
MOX-Report No. 32/2018

# An algebraic least squares reduced basis method for the solution of nonaffinely parametrized Stokes equations

Dal Santo, N.; Deparis, S.; Manzoni, A.; Quarteroni, A.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

# An algebraic least squares reduced basis method for the solution of nonaffinely parametrized Stokes equations

Niccolò Dal Santo[*], Simone Deparis[*], Andrea Manzoni[†], Alfio Quarteroni[†]

June 3, 2018

**Abstract**

In this paper we propose a new, purely algebraic, Petrov-Galerkin reduced basis (RB) method to solve the parametrized Stokes equations, where parameters serve to identify the (variable) domain geometry. Our method is obtained as an algebraic least squares reduced basis (aLS-RB) method, and improves the existing RB methods for Stokes equations in several directions. First of all, it does not require to enrich the velocity space, as often done when dealing with a velocity-pressure formulation, relying on a Petrov-Galerkin RB method rather than on a Galerkin RB (G-RB) method. Then, it exploits a suitable approximation of the matrix-norm in the definition of the (global) supremizing operator. The proposed method also provides a fully automated procedure to assemble and solve the RB problem, able to treat any kind of parametrization, and we rigorously prove the stability of the resulting aLS-RB problem (in the sense of a suitable inf-sup condition). Next, we introduce a *coarse* aLSRB (caLSRB) method, which is obtained by employing an approximated RB test space, and further improves the efficiency of the aLSRB method both offline and online. We provide numerical comparisons between the proposed methods and the current state-of-art G-RB methods. The new approach results in a more convenient option both during the offline and the online stage of computation, as shown by the numerical results.

## 1 Introduction

Solving saddle-point problems depending on a set of input parameters is a relevant task in several engineering contexts. In this work, as a special instance of such a problem, we consider the parametrized Stokes equations, which describe a steady viscous incompressible fluid when nonlinear convective terms are neglected. The numerical solution of the Stokes equations is in general a challenging task and a variety of methods have been proposed for their numerical approximation, see e.g. [20, 30, 34]. Here we deal with the efficient solution, by means of the reduced basis (RB) method, of parametrized steady Stokes equations

$$\begin{cases} -\nu^{\boldsymbol{\mu}}\Delta\vec{u}^{\boldsymbol{\mu}} + \nabla p^{\boldsymbol{\mu}} = \vec{f}^{\boldsymbol{\mu}} & \text{in } \Omega^{\boldsymbol{\mu}} \\ \nabla\cdot\vec{u}^{\boldsymbol{\mu}} = 0 & \text{in } \Omega^{\boldsymbol{\mu}} \\ + \text{ boundary conditions} \end{cases} \tag{1}$$

modeling the velocity $\vec{u}^{\boldsymbol{\mu}}$ and the pressure $p^{\boldsymbol{\mu}}$ of a viscous incompressible fluid with viscosity $\nu^{\boldsymbol{\mu}}$ in a domain $\Omega^{\boldsymbol{\mu}} \subset \mathbb{R}^d$, $d = 2, 3$. The vector of parameters $\boldsymbol{\mu}$ describes physical and/or geometrical properties of the system, whereas $\vec{f}^{\boldsymbol{\mu}}$ is the known right hand side collecting all the problem data; system (1) must be supplied with proper boundary conditions, which may also depend on $\boldsymbol{\mu}$.

In particular, we are interested in the efficient solution of (1) for several (say, thousands) instances of $\boldsymbol{\mu}$. This requirement – arising, e.g., when dealing with uncertainty quantification,

[1]Institute of Mathematics, Ecole Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland, niccolo.dalsanto@epfl.ch, simone.deparis@epfl.ch

[2]MOX, Department of Mathematics, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133 Milano, Italy, andrea1.manzoni@polimi.it, alfio.quarteroni@polimi.it

sensitivity analysis or PDE-constrained optimization – makes usual high-fidelity approximation techniques, such as the finite element (FE) method, extremely expensive, if not computationally unaffordable. Such a task can be more easily tackled by means of reduced-order modeling (ROM) techniques, such as the reduced basis (RB) method we focus on to deal with (1).

## 1.1 Existing RB methods for Stokes equations

The main idea of the RB method for parameter dependent partial differential equations (PDEs) is to approximate their solution by a linear combination of few global basis functions, obtained from a set of FE solutions (or *snapshots*) corresponding to different parameter values [32, 22]. This strategy is pursued in two stages: an *offline phase* and an *online phase*. In the former, we construct a RB space $V_N$ of dimension $N \ll N_h$ obtained by (properly orthonormalized) linear combinations of FE solutions of the parametrized PDE computed for different parameter values. In the latter, we require the projection of the residual of the FE problem onto the RB test space to vanish, obtaining a small problem which replaces the original high-fidelity FE problem.

Initially applied to linear elliptic PDEs, the RB method has been extended to saddle-point problems such as the Stokes equations, and extensively investigated in the past decade; a non-exhaustive list includes, among others: Stokes flows featuring affine [37, 21, 36] and nonaffine parameter dependence [35]; Navier-Stokes (NS) flows [17, 33, 18, 24]; parametrized optimal control problems [29] or shape optimization problems [26]. In all these cases, the RB method relies on:

1. a (weak) greedy algorithm for the incremental construction of the RB space, performed by selecting a new basis function for the velocity and the pressure at each step upon the use of a residual-based a posteriori error estimator.
2. a Galerkin projection onto the RB space to generate the RB problem (G-RB method).

