathbf{W} = [\mathbf{W_1} \quad \mathbf{W_2}] \in \mathbb{M}^{m \times m} \qquad \mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda_1} & \\ & \mathbf{\Lambda_2} \end{bmatrix}$$

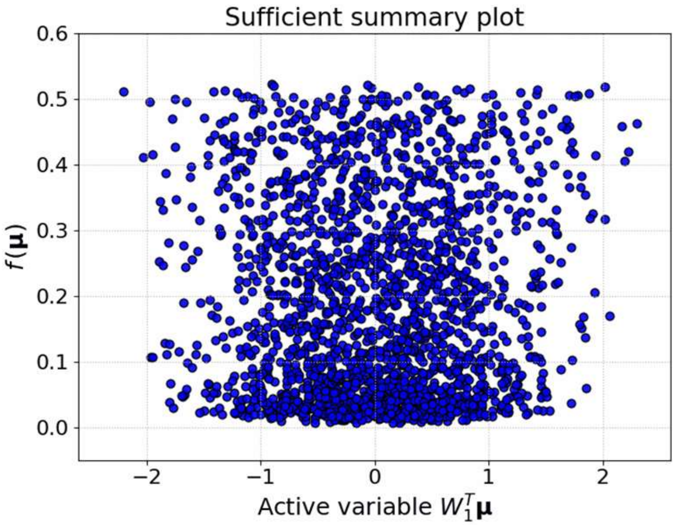
- With the basis identified, we can map forward to the active subspace. So **y is the active variable** and z the inactive one. The **surrogate model** g is used to approximate f

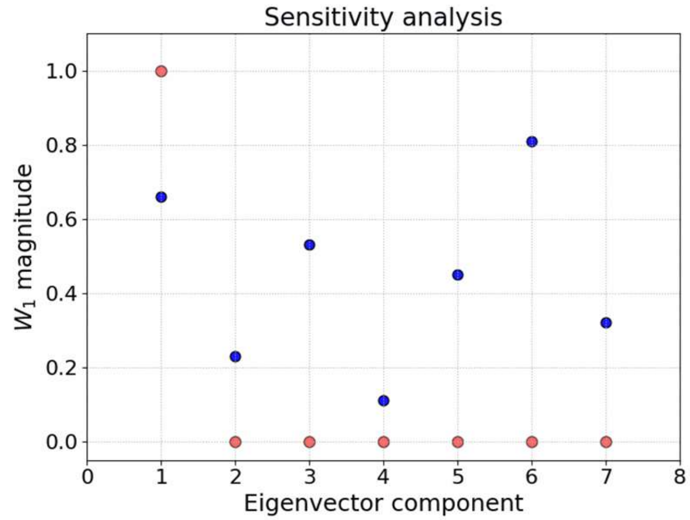
$$\mathbf{y} = \mathbf{W_1^T} \mathbf{x} \in \mathbb{R}^n \qquad \mathbf{z} = \mathbf{W_2^T} \mathbf{x} \in \mathbb{R}^{m-n} \qquad f(\mathbf{x}) \approx g(\mathbf{W_1^T} \mathbf{x}) = g(\mathbf{y})$$

References:

1. Constantine. “Active subspaces: Emerging ideas for dimension reduction in parameter studies.” SIAM, 2015.

Active subspaces - A quadratic example





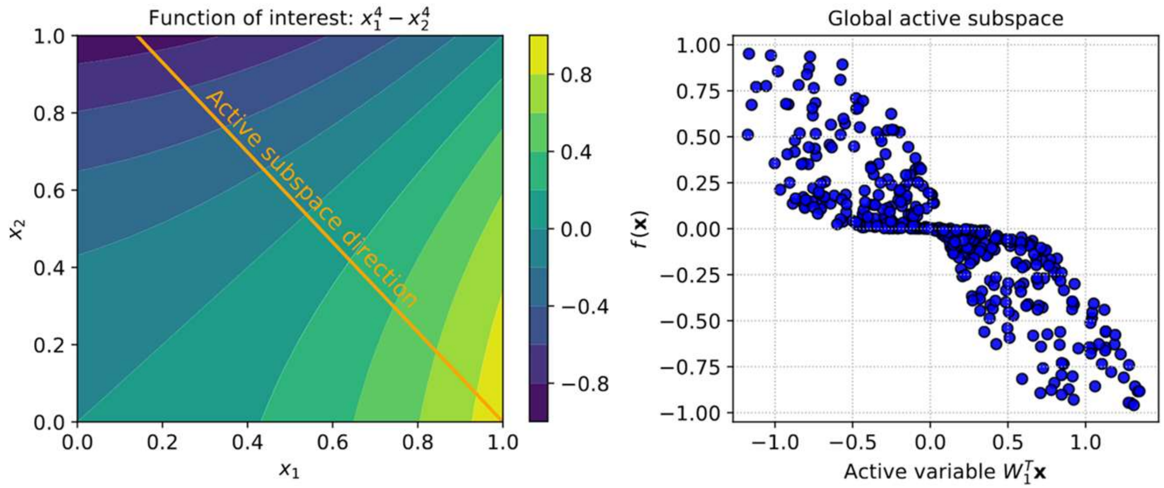
References:

