Finite volume POD-Galerkin stabilized reduced order methods for the parametrized incompressible Navier- Stokes equations

G. Stabile, G. Rozza

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

MoRePaS 2018, Model Reduction of Parametrized Systems
10-13/04/2018 Nantes (France)

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

Outline

1. Introduction

- Overview of the physical problems
- Why Finite Volumes?

2. The Full Order Model

- The Finite Volume numerical discretisation

3. The Reduced Order Model

- POD-Galerkin ROM for finite volume
- Issues in pressure reconstruction for incompressible flows
- Presentation of different methods for pressure reconstruction

4. Numerical Examples

- Lid driven cavity
- Flow around a circular cylinder

5. Conclusions and Outlooks

6. Other activities and research interests

Introduction

Overview of the physical problems

The interest is in **viscous steady and unsteady parametrized incompressible flows**

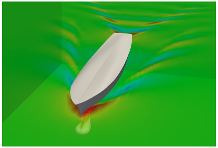


Figure: Naval Eng.

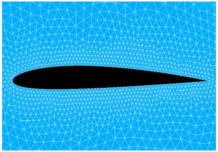


Figure: Aeronautics

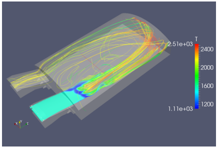


Figure: Industrial App.

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 ..)

Why Finite Volumes?

The finite element method is nowadays the standard in the reduced order modelling community so why to use a different discretisation technique?

- It became the standard for **real world applications** in several engineering fields (Aeronautics, Industrial flows, Automotive, Naval Engineering)
- One can find well developed open source libraries, **OpenFOAM** is today probably the most spread CFD open-source solver.
- For increasing Reynolds numbers there are less problems concerning stability and several **turbulence models** are already available.
- More difficulties into the **affine decomposition** of the differential operators.
- The ROM methodology, mainly developed for **FEM solvers**, needs to be adapted.
- The **geometrical parametrization** includes many **more difficulties** respect to a finite element setting

Issues in FV and Reduced Order Modelling

To export the ROM methodology, mainly developed for finite element solvers, into a **Finite Volume setting** several issues need to be tackled.

- Adapt ROM methods to **finite volume approximations** [Hassdonk and Ohlberger (2008)].
- Implement efficient **POD-Galerkin** strategies [Lorenzi et al (2016)].
- **Geometrical Parametrization** for non-linear problems [Drohmann et al (2009)].
- **Stabilization** issues for **incompressible** flows [Rozza et al., Noack, Akhtar..].
- **Stabilization for compressible flows** and **long time** intervals [Carlberg et al (2017) , Balajewicz et al. (2016))].
- Develop ROMs **beyond** the **laminar assumption**.

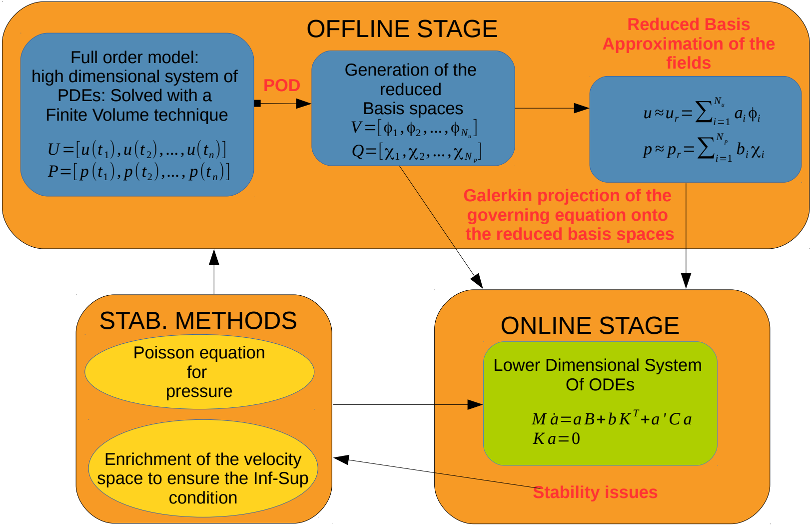
Issues in FV and Reduced Order Modelling

To export the ROM methodology, mainly developed for finite element solvers, into a **Finite Volume setting** several issues need to be tackled.

- Adapt ROM methods to **finite volume approximations** [Hassdonk and Ohlberger (2008)].
- Implement efficient **POD-Galerkin** strategies [Lorenzi et al (2016)].
- **Geometrical Parametrization** for non-linear problems [Drohmann et al (2009)].
- **Stabilization** issues for **incompressible** flows [Rozza et al., Noack, Akhtar..].
- **Stabilization for compressible flows** and **long time** intervals [Carlberg et al (2017) , Balajewicz et al. (2016))].
- Develop ROMs **beyond** the **laminar assumption**.

