Reduced Order Modeling with OpenFOAM using intrusive and non-intrusive methods

Giovanni Stabile

mathLab, Mathematics Area, SISSA International School for Advanced Studies, Trieste, Italy

Presentation at 3rd German OpenFoam User meetiNg
GOFUN - 2019
Braunschweig
27/28/2019

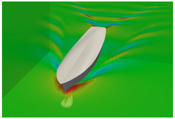
A team developing **Advanced Reduced Order Methods** with special focus on **Computational Fluid Dynamics**

We are at the International School for Advanced Studies (**SISSA**) which is located in **Trieste** (Italy).

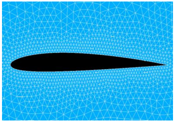
Overview of the physical problems

The interest is in **viscous parametrized incompressible flows**

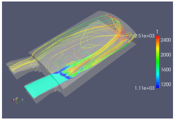
Industrial Flows



Naval Eng.

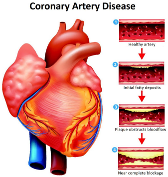


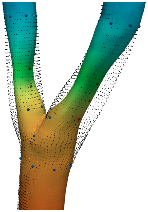
Aeronautics



Industrial App.

Biomedical Applications





Possible applications can be found in **naval** and **nautical** engineering, **aeronautical** engineering and **industrial** engineering.

In general any application dealing with incompressible fluid dynamic problems that has the response depending on **parameter changes** (Reynolds Number, Grashof Number, Geometrical parameters ..)

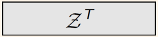
#CFD

Intrusive Model Order Reduction

Intrusive Reduced Order Methods in a nutshell

- $()^{\mathcal{N}}$: "truth" high order method (FEM, FV, FD, SEM) – to be accelerated
- $()_N$: reduced order method (ROM) – *the accelerator*

- **Offline**: very expensive preprocessing (high order): basis calculation (done *once*) after suitable parameters sampling (greedy, POD, ...)

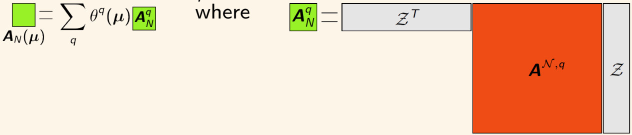


- **Online**: extremely fast (reduced order): real-time input-output evaluation

$$\boldsymbol{\mu} \quad \rightarrow \quad s_N(\boldsymbol{\mu})$$

thanks to an efficient assembly of problem operators

$$\boldsymbol{A}_N(\boldsymbol{\mu}) = \sum_q \theta^q(\boldsymbol{\mu}) \boldsymbol{A}_N^q, \text{ where } \boldsymbol{A}_N^q = \mathcal{Z}^T \boldsymbol{A}^{\mathcal{N},q} \mathcal{Z}$$



- Numerical issues: stability, error bounds, efficient parametrization, sampling, ...

J. S. Hesthaven, G. Rozza, B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer, 2015

The Full Order Model

Governing Equations - The incompressible Navier Stokes Equations

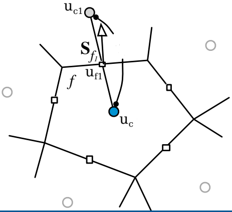
The considered system of PDEs are the **unsteady parametrized incompressible Navier Stokes Equations.**

$$\begin{cases} \boldsymbol{u_t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot 2\nu \boldsymbol{\nabla}^s \boldsymbol{u} = -\boldsymbol{\nabla} p & \text{in } Q, \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = \boldsymbol{0} & \text{in } Q, \\ \boldsymbol{u}(t,x) = \boldsymbol{f}(\boldsymbol{x}) & \text{on } \Gamma_{In} \times [0,T], \\ \boldsymbol{u}(t,x) = \boldsymbol{0} & \text{on } \Gamma_0 \times [0,T], \\ (\nu(\mu)\nabla \boldsymbol{u} - p\boldsymbol{I})\boldsymbol{n} = \boldsymbol{0} & \text{on } \Gamma_{Out} \times [0,T], \\ \boldsymbol{u}(0,\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x}) & \text{in } T_0, \end{cases} \tag{1}$$

with $Q = \Omega \times [0,T] \subset \mathbb{R}^d \times \mathbb{R}^+$ with $d = 2,3$ and the boundary is considered to be $\partial\Omega = \partial\Omega_{,in} \cup \partial\Omega_{,0} \cup \partial\Omega_{,out}$