Of course, this is not the only available choice. Regarding point 1., proper orthogonal decomposition (POD), rather than greedy algorithms, can be used to build the RB space. When such a strategy is employed, a set of FE solutions, called snapshots, are computed and the RB spaces for the velocity and the pressure are constructed, either jointly or separately, by performing the singular value decomposition of the resulting snapshot matrices, [3, 8, 12, 19, 23, 39]. This option has been considered, e.g., in [4] where two-dimensional Navier-Stokes flows on simple geometries affinely parametrized have been treated.

Concerning point 2., a more general Petrov-Galerkin (rather than Galerkin) projection – such as in the case of a least-squares (LS) method – can be performed, choosing a test space different from the trial space, see e.g. [14, 16]. This option has been first explored in the case of two-dimensional, affinely parametrized Stokes problems on simple geometries in [1]. In these works parameter-dependent domains $\Omega^{\boldsymbol{\mu}}$ were obtained as images of a reference domain $\Omega^0$ through a parameter-dependent map whose expression was known analytically. This is a relevant limitation toward the application of RB methods to more general domains with varying shape, not necessarily obtained in an explicit way from *a priori* known, parametrized deformations[1].

What makes the RB approximation of parametrized Stokes equations hard is ensuring the stability of the resulting RB problem. Indeed, it is well-known that in the FE case an *inf-sup* condition must be satisfied at the finite dimensional level to ensure the well-posedness of the numerical problem. This condition is fulfilled, for instance, if either $\mathbb{P}^2 - \mathbb{P}^1$ (Taylor-Hood) couples of FE spaces are used for discretizing the velocity and pressure fields, respectively, which is the FE discretization technique used in this work. Concerning the stability of the RB approximation, a stable couple of reduced subspaces for velocity and pressure, satisfying an equivalent *inf-sup* condition at the reduced level, ensures that the RB Stokes problem is well-posed. This property is not automatically fulfilled if the RB problem is constructed through a Galerkin projection employing RB spaces made solely of orthonormalized solutions of (1) obtained for different values of parameters. To overcome this shortcoming, two strategies have been designed.

A. The velocity space can be augmented by means of a set of *enriching* basis functions computed through the pressure supremizing operator, which depends on the divergence term. This yields a RB problem with additional degrees of freedom for the velocity field (as many as

[1] The proposed strategy can be meant as a step towards the efficient reduction of fluid-structure interaction problems, where domain deformations are computed through a structural problem coupled with a fluid one.

the pressure variable), see [36] for the details. In presence of $\boldsymbol{\mu}$-dependent domains, the supremizing operator is $\boldsymbol{\mu}$-dependent, so that to recover computational efficiency (and avoid the construction of the pressure supremizing operator online, for each $\boldsymbol{\mu}$), an offline enrichment is employed. This strategy leads to a RB problem which is *inf-sup* stable in practice, but whose well-posedness is not rigorously proven. Such a framework has been originally introduced in conjunction with a (weak) greedy algorithm [37, 21, 36], and later for the POD case. In the former case, for each selected pressure basis, a supremizing function is used to augment the velocity space. In the latter case, however, the basis functions are not directly related to any precise $\boldsymbol{\mu}$ instance, so that a set of enriching functions for the velocity space must be computed in advance starting from pressure snapshots, then applying POD to build the enriching basis [4]. A priori, it is not clear how many supremizing functions are needed to stabilize the problem; taking as many of them as the number of velocity and pressure basis is a working rule of thumb. Galerkin projection is employed to build a ROM in the case of unsteady Navier-Stokes equations, too – without, however, dealing with physical/geometrical parameters – in [12], where the employed divergence-free velocity basis functions ensure the well-posedness of the reduced problem. In the problems we focus on, instead, stability is not automatically guaranteed since we deal with parametrized domains and the resulting velocity basis functions are not, in principle, divergence-free.

B. We can exploit a Petrov-Galerkin (PG) method [1, 32] to build an automatically stable RB problem, relying, for instance, on the least squares (LS) method. The resulting LS-RB method uses a test space which is obtained as the image of the RB space through a global supremizing operator involving both velocity and pressure fields. This approach leads to an automatically stable RB problem, which satisfies the required *inf-sup* condition [32]. However, the existing formulation of the LS-RB method for Stokes equations proposed in [1] presumes the existence of an explicit $\boldsymbol{\mu}-$dependent function which enables to recast the problem on a reference domain. Without this function available, as in the case where domain deformation results from the solution of a further FE problem, the computational work to build the RB problem is unbearable, see Section 3.2.2.

Eventually, in the case a stabilized finite element discretization (like, e.g., a $\mathbb{P}^1 - \mathbb{P}^1$ Streamline Upwind Petrov-Galerkin (SUPG) finite element method) is employed, the resulting Galerkin RB approximation is well-posed; neither an enrichment of the velocity space, nor a LS-RB formulation, are necessary to ensure the stability of the corresponding RB system. This option, however, is not the focus of our current investigation.

## 1.2 Novelties of the proposed approach

In this work we propose a new family of purely algebraic PG-RB methods to address large-scale parametrized Stokes equations in domains with varying geometry. Our method can be seen as an algebraic LS-RB method; for this reason we refer to it as aLS-RB method. The aLS-RB method extends and improves the existing RB methods for Stokes equations in several directions, potentially becoming a paradigm for the *efficient* construction of a *stable* and *accurate* RB method for Stokes equations and, more generally speaking, weakly coercive problems. Indeed:

1. it relies on POD for the construction of the RB spaces for velocity and pressure; however, like in the greedy case, an exponential decay of the residuals with respect to $N$ is obtained;
2. it does not need an enrichment of the velocity space, by relying on a Petrov-Galerkin method;
3. it exploits suitable approximations of the matrix-norm to define the supremizing operator;
4. the resulting aLS-RB problem rigorously is *inf-sup* stable;
5. it does not require an analytical map between a reference and the physical domain $\Omega^{\boldsymbol{\mu}}$;
6. it provides a fully automated procedure to assemble and solve the RB problem, able to treat any kind of parametrization.