Reduced Order Modelling

Most of the problems require high dimensional parametrized simulations.



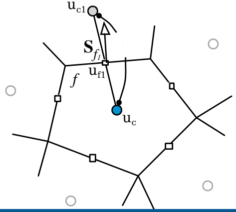
Governing Equations - The incompressible Navier Stokes Equations

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:



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

The Finite Volume Method

In particular the high fidelity simulations have been carried out using the finite volume solver **OpenFOAM®**.

The term $\boldsymbol{u}_{f_i}$ can be determined using a variety of schemes. The convection term, in order to ensure stability needs particular care:

- **The diffusion term** is evaluated using a central differencing scheme.
- **The convection term** is evaluated using an upwind scheme with non-orthogonal correction.
- **The gradient term** is evaluated using a central differencing scheme.
- **The time integration** is performed using a backward Euler method.

The discretisation permits to transform the system of PDEs into a system of algebraic non-linear equations:

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

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**:

$$\mathcal{U}' = [\boldsymbol{u}'(t_1), \boldsymbol{u}'(t_2), ..., \boldsymbol{u}'(t_n)] \text{ with } \boldsymbol{u}'(t) = \boldsymbol{u}(t) - \overline{\boldsymbol{u}} \tag{4}$$

$$\mathcal{P} = [\boldsymbol{p}(t_1), \boldsymbol{p}(t_2), ..., \boldsymbol{p}(t_n)] \tag{5}$$

$$\mathcal{U}' = \mathcal{W}^u \Sigma^u \mathcal{V}^{uT}, \quad \mathcal{W}^p = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, ..., \boldsymbol{\varphi}_n], \quad \Sigma^u_{ii} = \lambda^u_i \tag{6}$$

$$\mathcal{P} = \mathcal{W}^p \Sigma^p \mathcal{V}^{pT}, \quad \mathcal{W}^p = [\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, ..., \boldsymbol{\chi}_n], \quad \Sigma^p_{ii} = \lambda^p_i \tag{7}$$

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:

$$\begin{cases} (\boldsymbol{u_t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot 2\nu \boldsymbol{\nabla}^s \boldsymbol{u} + \boldsymbol{\nabla} p, \boldsymbol{\varphi})_{L^2(\Omega)} = \boldsymbol{0} \qquad \forall \boldsymbol{\varphi} \in \mathbb{V}_{N_u} \\ (\boldsymbol{\nabla} \cdot \boldsymbol{u}, \chi)_{L^2(\Omega)} = \boldsymbol{0} \qquad \forall \chi \in \mathbb{Q}_{N_p} \end{cases} \tag{8}$$

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

$$\boldsymbol{u}^r \approx \sum_{i=1}^{N_u^r} a_i(t,\mu) \boldsymbol{\varphi_i}(\boldsymbol{x}), \qquad p^r \approx \sum_{i=1}^{N_p^r} b_i(t,\mu) \chi_i(\boldsymbol{x}). \tag{9}$$

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

$$\begin{aligned} \boldsymbol{M_r} \dot{\boldsymbol{a}} - \nu \boldsymbol{A_r} \boldsymbol{a} + \boldsymbol{C_r}(\boldsymbol{a}) \boldsymbol{a} + \boldsymbol{B_r} \boldsymbol{b} = \boldsymbol{0} \\ \boldsymbol{P_r} \boldsymbol{a} = 0, \end{aligned} \tag{10}$$

where the terms inside equation (10) are evaluated with:

$$\begin{aligned} M_{r_{ij}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle_{L_2(\Omega)} \ , \ A_{r_{ij}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\nabla} \cdot 2 \boldsymbol{\nabla}^s \boldsymbol{\varphi_j} \rangle_{L_2(\Omega)} \ , \\ B_{r_{ij}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\nabla} \chi_j \rangle_{L_2(\Omega)} \ , \ P_{r_{ij}} = \langle \chi_i, \boldsymbol{\nabla} \cdot \boldsymbol{\varphi_j} \rangle_{L_2(\Omega)}. \end{aligned} \tag{11}$$