The governing equations are discretised using a **Finite Volume approach**. Each term is integrated over a control volume and transformed into a surface integral making use of the Green's theorem:

$$\int_\Omega \boldsymbol{\nabla} \cdot \boldsymbol{u} dv = \int_{\partial\Omega} \boldsymbol{u} \cdot \boldsymbol{n} ds = \sum_{i=1}^{N_{S_f}} \boldsymbol{u_{f_i}} \cdot \boldsymbol{S_{f_i}} \tag{2}$$



Generation of the POD spaces

There are several techniques to obtain the hierarchical reduced order spaces later used for the Galerkin projection:

- **POD**
- RB with greedy sampling algorithm

The reduced order space V_u and Q_p are constructed using a **SVD** on the snapshots matrices of **velocity** and **pressure**:

$$\mathcal{U}' = [\boldsymbol{u}'(t_1), \boldsymbol{u}'(t_2), ..., \boldsymbol{u}'(t_n)] \text{ with } \boldsymbol{u}'(t) = \boldsymbol{u}(t) - \overline{\boldsymbol{u}} \tag{3}$$

$$\mathcal{P} = [\boldsymbol{p}(t_1), \boldsymbol{p}(t_2), ..., \boldsymbol{p}(t_n)] \tag{4}$$

$$\mathcal{U}' = \mathcal{W}^u \Sigma^u \mathcal{V}^{uT}, \quad \mathcal{W}^u = [\boldsymbol{\varphi_1}, \boldsymbol{\varphi_2}, ..., \boldsymbol{\varphi_n}], \quad \Sigma^u_{ii} = \lambda^u_i \tag{5}$$

$$\mathcal{P} = \mathcal{W}^p \Sigma^p \mathcal{V}^{pT}, \quad \mathcal{W}^p = [\boldsymbol{\chi_1}, \boldsymbol{\chi_2}, ..., \boldsymbol{\chi_n}], \quad \Sigma^p_{ii} = \lambda^p_i \tag{6}$$

We can **truncate** the dimension of the reduced basis space looking at the eigenvalues and we can finally construct the reduced basis spaces for the **Galerkin projection**:

$$\mathbb{V}_{N_u} = \text{span}(\boldsymbol{\varphi_1}, \boldsymbol{\varphi_2}, ..., \boldsymbol{\varphi_{N_u}}) \qquad \mathbb{Q}_{N_p} = \text{span}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2}, ..., \boldsymbol{\chi_{N_p}})$$

Galerkin Projection

After the reduced basis are set one can perform a Galerkin projection onto the RB spaces:

$$\begin{cases} (\boldsymbol{u_t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot 2\nu \boldsymbol{\nabla}^s \boldsymbol{u} + \boldsymbol{\nabla} p, \boldsymbol{\varphi})_{L^2(\Omega)} = \mathbf{0} \qquad \forall \boldsymbol{\varphi} \in \mathbb{V}_{N_u} \\ (\boldsymbol{\nabla} \cdot \boldsymbol{u}, \chi)_{L^2(\Omega)} = \mathbf{0} \qquad \forall \boldsymbol{\chi} \in \mathbb{Q}_{N_p} \end{cases} \tag{7}$$

and the pressure and velocity fields are approximated using the the POD modes for velocity and pressure respectively.

$$\boldsymbol{u}^{\boldsymbol{r}} \approx \sum_{i=1}^{N_u^r} a_i(t,\mu)\boldsymbol{\varphi_i}(\boldsymbol{x}), \qquad p^r \approx \sum_{i=1}^{N_p^r} b_i(t,\mu)\chi_i(\boldsymbol{x}). \tag{8}$$

The system can be recast in matrix form with the reduced matrices.