When an affine decomposition of the RB arrays is employed, the aLSRB approximation leads to a formulation which depends quadratically on the number of affine terms of the RB matrix and the right hand side. To further enhance the efficiency of the method, we propose a *coarse* aLSRB (caLSRB) option, where a suitable approximation of the RB matrix is employed when defining the

RB test space. The resulting caLSRB approximation enjoys the same accuracy of the one provided by the aLSRB method, but allows a more efficient assembling phase, since it depends linearly - and not quadratically - on the number of affine terms, similarly to the Galerkin case.

The paper is organized as follows. After recalling in Section 2 the Stokes equations and their FE approximation, in Section 3 we introduce their RB approximation, highlighting how to construct a stable RB method either using Galerkin or Petrov-Galerkin projections. In Section 4, we present the new (c)aLS-RB methods and analyse the well-posedness of the resulting RB formulations, showing that each of them satisfies an *inf-sup* condition. To enhance the efficiency of the proposed methods, discrete empirical interpolation method (DEIM) [5, 15], and Matrix DEIM (M-DEIM) [28] are employed to find an affine approximation of the (generally nonaffine) right hand sides and matrices. In Section 6 we present numerical results obtained with the different RB methods on a three-dimensional Stokes problem defined over parameter-dependent domains whose deformations with respect to a reference domain are not necessarily known analytically. In Section 7 we finally draw some conclusions.

# 2 Parametrized Stokes equations: setting and preliminaries

In this section we introduce the Stokes equations in parametrized domains, their weak formulation, and their FE approximation. Concerning notation, hereon we denote scalar fields by lower case letters, as $a \in \mathbb{R}$, vector fields with an arrow, as $\vec{a} \in \mathbb{R}^d$, for $d = 2, 3$, vectors (like finite elements vectors) by bold lower case letters, as $\mathbf{a} \in \mathbb{R}^n$, and matrices by bold capital letters, as $\mathbf{A} \in \mathbb{R}^{n\times n}$. We denote by $(\cdot,\cdot)_2$ the Euclidean scalar product and by $\mathcal{K}_2(\mathbf{A})$ the condition number of $\mathbf{A}$ with respect the Euclidean norm. Moreover, given a symmetric and positive definite matrix $\mathbf{Y} \in \mathbb{R}^{n\times n}$, we denote by $(\cdot,\cdot)_{\mathbf{Y}}$ the scalar product and by $\|\cdot\|_{\mathbf{Y}}$ the norm defined as $(\mathbf{a},\mathbf{b})_{\mathbf{Y}} = \mathbf{a}^T\mathbf{Y}\mathbf{b}\ \forall\, \mathbf{a},\ \mathbf{b} \in \mathbb{R}^n$ and $\|\mathbf{a}\|_{\mathbf{Y}} = \sqrt{(\mathbf{a},\mathbf{a})_{\mathbf{Y}}}\ \forall\, \mathbf{a} \in \mathbb{R}^n$, respectively. We also denote by $\mathcal{K}_{\mathbf{Y}}(\mathbf{A})$ the condition number of $\mathbf{A}$ with respect to $\|\cdot\|_{\mathbf{Y}}$. Finally, we indicate by $\mathcal{D} \subset \mathbb{R}^l$, $l \in \mathbb{N}$ the parameter space and by $\boldsymbol{\mu} \in \mathcal{D}$ a vector of parameters; the superscript $\boldsymbol{\mu}$ highlights $\boldsymbol{\mu}$-dependence.

Given a $\boldsymbol{\mu}-$dependent domain $\Omega^{\boldsymbol{\mu}} \subset \mathbb{R}^d$, $d = 2,\ 3$, such that $\partial\Omega^{\boldsymbol{\mu}} = \Gamma^{\boldsymbol{\mu}}_{out} \cup \Gamma^{\boldsymbol{\mu}}_{in} \cup \Gamma^{\boldsymbol{\mu}}_{w}$, $\mathring{\Gamma}^{\boldsymbol{\mu}}_{out} \cap \mathring{\Gamma}^{\boldsymbol{\mu}}_{in} = \mathring{\Gamma}^{\boldsymbol{\mu}}_{w} \cap \partial\mathring{\Gamma}^{\boldsymbol{\mu}}_{in} = \mathring{\Gamma}^{\boldsymbol{\mu}}_{out} \cap \mathring{\Gamma}^{\boldsymbol{\mu}}_{w} = \emptyset$ and $\Gamma^{\boldsymbol{\mu}}_{out} \neq \partial\Omega$, for any $\boldsymbol{\mu} \in \mathcal{D}$, the Stokes equations read