The Reduced Order Model

In order to ensure an efficient online/offline decoupling the reduced matrices must be precomputed during the offline stage

$$\begin{aligned} \boldsymbol{M_r}\dot{\boldsymbol{a}} - \nu \boldsymbol{A_r}\boldsymbol{a} + \boldsymbol{C_r}(\boldsymbol{a})\boldsymbol{a} + \boldsymbol{B_r}\boldsymbol{b} = \boldsymbol{0} \\ \boldsymbol{P_r}\boldsymbol{a} = 0, \end{aligned} \tag{12}$$

with this regard the **non-linear convective term** $\boldsymbol{C_r}(\boldsymbol{a})$ needs particular attention. The idea here is to use a third order tensor $\mathbf{C}_r$

$$\mathrm{C}_{r_{ijk}} = \langle \boldsymbol{\varphi_i}, \boldsymbol{\nabla} \cdot (\boldsymbol{\varphi_j} \otimes \boldsymbol{\varphi_k}) \rangle_{L_2(\Omega)}. \tag{13}$$

and at each fixed point iteration of the solution procedure, each entry of the contribution to the reduced residual given by the convective term $\boldsymbol{\mathcal{R}}_c^r = \boldsymbol{C_r}(\boldsymbol{a})\boldsymbol{a}$, can be computed with:

$$\boldsymbol{\mathcal{R}}_{c_i}^r = (\boldsymbol{C_r}(\boldsymbol{a})\boldsymbol{a})_i = \boldsymbol{a}^T \mathbf{C}_{r_{i\bullet\bullet}} \boldsymbol{a}. \tag{14}$$

Also other approaches are possible such as empirical interpolation, gabby POD,...

The resulting system of non-linear ODEs

The Galerkin projection gives rise to a **non-linear system of ODE's**

$$\begin{cases} \dot{\boldsymbol{a}} = \nu \boldsymbol{A_r} \boldsymbol{a} - \boldsymbol{a}^T \mathbf{C_r} \boldsymbol{a} - \boldsymbol{B_r} \boldsymbol{b} \\ \boldsymbol{P_r} \boldsymbol{a} = 0 \end{cases} \tag{15}$$

In the Galerkin projection there is also the **gradient of pressure**:

$$(\varphi, \nabla p)_{L^2(\Omega)} = - \int_\Omega p(\nabla \cdot \varphi) \mathrm{dv} + \int_{\partial\Omega} p(\varphi \cdot \boldsymbol{n}) \mathrm{ds} \tag{16}$$

The resulting system of non-linear ODEs

The Galerkin projection gives rise to a **non-linear system of ODE's**

$$\begin{cases} \dot{\boldsymbol{a}} = \nu \boldsymbol{A_r a} - \boldsymbol{a}^T \mathbf{C}_r \boldsymbol{a} - \cancel{\boldsymbol{B_r b}} \\ \cancel{\boldsymbol{P_r a} = 0} \end{cases} \tag{15}$$

In the Galerkin projection there is also the **gradient of pressure**:

$$\cancel{(\varphi, \nabla p)_{L^2(\Omega)} = -\int_\Omega p(\nabla \cdot \varphi)\mathrm{dv} + \int_{\partial\Omega} p(\varphi \cdot \boldsymbol{n})\mathrm{ds}} \tag{16}$$

This term is on most of the cases **numerically zero** so it is neglected and **only** the **momentum equation**, without the pressure term is solved.

In many applications the **pressure** is the field in which we are interested (Fluid-Structure interaction problems, Evaluation of drag and lift forces)

Pressure reconstruction and stability

Using the standard Navier-Stokes equation the **online Inf-Sup** condition is not anymore satisfied.

Pressure reconstruction and inf-sup condition

- Reconstruction using a Poisson equation for pressure [Akhtar et al. TCFD (2009), Noack et al. JFM (2005)]
- Online Inf-Sup approximation [Rozza and Veroy, CMAME (2007)]Rozza et al, Numerische Mathematik(2013)][Ballarin et al. IJNME (2015)]
- Residual-based stabilization [Caiazzo, Iliescu et al. JCP(2013)]
- Petrov-Galerkin projection [Carlberg et al., IJNME (2011)][Dahmen et al., ESAIM: M2AN (2014)][Abdulle and Budáč, CRAS (2015)]

Pressure reconstruction and stability

Using the standard Navier-Stokes equation the **online Inf-Sup** condition is not anymore satisfied.

Pressure reconstruction and inf-sup condition

- Reconstruction using a Poisson equation for pressure [Akhtar et al. TCFD (2009), Noack et al. JFM (2005)]
- Online Inf-Sup approximation [Rozza and Veroy, CMAME (2007)]Rozza et al, Numerische Mathematik(2013)][Ballarin et al. IJNME (2015)]
- Residual-based stabilization [Caiazzo, Iliescu et al. JCP(2013)]
- Petrov-Galerkin projection [Carlberg et al., IJNME (2011)][Dahmen et al., ESAIM: M2AN (2014)][Abdulle and Budáč, CRAS (2015)]