1. M. Tezzele, F. Ballarin and G. Rozza "Combined parameter and model reduction of cardiovascular problems by means of active subspaces and POD-Galerkin methods". SEMA-SIMAI Springer Series 2018

2. M. Tezzele, F. Salmoiraghi, A. Mola, G. Rozza. "Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems". AMSES 2018

Local Active Subspaces

Local active subspaces exploits locality by finding clusters for better function variability.

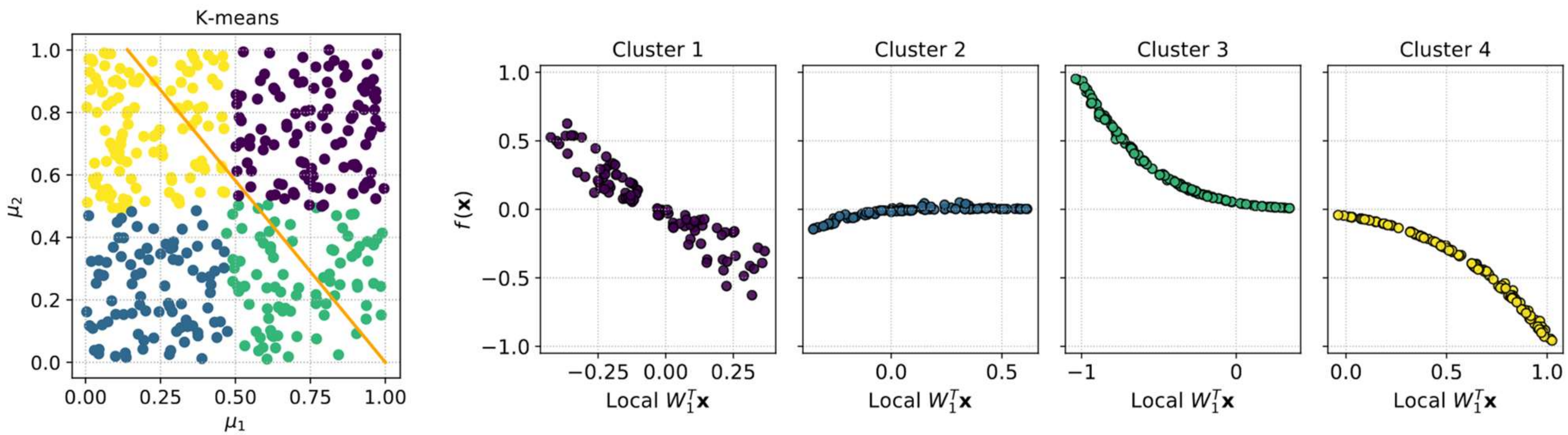


References:

1. Romor, F., Tezzele M., Mrosek M., Othmer C., Rozza G. (2023). Multi-fidelity data fusion through parameter space reduction with applications to automotive engineering. International Journal for Numerical Methods in Engineering.
2. Rozza G., Stabile G., Ballarin F. (2022). Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics.

Local Active Subspaces

Local active subspaces exploits locality by finding clusters for better function variability.