$$\begin{cases} -\nu^{\boldsymbol{\mu}}\Delta\vec{u}^{\boldsymbol{\mu}} + \nabla p^{\boldsymbol{\mu}} = \vec{f}^{\boldsymbol{\mu}} & \text{in } \Omega^{\boldsymbol{\mu}} \\ \nabla\cdot\vec{u}^{\boldsymbol{\mu}} = 0 & \text{in } \Omega^{\boldsymbol{\mu}} \\ \vec{u} = \vec{g}^{\boldsymbol{\mu}}_D & \text{on } \Gamma^{\boldsymbol{\mu}}_{in} \\ \vec{u} = \vec{0} & \text{on } \Gamma^{\boldsymbol{\mu}}_{w} \\ -p^{\boldsymbol{\mu}}\vec{n}^{\boldsymbol{\mu}} + \nu^{\boldsymbol{\mu}}\dfrac{\partial\vec{u}^{\boldsymbol{\mu}}}{\partial\vec{n}^{\boldsymbol{\mu}}} = \vec{g}^{\boldsymbol{\mu}}_N & \text{on } \Gamma^{\boldsymbol{\mu}}_{out}, \end{cases} \tag{2}$$

where $(\vec{u}^{\boldsymbol{\mu}}, p^{\boldsymbol{\mu}})$ are the velocity and pressure fields of a viscous incompressible Newtonian fluid with viscosity $\nu^{\boldsymbol{\mu}}$, respectively. We introduce a (regular enough) lifting function $\vec{r}^{\boldsymbol{\mu}}_{\vec{g}_D} \in \left(H^1(\Omega^{\boldsymbol{\mu}})\right)^d$ and the following $\boldsymbol{\mu}$-dependent spaces

$$V^{\boldsymbol{\mu}} = \left\{\vec{v} \in \left(H^1(\Omega^{\boldsymbol{\mu}})\right)^d : \vec{v}\big|_{\Gamma^{\boldsymbol{\mu}}_w} = \vec{v}\big|_{\Gamma^{\boldsymbol{\mu}}_{in}} = \vec{0}\right\},$$
$$Q^{\boldsymbol{\mu}} = L^2(\Omega^{\boldsymbol{\mu}}) \quad \text{or} \quad Q^{\boldsymbol{\mu}} = L^2_0(\Omega^{\boldsymbol{\mu}}) \text{ if } \Gamma^{\boldsymbol{\mu}}_{out} = \emptyset,$$

equipped with scalar products $(\cdot,\cdot)_{V^{\boldsymbol{\mu}}} = (\cdot,\cdot)_{(H^1_0(\Omega^{\boldsymbol{\mu}}))^d}$ and $(\cdot,\cdot)_{Q^{\boldsymbol{\mu}}} = (\cdot,\cdot)_{L^2(\Omega^{\boldsymbol{\mu}})}$. For a given $\boldsymbol{\mu} \in \mathcal{D}$, the weak formulation of problem (2) reads: find $(\vec{u}^{\boldsymbol{\mu}}, p^{\boldsymbol{\mu}}) \in V^{\boldsymbol{\mu}} \times Q^{\boldsymbol{\mu}}$ such that

$$\begin{cases} a^{\boldsymbol{\mu}}(\vec{u}^{\boldsymbol{\mu}},\vec{v}) + b^{\boldsymbol{\mu}}(\vec{v}, p^{\boldsymbol{\mu}}) = f^{\boldsymbol{\mu}}(\vec{v}) \qquad \forall \vec{v} \in V^{\boldsymbol{\mu}} \\ b^{\boldsymbol{\mu}}(\vec{u}^{\boldsymbol{\mu}}, q) = -b^{\boldsymbol{\mu}}(\vec{r}^{\boldsymbol{\mu}}_{\vec{g}_D}, q) \qquad \forall q \in Q^{\boldsymbol{\mu}}, \end{cases} \tag{3}$$

where we define the forms in (3) for $\vec{u}, \vec{v} \in V^{\boldsymbol{\mu}}$, $q \in Q^{\boldsymbol{\mu}}$, as

$$a^{\boldsymbol{\mu}}(\vec{u},\vec{v}) = \int_{\Omega^{\boldsymbol{\mu}}} \nu^{\boldsymbol{\mu}}\nabla\vec{u} : \nabla\vec{v}\, d\Omega^{\boldsymbol{\mu}}, \qquad b^{\boldsymbol{\mu}}(\vec{v}, q) = -\int_{\Omega^{\boldsymbol{\mu}}} q\nabla\cdot\vec{v}\, d\Omega^{\boldsymbol{\mu}}$$
$$f^{\boldsymbol{\mu}}(\vec{v}) = \int_{\Omega^{\boldsymbol{\mu}}} \vec{f}^{\boldsymbol{\mu}}\cdot\vec{v}\, d\Omega^{\boldsymbol{\mu}} + \int_{\Gamma^{\boldsymbol{\mu}}_{out}} \vec{g}^{\boldsymbol{\mu}}_N\cdot\vec{v}\, d\Gamma^{\boldsymbol{\mu}}_{out} - a^{\boldsymbol{\mu}}(\vec{r}^{\boldsymbol{\mu}}_{\vec{g}_D}, \vec{v}).$$