The Poisson Equation for Pressure

- One possible way to reconstruct the pressure is to exploit a **Poisson equation for pressure** obtained taking the divergence of the Momentum equation and exploiting the **divergence-free** constraint.

$$\begin{cases} \boldsymbol{u_t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot 2\nu \boldsymbol{\nabla}^{\boldsymbol{s}} \boldsymbol{u} = -\boldsymbol{\nabla} p & \text{in } Q \\ \Delta p = -\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u})) & \text{in } Q, \\ \boldsymbol{u}(t, \boldsymbol{x}) = \boldsymbol{0} & \text{on } \Gamma_0 \times [0, T], \\ \boldsymbol{u}(t, \boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) & \text{on } \Gamma_{In}, \\ \frac{\partial p}{\partial \boldsymbol{n}} = -\nu \boldsymbol{n} \cdot (\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \boldsymbol{u}) - \boldsymbol{n} \cdot \boldsymbol{f}_t & \text{on } \Gamma. \end{cases} \tag{17}$$

The resulting equations can be then projected onto the reduced basis spaces of velocity and pressure.

$$\langle \boldsymbol{\varphi_i}, \boldsymbol{u_t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) + \boldsymbol{\nabla} p - \boldsymbol{\nabla} \cdot 2\nu \boldsymbol{\nabla}^{\boldsymbol{s}} \boldsymbol{u} \rangle_{L_2(\Omega)} = 0, \tag{18a}$$

$$\begin{aligned} &\langle \nabla \chi_i, \nabla p \rangle_{L_2(\Omega)} + \langle \nabla \chi_i, \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) \rangle_{L_2(\Omega)} \\ &- \nu \langle \boldsymbol{n} \times \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \times \boldsymbol{u} \rangle_{\Gamma} - \langle \chi_i, \boldsymbol{n} \cdot \boldsymbol{f_t} \rangle_{\Gamma} = 0. \end{aligned} \tag{18b}$$

Where $\boldsymbol{u_r}$ and p_r are the reduced order approximation of velocity and pressure:

$$\boldsymbol{u} \approx \boldsymbol{u_r} = u_D \boldsymbol{\varphi_L} + \sum_{i=1}^{N_u} a_i(t) \varphi_i(\boldsymbol{x}) \tag{19}$$

$$\boldsymbol{p_r} = \sum_{i=1}^{N_p} b_i(t) \chi_i(\boldsymbol{x}) \tag{20}$$

The Poisson Equation for Pressure

The Poisson equation for pressure could be exploited in different ways [Caiazzo, Iliescu et al. JCP(2013)]:

- A posteriori reconstruction of the pressure using the velocities ROM solution
- Projection of the Poisson equation onto the POD pressure space

$$\langle \boldsymbol{\varphi}_i, \boldsymbol{u}_t + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p - \nabla \cdot 2\nu \nabla^s \boldsymbol{u} \rangle_{L_2(\Omega)} = 0, \tag{21a}$$

$$\begin{aligned} &\langle \nabla \chi_i, \nabla p \rangle_{L_2(\Omega)} + \langle \nabla \chi_i, \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) \rangle_{L_2(\Omega)} \\ &- \nu \langle \boldsymbol{n} \times \nabla \chi_i, \nabla \times \boldsymbol{u} \rangle_{\Gamma} - \langle \chi_i, \boldsymbol{n} \cdot \boldsymbol{f_t} \rangle_{\Gamma} = 0. \end{aligned} \tag{21b}$$

$$\boldsymbol{M_r} \dot{\boldsymbol{a}} - \nu \boldsymbol{A_r} \boldsymbol{a} + \boldsymbol{a}^T \mathbf{C_r} \boldsymbol{a} + \boldsymbol{B_r} \boldsymbol{b} = 0, \tag{22a}$$

$$\boldsymbol{D_r} \boldsymbol{b} + \boldsymbol{a}^T \mathbf{G_r} \boldsymbol{a} - \nu \boldsymbol{N_r} \boldsymbol{a} - \boldsymbol{F_r} = 0. \tag{22b}$$

$$\begin{aligned} D_{r_{ij}} &= \langle \nabla \chi_i, \nabla \chi_j \rangle_{L_2(\Omega)} , \\ \mathrm{G}_{r\,ijk} &= \langle \nabla \chi_i, \nabla \cdot (\boldsymbol{\varphi_j} \otimes \boldsymbol{\varphi_k}) \rangle_{L_2(\Omega)}, \end{aligned} \tag{23}$$

$$\begin{aligned} N_{r_{ij}} &= \langle \boldsymbol{n} \times \nabla \chi_i, \nabla \times \boldsymbol{\varphi_j} \rangle_{\Gamma}, \\ F_{r_i} &= \langle \chi_i, \boldsymbol{n} \cdot \boldsymbol{f_t} \rangle_{\Gamma}. \end{aligned} \tag{24}$$