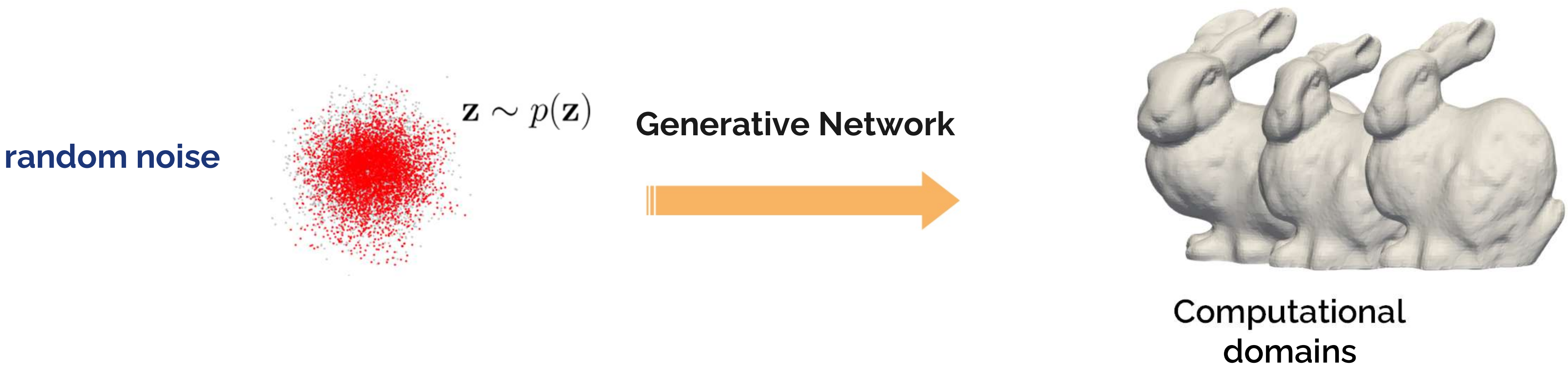


References:

1. Romor, F., Tezzele M., Mrosek M., Othmer C., Rozza G. (2023). Multi-fidelity data fusion through parameter space reduction with applications to automotive engineering. International Journal for Numerical Methods in Engineering.
2. Rozza G., Stabile G., Ballarin F. (2022). Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics.

Generative Models - Quantify Model Uncertainty

- → Generative modelling learns **probability distributions on the data.**
- → **A priori uncertainty quantifications** can be done with probability distributions.
- → **Learning distribution of computational domains.**



Constrained Generative Models

We adopt generative models for **sampling new geometries.**

The geometries are parametrized using the **reduced latent space**

- ➔ alternative to **Active Subspaces**, with the advantage of having the **same parameterization** for every function of interest.

The advantages with respect to Constrained Free Form Deformation are:

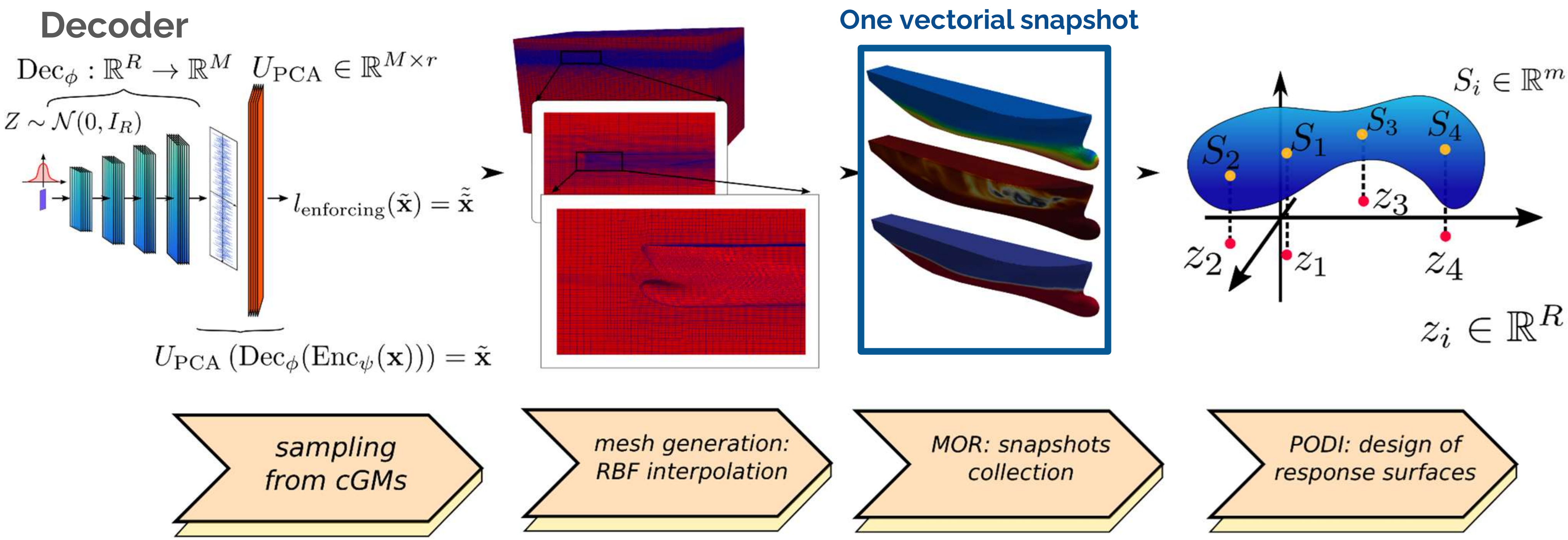
- ➔ **lower parameter dimension**
- ➔ the generation of the new meshes is **significantly faster**.