Problem (3) can be written as a symmetric non-coercive problem, provided we define the space $X^{\boldsymbol{\mu}} = V^{\boldsymbol{\mu}} \times Q^{\boldsymbol{\mu}}$, equipped with the scalar product and the norm

$$((\vec{u},p),(\vec{v},q))_{X^{\boldsymbol{\mu}}} = (\vec{u},\vec{v})_{V^{\boldsymbol{\mu}}} + (p,q)_{Q^{\boldsymbol{\mu}}}, \qquad (\vec{u},p),(\vec{v},q) \in X^{\boldsymbol{\mu}},$$
$$\|(\vec{v},q)\|_{X^{\boldsymbol{\mu}}} = \sqrt{((\vec{v},q),(\vec{v},q))_{X^{\boldsymbol{\mu}}}}, \qquad (\vec{v},q) \in X^{\boldsymbol{\mu}},$$

respectively. By introducing the forms $\mathcal{A}^{\boldsymbol{\mu}} : X^{\boldsymbol{\mu}} \times X^{\boldsymbol{\mu}} \to \mathbb{R}$, $\mathcal{F}^{\boldsymbol{\mu}} : X^{\boldsymbol{\mu}} \to \mathbb{R}$ given by

$$\mathcal{A}^{\boldsymbol{\mu}}((\vec{u}^{\boldsymbol{\mu}},p^{\boldsymbol{\mu}}),(\vec{v},q)) = a^{\boldsymbol{\mu}}(\vec{u},\vec{v}) + b^{\boldsymbol{\mu}}(\vec{v},p^{\boldsymbol{\mu}}) + b^{\boldsymbol{\mu}}(\vec{u}^{\boldsymbol{\mu}},q),$$
$$\mathcal{F}^{\boldsymbol{\mu}}((\vec{v},q)) = f^{\boldsymbol{\mu}}(\vec{v}) - b^{\boldsymbol{\mu}}(\vec{r}^{\boldsymbol{\mu}}_{\vec{g}_D},q),$$

system (3) can be equivalently written as: find $\vec{z}^{\boldsymbol{\mu}} \in X^{\boldsymbol{\mu}}$ such that

$$\mathcal{A}^{\boldsymbol{\mu}}(\vec{z}^{\boldsymbol{\mu}},\vec{w}) = \mathcal{F}^{\boldsymbol{\mu}}(\vec{w}) \qquad \forall \vec{w} \in X^{\boldsymbol{\mu}}. \tag{4}$$

The well-posedness of problem (4) is ensured according to the general theory of saddle-point problems see, e.g., [10, 11].

## 2.1 Finite element approximation of the Stokes equations

Numerical methods based on (Petrov-)Galerkin projection onto a finite dimensional subspace, as the finite element (FE) or spectral element methods, represent a successful technique to handle the numerical approximation of (2), see e.g. [13, 20, 31]. However, when they are employed, a discrete *inf-sup* condition must be satisfied to ensure the well-posedness of the numerical problem.

The ROMs considered in this paper for the efficient solution of the parametrized problem (4) hinge upon a high-fidelity finite element approximation, which we introduce in this section. We consider a domain deformation dependent on $\boldsymbol{\mu}$; the corresponding meshes are also taken as a deformation of a reference mesh, hence not affecting the topology of the degrees of freedom.

Let us denote by $V_h^{\boldsymbol{\mu}}$ and $Q_h^{\boldsymbol{\mu}}$ two finite dimensional FE spaces of dimension $N_h^u$ and $N_h^p$, respectively, with $V_h^{\boldsymbol{\mu}} \subset V$ and $Q_h^{\boldsymbol{\mu}} \subset Q$. Moreover, set $X_h^{\boldsymbol{\mu}} = V_h^{\boldsymbol{\mu}} \times Q_h^{\boldsymbol{\mu}}$ with $N_h = N_h^u + N_h^p$. Note that, for any given $\boldsymbol{\mu}$, $X_h^{\boldsymbol{\mu}}$ denotes a single FE space rather than a family of spaces depending on $h$. Indeed, in this work we do not focus on $h$-refinement; rather, we consider a fixed regular mesh which is fine enough for the problem at hand.

The FE approximation of problem (4) reads: find $\vec{z}_h^{\boldsymbol{\mu}} \in X_h^{\boldsymbol{\mu}}$ such that

$$\mathcal{A}^{\boldsymbol{\mu}}(\vec{z}_h^{\boldsymbol{\mu}},\vec{w}_h) = \mathcal{F}^{\boldsymbol{\mu}}(\vec{w}_h) \qquad \forall \vec{w}_h \in X_h^{\boldsymbol{\mu}}. \tag{5}$$

Problem (5) can be equivalently written as a parametrized linear system

$$\mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{z}_h^{\boldsymbol{\mu}} = \mathbf{g}_h^{\boldsymbol{\mu}} \tag{6}$$

featuring a saddle-point structure, where

$$\mathbf{A}_h^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{D}_h^{\boldsymbol{\mu}} & (\mathbf{B}_h^{\boldsymbol{\mu}})^T \\ \mathbf{B}_h^{\boldsymbol{\mu}} & 0 \end{bmatrix} \in \mathbb{R}^{N_h \times N_h}, \qquad \mathbf{z}_h^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{u}_h^{\boldsymbol{\mu}} \\ \mathbf{p}_h^{\boldsymbol{\mu}} \end{bmatrix} \in \mathbb{R}^{N_h} \qquad \text{and} \qquad \mathbf{g}_h^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{f}_h^{\boldsymbol{\mu}} \\ \mathbf{r}_h^{\boldsymbol{\mu}} \end{bmatrix} \in \mathbb{R}^{N_h}. \tag{7}$$