The Poisson Equation for Pressure

The Poisson equation for pressure could be exploited in different ways [Caiazzo, Iliescu et al. JCP(2013)]:

- A posteriori reconstruction of the pressure using the velocities ROM solution
- Projection of the Poisson equation onto the POD pressure space

$$\langle \boldsymbol{\varphi_i}, \boldsymbol{u_t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) + \boldsymbol{\nabla} p - \boldsymbol{\nabla} \cdot 2\nu \boldsymbol{\nabla}^s \boldsymbol{u} \rangle_{L_2(\Omega)} = 0, \tag{21a}$$

$$\begin{aligned} &\langle \nabla \chi_i, \nabla p \rangle_{L_2(\Omega)} + \langle \nabla \chi_i, \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) \rangle_{L_2(\Omega)} \\ &- \nu \langle \boldsymbol{n} \times \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \times \boldsymbol{u} \rangle_\Gamma - \langle \chi_i, \boldsymbol{n} \cdot \boldsymbol{f_t} \rangle_\Gamma = 0. \end{aligned} \tag{21b}$$

$$\boldsymbol{M_r} \dot{\boldsymbol{a}} - \nu \boldsymbol{A_r} \boldsymbol{a} + \boldsymbol{a}^T \mathbf{C_r} \boldsymbol{a} + \boldsymbol{B_r} \boldsymbol{b} = 0, \tag{22a}$$

$$\boldsymbol{D_r} \boldsymbol{b} + \boldsymbol{a}^T \mathbf{G_r} \boldsymbol{a} - \nu \boldsymbol{N_r} \boldsymbol{a} - \boldsymbol{F_r} = 0. \tag{22b}$$

$$\begin{aligned} D_{r_{ij}} &= \langle \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \chi_j \rangle_{L_2(\Omega)} \ , \\ \mathrm{G}_{\mathrm{r}\, ijk} &= \langle \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \cdot (\boldsymbol{\varphi_j} \otimes \boldsymbol{\varphi_k}) \rangle_{L_2(\Omega)}, \end{aligned} \tag{23}$$

$$\begin{aligned} N_{r_{ij}} &= \langle \boldsymbol{n} \times \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \times \boldsymbol{\varphi_j} \rangle_\Gamma, \\ F_{r_i} &= \langle \chi_i, \boldsymbol{n} \cdot \boldsymbol{f_t} \rangle_\Gamma . \end{aligned} \tag{24}$$

The Supremizer Approach

We know that in a Galerkin approach to ensure the solvability and stability of the problem the reduced basis spaces must fulfill the LBB parametrized **inf-sup** condition.

$$\inf_{q \in Q} \sup_{\boldsymbol{v} \in V} \frac{b(q, \boldsymbol{v}; \mu)}{\|q\|_Q \|\boldsymbol{v}\|_V} = \beta(\mu) > 0 \quad (25)$$

$$b(q, \boldsymbol{v}) = \int_\Omega q \nabla \cdot \boldsymbol{v} d\boldsymbol{x} \quad (26)$$

Normally the resulting spaces obtained with a POD or a Reduced Basis approach **do not** fulfil this condition. In order to fulfil this condition at reduced order level a **supremizer problem** is solved.

$$\begin{cases} \boldsymbol{\Delta s} = -\boldsymbol{\nabla} p & \text{in } \Omega \\ \boldsymbol{s} = \boldsymbol{0} & \text{on } \partial\Omega \end{cases} \quad (27)$$

where p is a general pressure mode used to create the reduced basis space.

Supremizer enrichment

The supremizer is the element, that given a certain $q \in Q$ satisfies the Inf-Sup condition.

The Supremizer Approach

It is possible to use different approaches to stabilize the problem: an **exact approach** and an **approximated** one.
In the exact one the supremizer problem is solved for each pressure mode of the pressure POD space:

$$\begin{cases} \boldsymbol{\Delta s} = -\boldsymbol{\nabla}\chi & \text{in } \Omega, \ \forall \boldsymbol{\chi} \in \mathbb{Q}_{N_p} \\ \boldsymbol{s} = \boldsymbol{0} & \text{on } \partial\Omega \end{cases} \tag{28}$$

And the supremizer space is constructed:

$$S_s = [\boldsymbol{s}(\chi_1), \boldsymbol{s}(\chi_2), ..., \boldsymbol{s}(\chi_{N_p})] \tag{29}$$