$$\begin{aligned} \boldsymbol{M_r}\dot{\boldsymbol{a}} - \nu \boldsymbol{A_r}\boldsymbol{a} + \boldsymbol{C_r}(\boldsymbol{a})\boldsymbol{a} + \boldsymbol{B_r}\boldsymbol{b} = \mathbf{0} \\ \boldsymbol{P_r}\boldsymbol{a} = 0, \end{aligned} \tag{9}$$

where the terms inside equation (9) are evaluated with:

$$\begin{aligned} M_{r_{ij}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle_{L_2(\Omega)} \ , \ A_{r_{ij}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\nabla} \cdot 2\boldsymbol{\nabla}^s \boldsymbol{\varphi_j} \rangle_{L_2(\Omega)} \ , \\ B_{r_{ij}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\nabla} \chi_j \rangle_{L_2(\Omega)} \ , \ P_{r_{ij}} = \langle \chi_i, \boldsymbol{\nabla} \cdot \boldsymbol{\varphi_j} \rangle_{L_2(\Omega)}. \end{aligned} \tag{10}$$

Instabilities in ROMs

It is well known that projection-based ROMs suffer from stability issues.

- Pressure Instabilities
  - Supremizer Stabilization (FV and FEM)
  - Poisson Equation for pressure (FV)

- Advection Dominated and Turbulent problems
  - Eddy viscosity model

- Long term integration instabilities

Sta-Ro (2018). Finite volume POD-Galerkin stabilized reduced order methods for the parametrised incompressible Navier-Stokes equations. *Computers & Fluids*, 173, 273-284.

Instabilities in ROMs

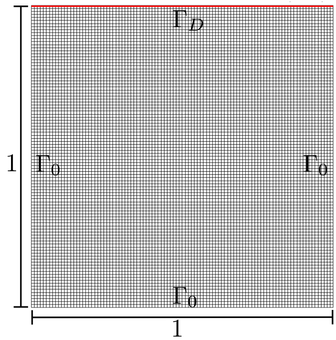
It is well known that projection-based ROMs suffer from stability issues.

- Pressure Instabilities
  - Supremizer Stabilization (FV and FEM)
  - Poisson Equation for pressure (FV)
- Advection Dominated and Turbulent problems
  - Eddy viscosity model
- Long term integration instabilities

Sta-Ro (2018). Finite volume POD-Galerkin stabilized reduced order methods for the parametrised incompressible Navier-Stokes equations. *Computers & Fluids*, 173, 273-284.

The lid driven cavity problem

The first proposed benchmark consists into the well known lid driven cavity problem:

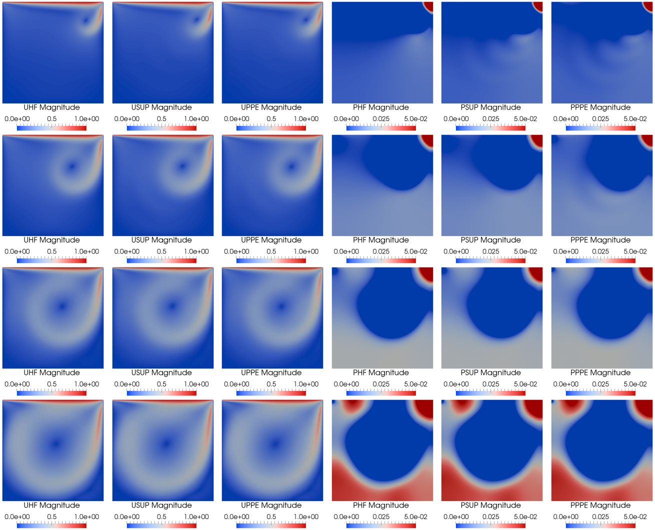


	Γ_D	Γ_0
$\boldsymbol{u}$	$\boldsymbol{u} = (1,0)$	$\boldsymbol{u} = (0,0)$
p	$\nabla p \cdot \boldsymbol{n} = 0$	$\nabla p \cdot \boldsymbol{n} = 0$

The mesh is structured and counts 40000 quadrilateral cells, 200 on each dimension of the square. The kinematic viscosity is equal to $\nu = 1 \times 10^{-4} \text{m}^2/\text{s}$ that leads to a Reynolds number of 10000. **In this case no parametrisation is introduced.**

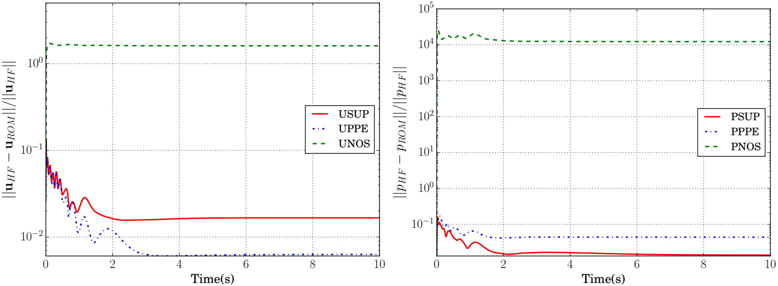
Numerical examples

Comparison of the velocity and pressure fields for high fidelity, SUP-ROM and PPE-ROM. The fields are depicted for different time instant equal to $t = 0.2\text{s}, 0.5\text{s}, 1\text{s}$ and 5s.