More precisely, $\mathbf{D}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^u \times N_h^u}$, $\mathbf{B}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^p \times N_h^u}$, $\mathbf{f}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^u}$ and finally $\mathbf{r}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^p}$. The solution of (6) usually exploits suitable iterative methods properly preconditioned, [6, 7]. Several techniques relying on, e.g., domain decomposition, multilevel methods and block factorizations have been proposed as preconditioners, see e.g. [20, 30, 38, 34] and references therein. Furthermore, we introduce the symmetric and positive definite matrix $\mathbf{X}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ is associated to the scalar product $(\cdot,\cdot)_{X^{\boldsymbol{\mu}}}$ on the space $X_h^{\boldsymbol{\mu}}$ and is built as a block diagonal matrix of the form

$$\mathbf{X}_h^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{X}_u^{\boldsymbol{\mu}} & 0 \\ 0 & \mathbf{X}_p^{\boldsymbol{\mu}} \end{bmatrix}; \tag{8}$$

$\mathbf{X}_u^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^u \times N_h^u}$ and $\mathbf{X}_p^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^p \times N_h^p}$ is associated to the $(H_0^1(\Omega^{\boldsymbol{\mu}}))^d$ and $L^2(\Omega^{\boldsymbol{\mu}})$ scalar products on $V_h^{\boldsymbol{\mu}}$ and $Q_h^{\boldsymbol{\mu}}$, respectively. Since the computational domain is $\boldsymbol{\mu}-$dependent, the matrix $\mathbf{X}_h^{\boldsymbol{\mu}}$ depends on $\boldsymbol{\mu}$, too. Then, the well-posedness of (6) is ensured by the following assumption.

**Assumption 2.1.** *There exists* $\beta^{min} > 0$ *such that*

$$\beta_h^{\boldsymbol{\mu}} = \inf_{\vec{z}_h \in \mathbb{R}^{N_h}} \sup_{\vec{w}_h \in \mathbb{R}^{N_h}} \frac{\mathbf{w}_h^T \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{z}_h}{\|\mathbf{z}_h\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \|\mathbf{w}_h\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \beta^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{9}$$

A couple of FE spaces which fulfills condition (9) is given, for instance, by $\mathbb{P}^2 - \mathbb{P}^1$ (Taylor-Hood) finite elements, for velocity and pressure, respectively; see, e.g., [20].

# 3 POD-based RB methods for the parametrized Stokes equations

The RB method represents a convenient framework for the reduction of parametrized PDEs [32]. It first generates a low-dimensional subspace where the RB solution is sought; then, it employs a Galerkin (or Petrov-Galerkin) projection onto this subspace to obtain the corresponding RB problem. Here we rely upon proper orthogonal decomposition (POD) in order to avoid the evaluation of a-posteriori error bounds, which would instead be required if a greedy algorithm were performed. Then, a new algebraic PGRB method, in two different versions, is investigated for the sake of the construction of the RB problem, and compared to the (indeed, more classical) Galerkin-RB method which relies on properly enriched RB spaces. In the following, we recall the essential elements of this technique and how it is used to build a RB approximation.

## 3.1 Proper orthogonal decomposition

In this section we recall POD, that we employ both for the construction of RB spaces (see Section 3.2) and for the efficient approximation of $\boldsymbol{\mu}$-dependent arrays (see Section 4.4). For simplicity here we focus on the former aspect; see, e.g., [28] for further insights on the latter.

Let us consider a set of $n_s$ FE vectors $\{\mathbf{s}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s} \subset \mathbb{R}^{N_h}$ (called *snapshots*) collected as columns of a matrix $\mathbf{S} = [\mathbf{s}^{\boldsymbol{\mu}_1} | \dots | \mathbf{s}^{\boldsymbol{\mu}_{n_s}}] \in \mathbb{R}^{N_h \times n_s}$. For any prescribed dimension $N$, the POD allows to find the $N$-dimensional subspace, spanned by the columns of $\mathbf{V} = [\boldsymbol{\xi}_1 | \dots | \boldsymbol{\xi}_N] \in \mathbb{R}^{N_h \times N}$, which best approximates $\{\mathbf{s}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ among all possible $N$-dimensional subspaces. The POD method takes advantage of the singular value decomposition (SVD) of the matrix $\mathbf{S}$

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^T, \tag{10}$$

with $\mathbf{U} \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{Z} \in \mathbb{R}^{n_s \times n_s}$ orthogonal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{N_h \times n_s}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_s} \geq 0$. Then, $\mathbf{V}$ is provided by the first $N$ columns of $\mathbf{U}$, which form by construction an orthonormal basis (with respect to the Euclidean scalar product) for the best $N$-dimensional approximation subspace. Similarly, one can obtain the SVD with repect to a scalar product induced by a symmetric positive definite matrix $\mathbf{X}$ by considering the SVD of $\mathbf{X}^{\frac{1}{2}}\mathbf{S}$, resulting in a $\mathbf{X}$-orthonormal basis. In particular, by denoting $\mathbf{I}_N$ the $N$-dimensional identity matrix, the following proposition holds [32].

**Proposition 3.1.** *Let* $\mathcal{V}_N = \{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T\mathbf{X}\mathbf{W} = \mathbf{I}_N\}$. *Then*

$$\sum_{i=1}^{n_s} \|\mathbf{s}^{\boldsymbol{\mu}_i} - \mathbf{V}\mathbf{V}^T\mathbf{X}\mathbf{s}^{\boldsymbol{\mu}_i}\|_{\mathbf{X}}^2 = \min_{\mathbf{W} \in \mathcal{V}_N} \sum_{i=1}^{n_s} \|\mathbf{s}^{\boldsymbol{\mu}_i} - \mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{s}^{\boldsymbol{\mu}_i}\|_{\mathbf{X}}^2 = \sum_{i=N+1}^{n_s} \sigma_i^2.$$