Constrained Generative Models and Active subspaces can be **combined**.

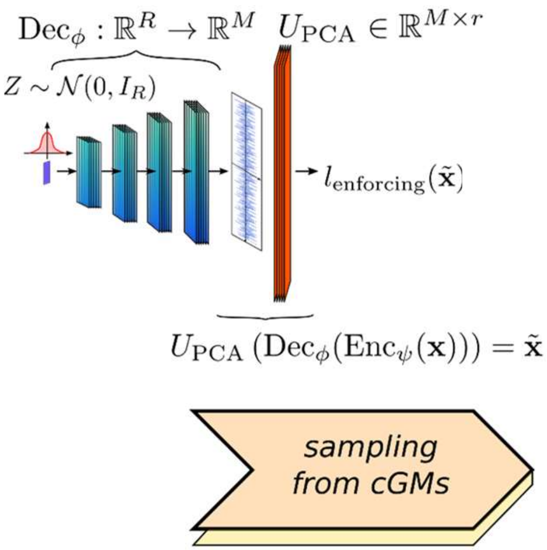
References:

1. Padula G., Romor F., Stabile G., Rozza G. (2024). Generative models for the deformation of industrial shapes with linear geometric constraints: Model order and parameter space reductions. Computer Methods in Applied Mechanics and Engineering.

ROM Building Workflow



Workflow: Constrained Generative Models



Autoencoder

Variational Autoencoder

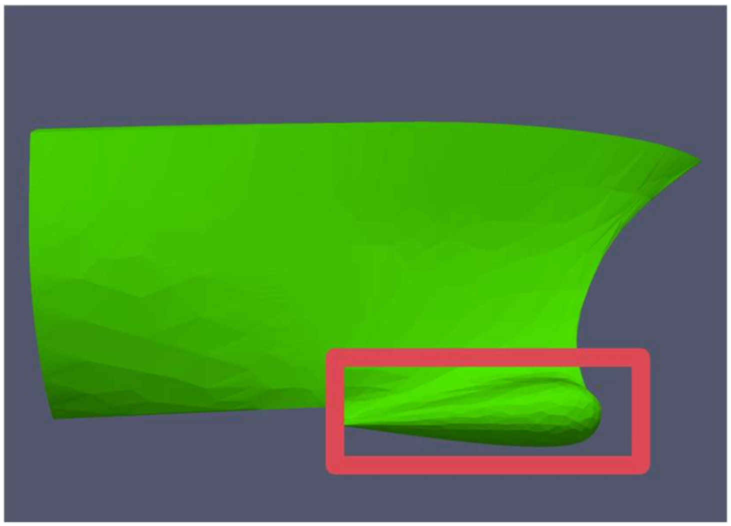
Adversarial Autoencoder

Boundary Equilibrium Generative Adversarial Network

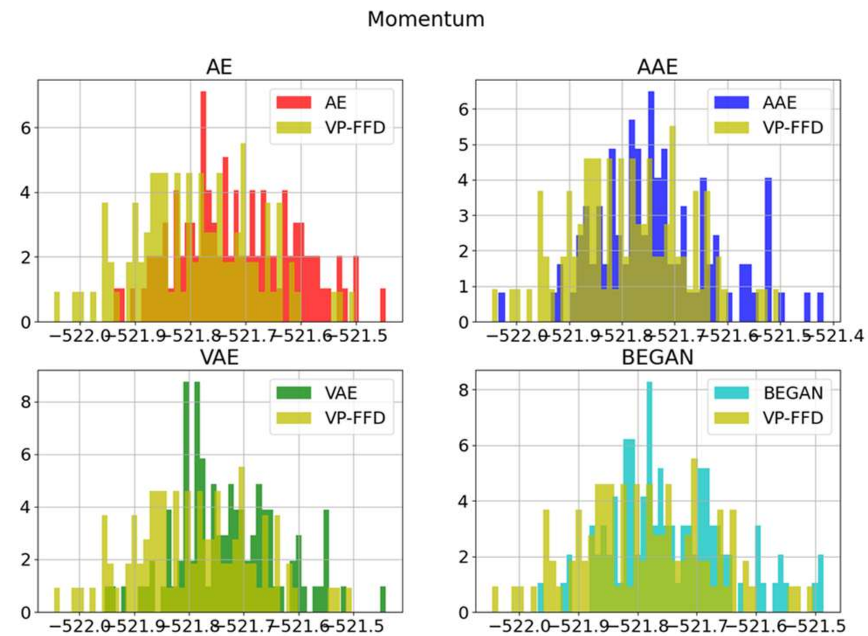
References:

1. Daly, G., Fieldsend E., Tabor G. (2022) Variation Autoencoder without the Variation. arXiv.
2. Kingma D.P., Welling M. (2014). Auto-Encoding Variational Bayes. ICLR 2014.
3. Makhzani A., Shlens G., Jaitly N., Goodfellow I., Frey B. (2016) Adversarial Autoencoders.. ICLR 2016.
4. Berthelot D., Schumm T., Metz L. (2022). BEGAN: Boundary Equilibrium Generative Adversarial Networks. arXiv.