Numerical examples

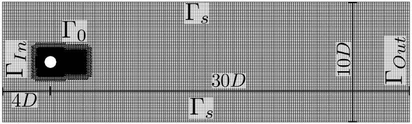
The L^2 norm of the relative error over time for three different models.

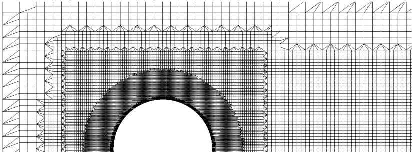


The table contains the cumulative eigenvalues for the lid driven cavity test. The last column contains the value of the inf-sup constant, in the supremizer stabilization case, for different different number of supremizer modes and with a fixed number of velocity and pressure modes.

N Modes	$\boldsymbol{u}$	p	$\boldsymbol{s}$	β
1	0.978946	0.975406	0.980260	9.264e-05
2	0.994184	0.991528	0.995232	9.264e-05
3	0.997737	0.995385	0.997912	7.175e-04
4	0.998990	0.998116	0.999400	7.175e-04
5	0.999483	0.999270	0.999844	7.175e-04
10	0.999971	0.999971	0.999997	1.551e-02

The flow around a circular cylinder



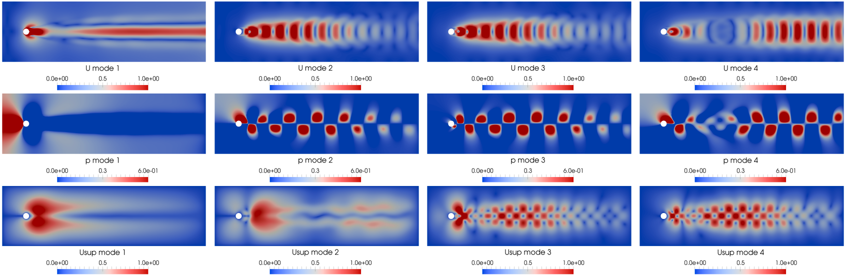


	Γ_{In}	Γ_0	Γ_s	Γ_{Out}
$\boldsymbol{u}$	$\boldsymbol{u} = (1,0)$	$\boldsymbol{u} = (0,0)$	$\boldsymbol{u} \cdot \boldsymbol{n} = 0$	$\nabla \boldsymbol{u} \cdot \boldsymbol{n} = 0$
p	$\nabla p \cdot \boldsymbol{n} = 0$	$\nabla p \cdot \boldsymbol{n} = 0$	$\nabla p \cdot \boldsymbol{n} = 0$	$p = 0$

The properties of the presented algorithms have been tested also with the benchmark of the **laminar flow around a circular cylinder**. In this case the viscosity have been parametrized and results refer to a parameter non experimented in the full order simulations. The parameter space is given by **5 different** values of the viscosity: $\nu \in [0.005, 0.01]$. These values of viscosity result into the values of the Reynolds number $\text{Re} \in [100, 200]$.