Notice that the columns of any $\mathbf{V} \in \mathcal{V}_N$ form a $\mathbf{X}$-orthonormal basis of a $N$-dimensional subspace of $\mathbb{R}^{N_h}$ and that in general the singular values $\sigma_i$, $i = 1, \dots, n_s$ depend on the choice of the matrix $\mathbf{X}$, that is, $\sigma_i = \sigma_i(\mathbf{X})$, $i = 1, \dots, n_s$, and the relative error on all the snapshots is related to $\{\sigma_i\}_i^{n_s}$ through the following relation

$$\frac{\sum_{i=1}^{n_s} \|\mathbf{s}^{\boldsymbol{\mu}_i} - \mathbf{V}\mathbf{V}^T\mathbf{X}\mathbf{s}^{\boldsymbol{\mu}_i}\|_{\mathbf{X}}^2}{\sum_{i=1}^{n_s} \|\mathbf{s}^{\boldsymbol{\mu}_i}\|_{\mathbf{X}}^2} = \frac{\sum_{i=N+1}^{n_s} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2}. \tag{11}$$

From a practical perspective, the POD basis construction is performed by means of Algorithm 1; for a given tolerance $\delta_{\text{POD}}$, (11) is employed to control the relative error on the approximation of the snapshots and to select $N$ basis functions. Alternatively, one could directly provide a dimension $N$ instead of $\delta_{\text{POD}}$, leading to a similar algorithm POD($\mathbf{S}$, $\mathbf{X}$, $N$).

**Algorithm 1** POD

**procedure** POD($\mathbf{S}$, $\mathbf{X}$, $\delta_{\text{POD}}$)
  form the correlation matrix $\mathbf{C}_{n_s} = \mathbf{S}^T\mathbf{X}\mathbf{S}$
  solve the eigenvalue problem $\mathbf{C}_{n_s}\boldsymbol{\psi_i} = \sigma_i^2\boldsymbol{\psi_i}, \quad i = 1, \dots, n_s$ and set $\boldsymbol{\xi}_i = \frac{1}{\sigma_i}\mathbf{S}\boldsymbol{\psi_i}$
  define $N$ as the minimum integer such that $\frac{\sum_{i=1}^{N}\sigma_i^2}{\sum_{i=1}^{n_s}\sigma_i^2} > 1 - \delta_{\text{POD}}^2$ and $\mathbf{V} = [\boldsymbol{\xi}_1|\dots|\boldsymbol{\xi}_N]$
**end procedure**

### 3.2 Projection-based RB methods

The RB method relies on the idea that the solution of the parametrized system (6), for a given value of $\boldsymbol{\mu}$, can be well approximated by a linear combination of $N$ basis functions $\{\vec{\xi}_i\}_{i=1}^N$ obtained by orthonormalizing the solutions of the same problem for other values of the parameter. The basis functions are collected in the so-called RB space, which is defined as

$$V_N = span\{\vec{\xi}_i,\, i = 1, \dots, N\} \tag{12}$$

and is of dimension $N \ll N_h$. From an algebraic standpoint, $V_N$ is represented by the matrix $\mathbf{V} = [\boldsymbol{\xi}_1|\dots|\boldsymbol{\xi}_N] \in \mathbb{R}^{N_h \times N}$, where $\boldsymbol{\xi}_i, i = 1, \dots, N$ are the FE vector representation of the basis $\vec{\xi}_i, i = 1, \dots, N$. From a practical standpoint, the vector basis $\{\boldsymbol{\xi}_i\}_{i=1}^N$ is constructed by applying POD on a set of snapshots given by $n_s$ solutions of the FE linear system (6) for $n_s$ instances of the parameter, that is, $\mathbf{s}^{\boldsymbol{\mu}_i} = \mathbf{z}_h^{\boldsymbol{\mu}_i}$, $i = 1, \dots, n_s$. Then, the RB approximation is constructed by introducing a set of (possibly $\boldsymbol{\mu}-$dependent) functions $\{w_i^{\boldsymbol{\mu}}\}_{i=1}^N$ such that a test space $W_N^{\boldsymbol{\mu}}$ is obtained as

$$W_N^{\boldsymbol{\mu}} = span\{w_i^{\boldsymbol{\mu}},\, i = 1, \dots, N\}.$$

Algebraically, $W_N^{\boldsymbol{\mu}}$ is represented by a matrix $\mathbf{W}^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N}$, which is generally different from $\mathbf{V}$ and might be $\boldsymbol{\mu}$-dependent. If $\mathbf{W}^{\boldsymbol{\mu}} \neq \mathbf{V}$ we have the more general Pevtrov Galerkin-RB approximation, otherwise if $\mathbf{W}^{\boldsymbol{\mu}} = \mathbf{V}$ we come up with the Galerkin case. For the sake of generality, we consider here the PGRB problem, which reads: find $\vec{z}_N^{\boldsymbol{\mu}} \in V_N$ such that

$$\mathcal{A}^{\boldsymbol{\mu}}(\vec{z}_N^{\boldsymbol{\mu}}, \vec{w}_N) = \mathcal{F}^{\boldsymbol{\mu}}(\vec{w}_N) \qquad \forall \vec{w}_N \in W_N^{\boldsymbol{\mu}}. \tag{13}$$