In this way, if we add a supremizer mode for each pressure mode it is possible to show that the reduced inf-sup condition is automatically met.
The resulting supremizer solutions are added to the velocity space and the enriched velocity space reads:

$$\begin{aligned} L_s &= [\boldsymbol{s}_1, \dots, \boldsymbol{s}_{\boldsymbol{N_s^r}}] \in \mathbb{R}^{N_u^h \times N_s^r}, \\ \tilde{V}_u &= [\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{\boldsymbol{N_u^r}}] \oplus [\boldsymbol{s}_1, \dots, \boldsymbol{s}_{\boldsymbol{N_s^r}}] \in \mathbb{R}^{N_u^h \times (N_u^r + N_s^r)}. \end{aligned} \tag{30}$$

The Supremizer Approach

Another way is to use an approximated approach. The supremizer problem is solved for each pressure snapshots to get a snapshots matrix of supremizer solutions:

$$\begin{cases} \boldsymbol{\Delta s} = -\boldsymbol{\nabla} p & \text{in } \Omega, \ \forall \boldsymbol{p} \in \boldsymbol{\mathcal{P}} \\ \boldsymbol{s} = \boldsymbol{0} & \text{on } \partial\Omega \end{cases} \tag{31}$$

A supremizer snapshots matrix is constructed:

$$S_s = [\boldsymbol{s}(p_1), \boldsymbol{s}(\chi_2), ..., \boldsymbol{s}(p_{N_p})] \tag{32}$$

A basis for the supremizer space can be constructed using a **POD approach** and we can truncate the basis retaining only the first N_s energetic modes:

$$V_s = \text{span}\{\psi_1, .., \psi_{N_s}\} \tag{33}$$

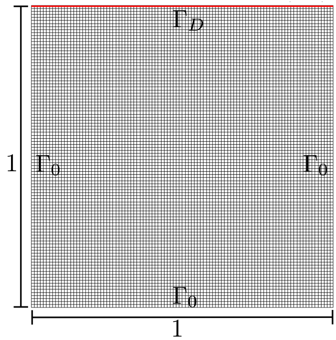
The final **enriched velocity space** $\tilde{V}_u$ will be then formed by the first N_u velocity modes and N_s supremizer modes

$$\tilde{V}_u = \text{span}\{\boldsymbol{\varphi_1}, ..., \boldsymbol{\varphi_{N_u}}\} \oplus \text{span}\{\boldsymbol{\psi_1}, .., \boldsymbol{\psi_{N_s}}\} \tag{34}$$

It has been heuristically verified that a number of supremizer modes equal or bigger respect to the number of pressure modes is usually enough to met the inf-sup condition.

The lid driven cavity problem

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

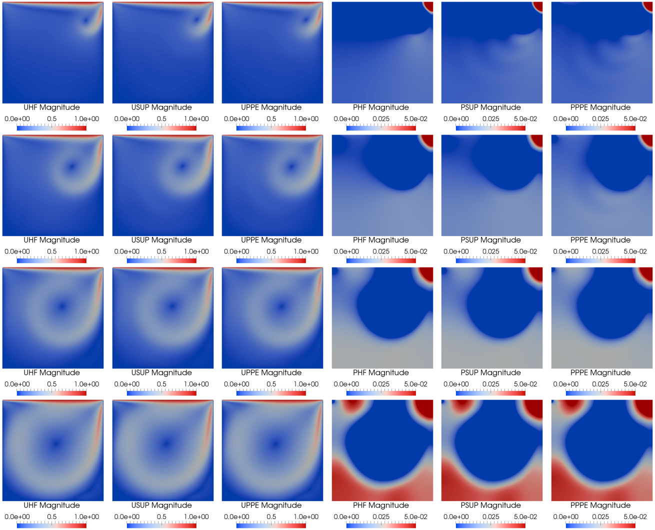


	Γ_D	Γ_0
$\boldsymbol{u}$	$\boldsymbol{u} = (1,0)$	$\boldsymbol{u} = (0,0)$
p	$\nabla p \cdot \boldsymbol{n} = 0$	$\nabla p \cdot \boldsymbol{n} = 0$

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

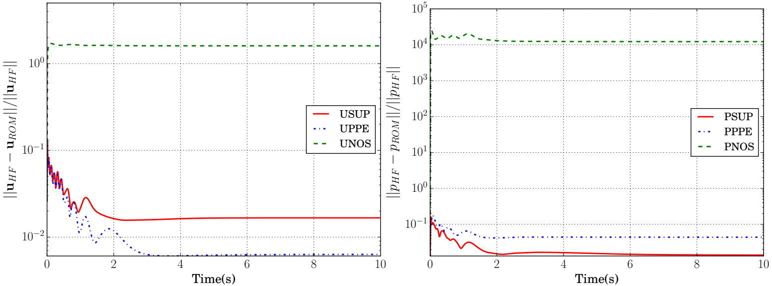
Numerical examples

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

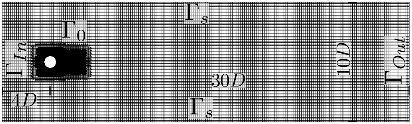
The L^2 norm of the relative error over time for three different models.