Numerical examples

First four modes for velocity pressure and supremizers

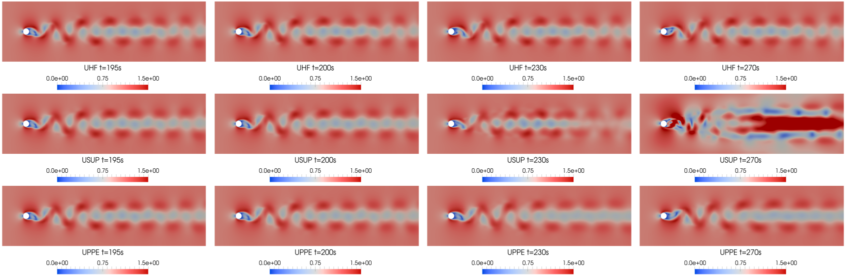


Cumulative eigenvalues

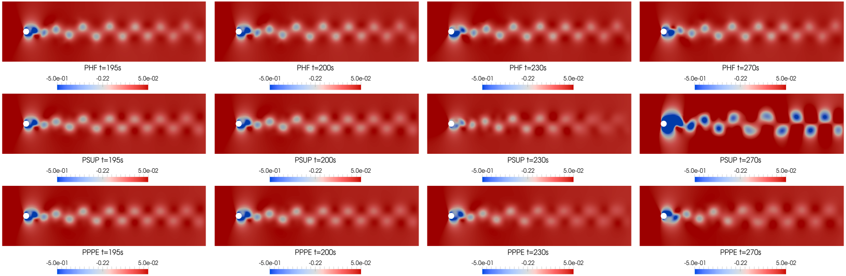
N Modes	$\boldsymbol{u}$	p	$\boldsymbol{s}$	β
1	0.390813	0.793239	0.921046	2.608e-04
2	0.598176	0.85809	0.941746	4.492e-04
3	0.802176	0.911636	0.961438	7.869e-03
4	0.879096	0.934997	0.978072	1.662e-02
5	0.949519	0.955578	0.98669	1.662e-02
10	0.986025	0.992347	0.998307	1.098e-01
15	0.995922	0.997994	0.999732	1.199e-01

Numerical examples

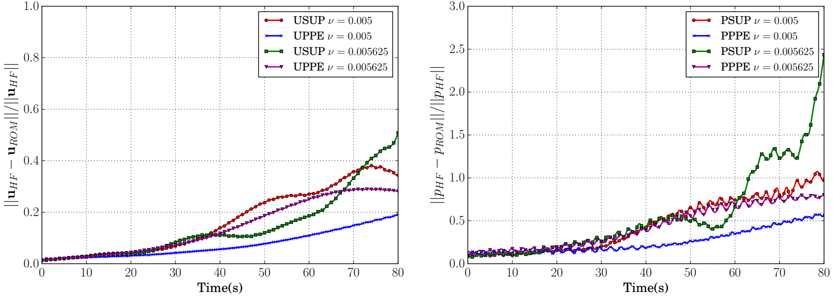
Comparison of the velocity field



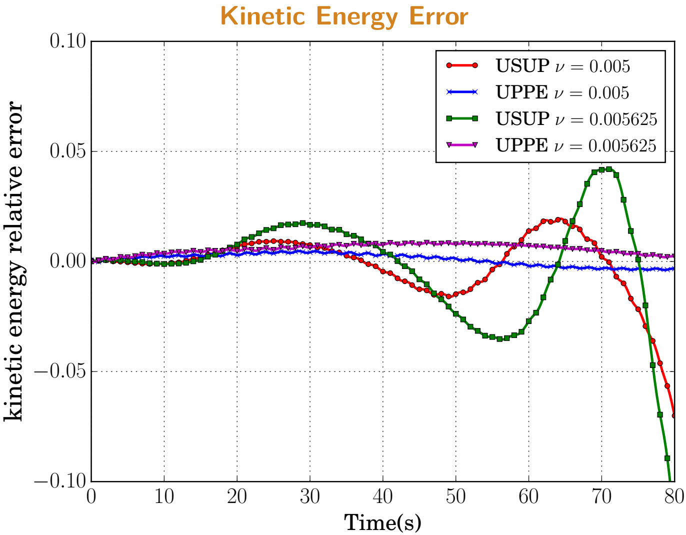
Comparison of the pressure field



Comparison on a longer time window



- Test to check the accuracy of the methods on a longer time span.
- Also different values of the parameters have been checked.
- For both pressure and velocity, on a longer time window, the **Poisson equation approach** gives better results.