Test Case: DTCHull

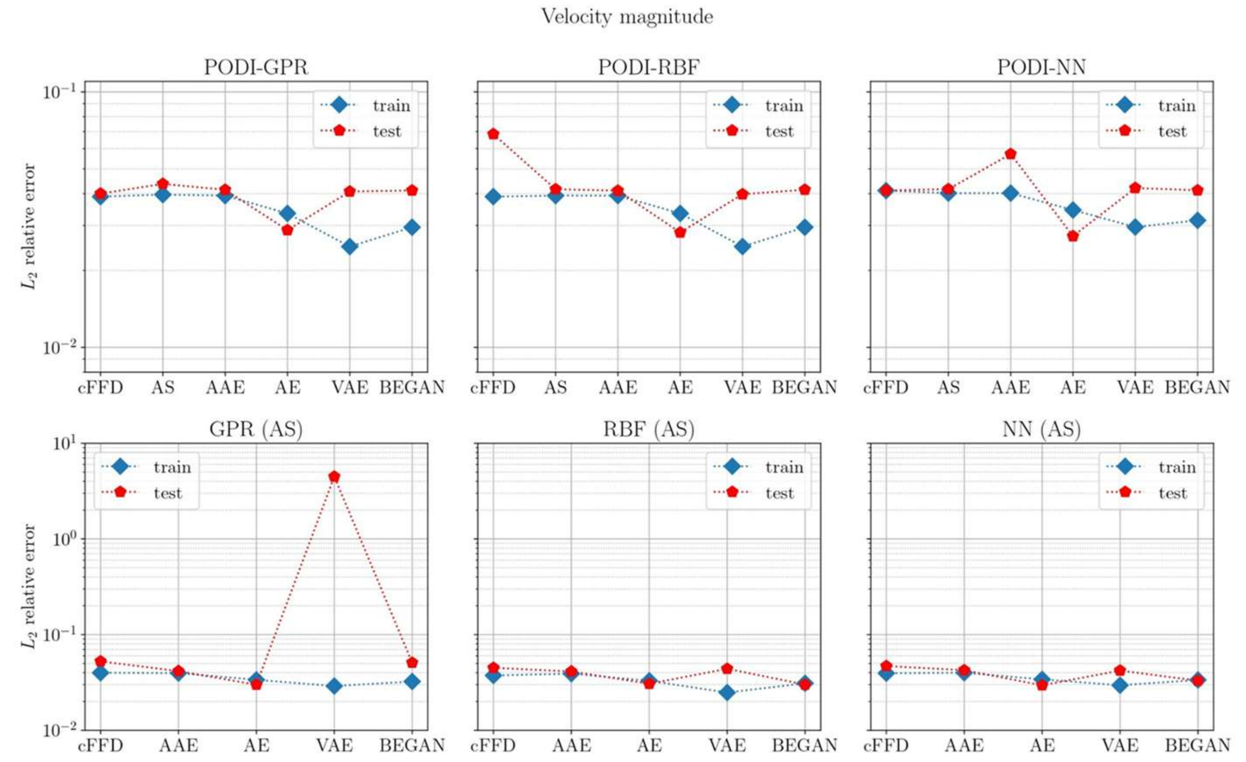


Tested on InterFoam



- **Quantity preserved:** Volume
- **Speedup over Constrained FFD:** 360x
- **Speedup model order reduction:** 432000x
- **Reduced parameter space:** 64 (CFFD) vs 10 (CGM) vs 1 (CGM+AS)

Applying cGM and ROM to Industrial Naval Hull



Generative Models for parameter reduction **decrease the reconstruction error** for all ROMs

Summary of the methods

Inference = Approximation + Parameter Reduction

- Neural networks
- Radial Basis functions
- Gaussian Process Regression
- PODI-GPR
- PODI-NN
- PODI-RBF

- Variational Autoencoders
- Boundary Equilibrium Generative Adversarial Networks
- Adversarial Autoencoder
- Autoencoders