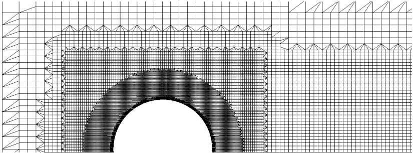


The table contains the cumulative eigenvalues for the lid driven cavity test. The last column contains the value of the inf-sup constant, in the supremizer stabilisation case, for different different number of supremizer modes and with a fixed number of velocity and pressure modes.

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

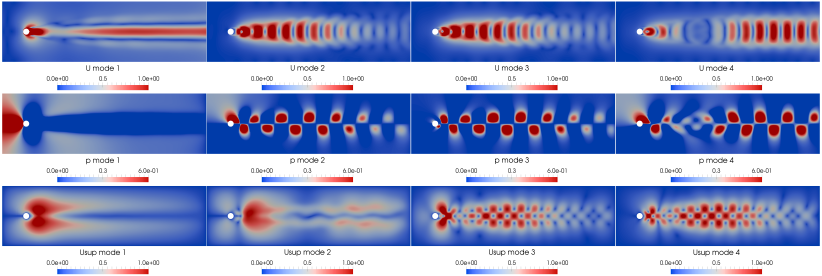




	Γ_{In}	Γ_0	Γ_s	Γ_{Out}
$\boldsymbol{u}$	$\boldsymbol{u} = (1,0)$	$\boldsymbol{u} = (0,0)$	$\boldsymbol{u} \cdot \boldsymbol{n} = 0$	$\nabla \boldsymbol{u} \cdot \boldsymbol{n} = 0$
p	$\nabla p \cdot \boldsymbol{n} = 0$	$\nabla p \cdot \boldsymbol{n} = 0$	$\nabla p \cdot \boldsymbol{n} = 0$	$p = 0$

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 $\text{Re} \in [100, 200]$.

First four modes for velocity pressure and supremizers

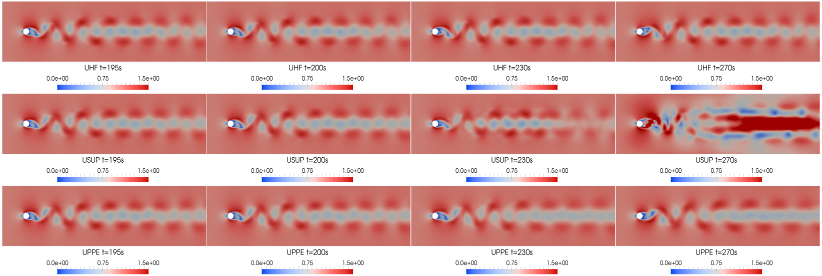


Cumulative eigenvalues

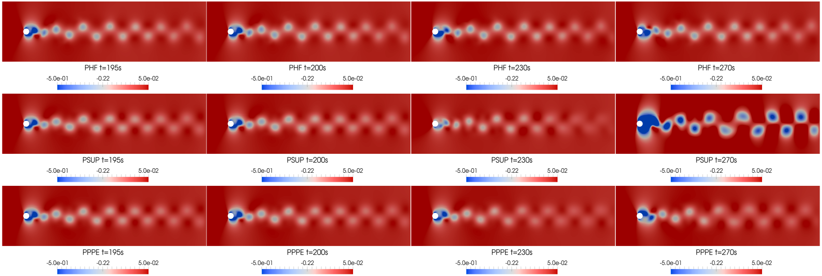
N Modes	***u***	***p***	***s***	β
1	0.390813	0.793239	0.921046	2.608e-04
2	0.598176	0.85809	0.941746	4.492e-04
3	0.802176	0.911636	0.961438	7.869e-03
4	0.879096	0.934997	0.978072	1.662e-02
5	0.949519	0.955578	0.98669	1.662e-02
10	0.986025	0.992347	0.998307	1.098e-01
15	0.995922	0.997994	0.999732	1.199e-01