Kinetic Energy Error
kinetic energy relative error
Time(s)
USUP $\nu = 0.005$
UPPE $\nu = 0.005$
USUP $\nu = 0.005625$
UPPE $\nu = 0.005625$
0.10
0.05
0.00
−0.05
−0.10
0
10
20
30
40
50
60
70
80

Computational costs

	HF	SUP-ROM	PPE-ROM
Cavity Exp.	25min	7.64s	4.86s
Cylinder Exp.	18.5min $\times$ 6proc.	3.14s	0.971s

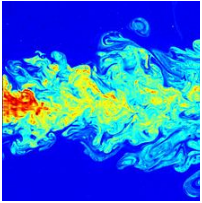
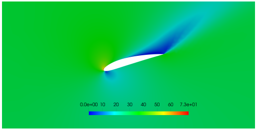
- The **cavity** example has run serially with OpenFOAM 6.0.
- The **cylinder** example has run in parallel with OpenFOAM 6.0.
- In the worst case the speed up is equal to approx. 200.

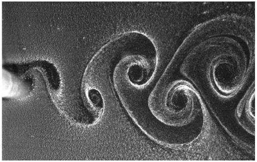
#CFD

ROM and Finite Volume Discretization for fluid mechanics of turbulent flows Joint Work with S. Hijazi

Reduced order methods for turbulent flows

- The goal is to develop reduced order methods dedicated for the treatment of **turbulent flows**.
- We developed a reduced order model which merges **projection-based** methods and **data-driven** techniques.
- The model has been tested on benchmark cases like the **Pitz-Daily** case, the flow around a **circular cylinder** and the flow around a NACA **airfoil** with parametrized **angle of attack**.
- The **Reynolds** number in these cases is up to $\mathbf{Re} = 10^4\text{-}10^6$.
- Challenges include: strong non-linearities in the full order model, long time integration and capturing some complex physical phenomenon at the reduced order level.



Numerical results : Flow around a cylinder, unsteady case

- Results for the mixed **Data-Driven** and projection-based Reduced Order Model (DD-ROM) proved accuracy and efficiency compared to the ones obtained from a fully projection-based strategy.

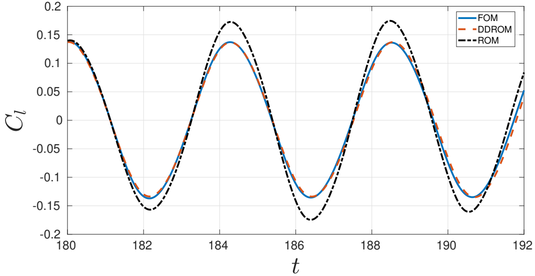


Figure: FOM,ROM and DD-ROM lift coefficients for the forces acting on the cylinder, in this case $Re = 10^4$.

- DD-ROM relative error is in the range of $1-5$ %, while ROM has a relative error of 20%.
- $T_{CPU_{FOM}} = 525.32\ s$, $T_{CPU_{DD-ROM}} = 1.095\ s$
- Speed up of 479.

Hi-Sta-Mo-Ro (2018) Data-Driven POD–Galerkin reduced order model for turbulent flows POD–Galerkin reduced order model for turbulent flows, *In Preparation*

The ITHACA-FV computational library

- **ITHACA-FV** (In real Time Highly Advanced Computational Applications for Finite Volumes) is a C++ implementation based on **OpenFOAM** of several reduced order modeling techniques.
- It is mainly developed and maintained at **SISSA mathLab** but counts already several developers and users around the world.
- It is Open-Source and publicly available on **GitHub** (https://github.com/mathLab/ITHACA-FV).
- It has been successfully used to perform **intrusive** and **non-intrusive** model order reduction for stationary and unstationary fluid dynamic problems, heat transfer problems, coupled heat transfer and fluid dynamics problems.
- Dense Linear Algebra is based on the **Eigen** C++ library.

#CFD

ROM using non Intrusive Techniques and OpenFOAM
Joint Work with N. Demo and M. Tezzele

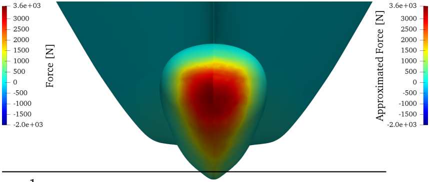
Proper orthogonal decomposition with interpolation

The **proper orthogonal decomposition with interpolation** is a method to approximate the numerical solution of a parametric partial differential equations as combination of few solutions computed for some properly chosen parameters.

$$\forall \boldsymbol{\mu}_k \in \mathcal{P}_{train} \boldsymbol{u}(\mu_k) \approx \boldsymbol{u}^N(\boldsymbol{\mu}_k) = \sum_{i=1}^{N} a_i(\boldsymbol{\mu}_k)\phi_i$$

$$\boldsymbol{u}^N_{NEW} = \sum_{i=1}^{N} a_i(\boldsymbol{\mu}_{NEW})$$

It relies only on the **snapshots**: it does not require any information about the system (**non-intrusive** approach).
This algorithm has been implemented by SISSA mathLab in an open source package called **EZyRB**[1], written in Python.