Numerical examples

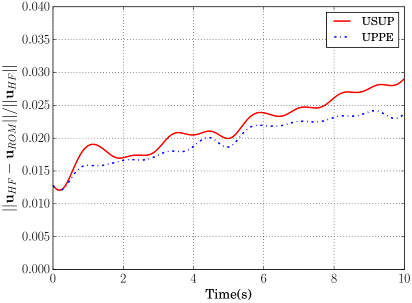
Comparison of the velocity field

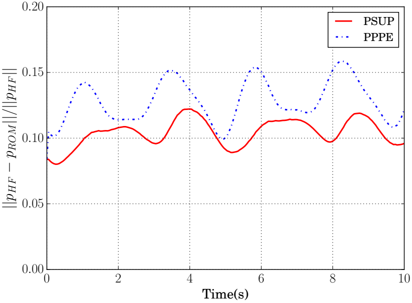


Comparison of the pressure field



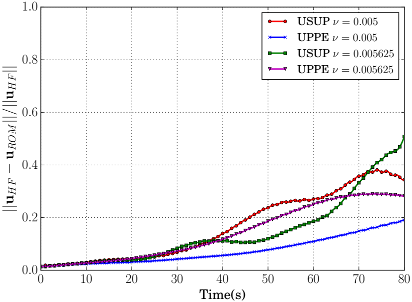
Comparison on the same time window

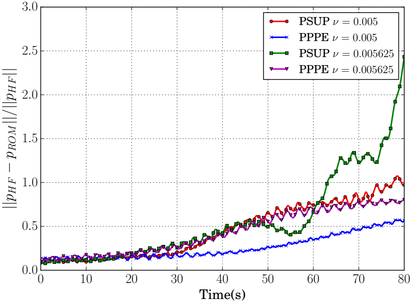




- The **velocity** field is reproduced in a more accurate way using the **Poisson equation** approach. This is due to the “pollution” given by the non-necessary supremizer modes.
- On the other side the **pressure** field is better reproduced using a supremizer approach.

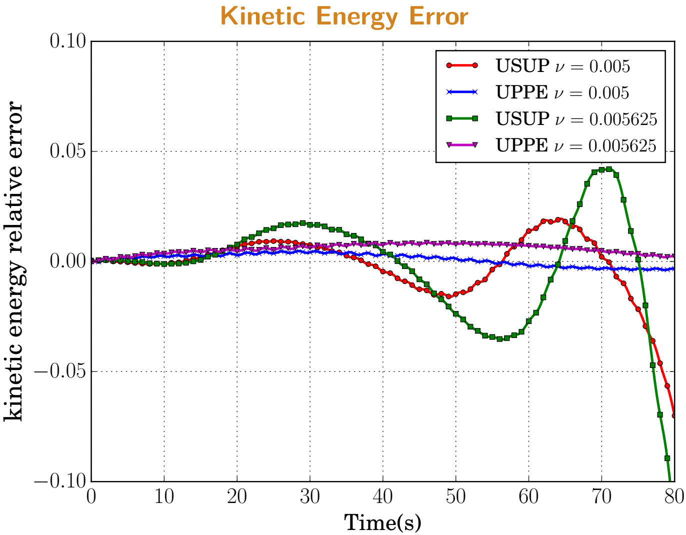
Comparison on a longer time window





- 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.

Numerical examples



Numerical examples

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 5.0.
- The **cylinder** example has run in parallel with OpenFOAM 5.0.
- The reduced order models have run in serial in **ITHACA-FV** (In real Time Highly Advanced Computational Applications for Finite Volumes) a C++ OpenFOAM based library (linear algebra is based on Eigen C++) that I developed. It is available on github (https://github.com/mathLab/ITHACA-FV).
- In the worst case the speed up is equal to approx. 200.

Conclusions

- A **supremizer stabilization** technique has been extended to a finite volume setting.
- The supremizer stabilization have demonstrated to provide **accurate results**.
- It leads to an **increase of the computational cost** respect to a PPE approach and into a less accurate reconstruction of pressure but permits to **avoid** (in some cases complex) **additional boundary conditions**.

Future Outlooks

- Introduce the **geometrical parametrization** in order to deal with **mesh motion** problems.
- Study efficient methods for **affine decomposition** of the differential operators.
- Investigate the **stability** of the ROM for **long-time integration**.
- **Increasing Reynolds** numbers (Poster Ali et al)
- Geometric parametrization for finite volumes
  - Mapping to a **reference Domain** - Not always possible
  - Exploitation of the **Immersed Boundary Method**

References

[1] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza. POD-Galerkin reduced order methods for CFD using finite volume discretisation: vortex shedding around a circular cylinder. Communications in Applied and Industrial Mathematics, 8(1), pp. 210-236, 2017.

[2] G. Stabile and G. Rozza, Stabilized Reduced order POD-Galerkin techniques for finite volume approximation of the parametrized Navier–Stokes equations, Computer & Fluids, in press, 2017.

[4] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, International Journal for Numerical Methods in Engineering, vol. 102, no. 5, pp. 1136–1161, 2015.

Thank you for the attention !!