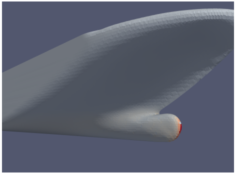


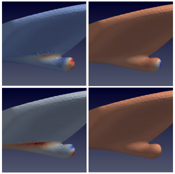
[1]EZyRB is available at: `https://github.com/mathLab/EZyRB`

Dynamic mode decomposition

The **dynamic mode decomposition** is an algorithm describing a nonlinear time-dependent problem as combination of few main structures that evolve linearly in time.

- it provides a linear approximation of the operator $\boldsymbol{A}$ defined as $x_{k+1} = \boldsymbol{A} x_k$, where the x_{k+1} and x_k are the system output at two sequential instants;
- it is an **equation free** algorithm: it operates just on the data and it does not require assumption about the original system;
- the algorithm has been implemented by SISSA mathLab in the package **PyDMD**[2], completely open source.





[2] PyDMD is available at: `https://github.com/mathLab/PyDMD`.

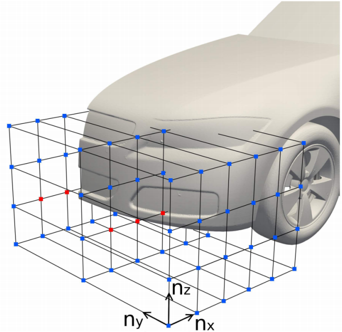
#CFD

Automotive Application, Shape optimization of a bumper, Work by F. Sailmoraghi[3](Former SISSA mathLab).

[3] In cooperation with OPTIMAD (A. Scardigli, H. Telib)

Shape Optimization of a bumper

- The objective is to perform **shape optimization** of the bumper of car in order to reduce the **drag coefficient** of the car.
- All the possible geometries are determined using a proper parameter sampling and the **free form deformation algorithms**.



General purpose shape morphing methods

- **PyGeM** is a python library using Free Form Deformation, Inverse Distance Weighting, and Radial Basis Function interpolation technique to parametrise and morph complex geometries.
- The main focus of PyGeM is to interact with the major industrial file formats used for **CAD** management. Since it has to integrate itself in the industrial workflow we have chosen python. It easily handles also **OpenFOAM meshes**.

PyGeM

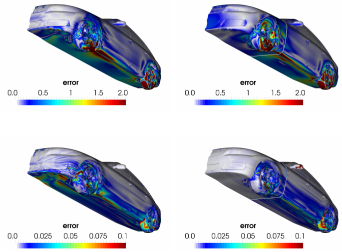
PyGeM is available at: `https://github.com/mathLab/PyGeM`

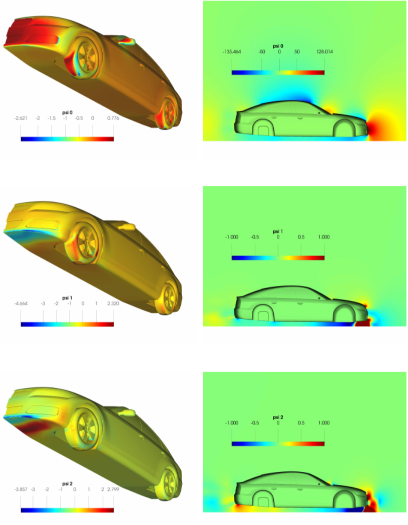
Free Form Deformation, mesh morphing and reduced order methods: enablers for efficient aerodynamic shape optimization, F. Sailmoraghi, A. Scardigli, H. Telib, G. Rozza, International Journal of Computational Fluid Dynamics, 4-5(32), 2018

Shape Optimization of a bumper

The Full order model is then run on a set of parameter values and **POD basis functions** are constructed for velocity and pressure acting on the surface of the car.

- The POD-I **algorithm** is constructed to approximate the solution on a much larger set of parameter values and some configurations that are minimizing the drag are selected.
- The **error** between the ROM and FOM can be computed easily running a full order model simulation only on the most promising configurations.





#CFD

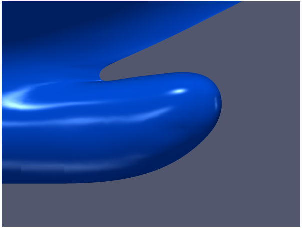
Naval Engineering Application, shape optimization of a bulbous bow
Joint Work with N. Demo and M. Tezzele

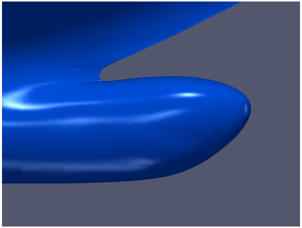
The shape optimization problem

Keywords: **#ShapeOptimization #ModelReduction #HullResistance**

A standard shape optimization system needs a **geometrical modeler** for shape morphing, an **high-fidelity solver** for evaluate the objective function and an **optimization algorithm**. The optimization cycle can last even **months**.

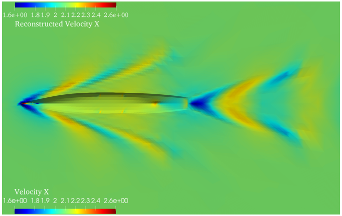
We introduce in the pipeline two different **model reduction** techniques in order to improve the performances.

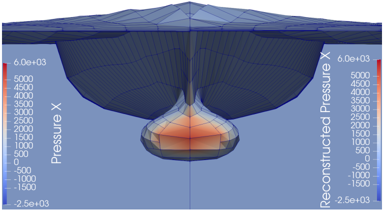


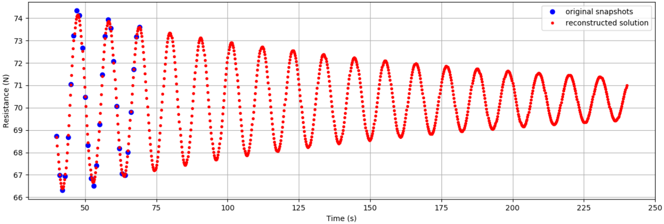


Dynamic mode decomposition

We exploit the capabilities of the DMD, reducing the computational cost of a numerical simulation: we collect few **snapshots** from the full-order system and we use them to approximate the system state at regime.





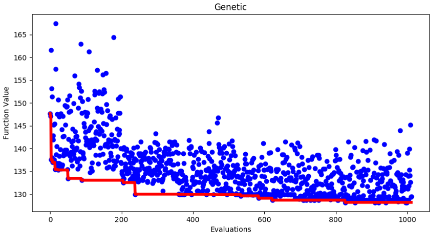


Optimization algorithm

Due to the nonlinearity of the problem, a gradient-based optimization does not assure a global optimization: we need some expensive global optimization algorithm, as the **genetic algorithm**.

The genetic algorithm is composed by following steps:

1. a population of individuals is randomly created; in our work, these individuals are the deformed shapes corresponding to a specific parameter;
2. for all the deformed hulls, a new mesh is created and the numerical solution is computed;
3. the total resistance is evaluated for all the individuals and the most-fit ones are crossed and randomly mutated in order to create a new population;
4. the procedure iterates until all the individuals converge to the optimal point.



Conclusion

What we have done...

- We developed a reduced order modeling pipeline based on OpenFOAM exploiting both intrusive and non-intrusive methods.
- All the developed methods are available as open source packages.
- We developed a shape optimization pipeline using FFE, DMD and the PODI techniques to improve the performances.
- The non-intrusive pipeline has been implemented in OpenFOAM but it is completely independent from the full-order solver used (you can plug what you prefer) and the parametrization tools.

... and what else?

- Reduction of the parameter space using advanced techniques such as active subspaces.
- Intrusive and non-intrusive methods for compressible flows.
- Better exploitation of machine learning tools for non-intrusive methods.
- We are organizing a Summer School on reduced order methods in Computational Fluid Dynamics from the 8th to 12th of July.

REGIONE AUTONOMA FRIULI VENEZIA GIULIA
UNIONE EUROPEA
Fondo Europeo di Sviluppo Regionale
FINCANTIERI
FSE FRIULI VENEZIA GIULIA POR 2014 2020
erc
prelica