Problem (13) leads to the following algebraic RB linear system

$$\mathbf{A}_N^{\boldsymbol{\mu}}\mathbf{z}_N^{\boldsymbol{\mu}} = \mathbf{g}_N^{\boldsymbol{\mu}}, \tag{14}$$

where the RB matrix $\mathbf{A}_N^{\boldsymbol{\mu}} \in \mathbb{R}^{N \times N}$ and the RB right hand side $\mathbf{g}_N^{\boldsymbol{\mu}} \in \mathbb{R}^N$ are defined as

$$\mathbf{A}_N^{\boldsymbol{\mu}} = (\mathbf{W}^{\boldsymbol{\mu}})^T\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}, \qquad \mathbf{g}_N^{\boldsymbol{\mu}} = (\mathbf{W}^{\boldsymbol{\mu}})^T\mathbf{g}_h^{\boldsymbol{\mu}}. \tag{15}$$

We highlight that the PGRB approximation depends on the choice of the test space $W_N^{\boldsymbol{\mu}}$. As remarked above, the matrix $\mathbf{V}$ is built employing POD, and results in

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{N_u} & 0 \\ 0 & \mathbf{V}_{N_p} \end{bmatrix} = \left[\boldsymbol{\xi}_1|\dots|\boldsymbol{\xi}_{N_u}|\boldsymbol{\xi}_{N_u+1}|\dots|\boldsymbol{\xi}_N\right], \tag{16}$$

where $\mathbf{V}_{N_u} \in \mathbb{R}^{N_h^u \times N_u}$ and $\mathbf{V}_{N_p} \in \mathbb{R}^{N_h^p \times N_p}$ encode the subspaces to approximate the velocity $\mathbf{u}_h^{\boldsymbol{\mu}}$ and the pressure $\mathbf{p}_h^{\boldsymbol{\mu}}$, respectively.

In particular

$$\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{\varphi}_i^u \\ \mathbf{0} \end{bmatrix} \quad i = 1, \dots, N_u, \qquad\qquad \boldsymbol{\xi}_{N_u+i} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\psi}_i^p \end{bmatrix} \quad i = 1, \dots, N_p,$$

where $\left\{\boldsymbol{\varphi}_i\right\}_i^{N_u}$ and $\left\{\boldsymbol{\psi}_i\right\}_i^{N_p}$ are the FE vector representation of the basis functions for the velocity and the pressure RB space, that is,

$$\mathbf{V}_{N_u} = \left[\boldsymbol{\varphi}_1^u|\dots|\boldsymbol{\varphi}_{N_u}^u\right], \qquad \mathbf{V}_{N_p} = \left[\boldsymbol{\psi}_1^p|\dots|\boldsymbol{\psi}_{N_p}^p\right].$$

The construction of the RB spaces is thus performed by first collecting a set of FE snapshots $\{\mathbf{u}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, $\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, solutions of (6) for different instances of the parameters $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$, and then performing POD separately on the two spaces, thus yielding

$$\mathbf{V}_{N_u} = \mathrm{POD}\Big(\mathbf{S}_{\vec{u}}, \mathbf{X}_u, \delta_{RB}\Big), \qquad \mathbf{V}_{N_p} = \mathrm{POD}\Big(\mathbf{S}_p, \mathbf{X}_p, \delta_{RB}\Big),$$

where $\delta_{RB}$ is a positive tolerance employed to control the relative error on the snapshots approximation. The matrices $\mathbf{V}_{N_u}$ and $\mathbf{V}_{N_p}$ are constructed by selecting the largest $N_u$ and $N_p$ eigenmodes respectively, according to Algorithm 1, finally obtaining $N_u + N_p = N$, see [32]. Notice that a priori $N_u \neq N_p$. The trial RB spaces are defined by considering the sets of basis $\{\vec{\varphi}_i\}_{i=1}^{N_u}$ and $\{\psi_i\}_{i=1}^{N_p}$, whose FE vector representations are given by $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_u}$ and $\{\boldsymbol{\psi}_i\}_{i=1}^{N_p}$. Then, we define

$$V_{N_u} = span\{\vec{\varphi}_i,\, i = 1, \ldots, N_u\} \qquad Q_{N_p} = span\{\psi_i,\, i = 1, \ldots, N_p\},$$

and $V_N = V_{N_u} \times Q_{N_p}$. We finally remark that the dimension $N = N_u + N_p$ of the RB system is smaller than the dimension $N_h$ of the FE linear system of several orders of magnitude: $N \ll N_h$, so that problem (14) is solved by direct methods.

In order to obtain a well-posed RB approximation, an *inf-sup* condition equivalent to (2.1) at the RB level must also be satisfied. Several ways to produce a well-posed Stokes RB problem, relying either on Galerkin or Petrov-Galerkin projection, are available. The following subsections are devoted to their description.

### 3.2.1 Galerkin-RB method with velocity enrichment

A Galerkin-RB formulation is obtained by choosing $W_N^{\boldsymbol{\mu}} = V_N$ (or algebraically $\mathbf{W}^{\boldsymbol{\mu}} = \mathbf{V}$) in (15), resulting in a RB approximation whose well-posedness is guaranteed by satisfying the following assumption: there must exist $\tilde{\beta}_N^{min} > 0$ such that

$$\tilde{\beta}_N^{\boldsymbol{\mu}} = \inf_{\mathbf{p}_{N_p} \in \mathbb{R}^{N_p}} \sup_{\mathbf{v}_{N_u} \mathbb{R}^{N_p}} \frac{\mathbf{u}_{N_u}^T \mathbf{V}_{N_u}^T (\mathbf{B}_h^{\boldsymbol{\mu}})^T \mathbf{V}_{N_p} \mathbf{p}_{N_p}}{\|\mathbf{V}_{N_u}\mathbf{u}_{N_u}\|_{\mathbf{X}_u} \|\mathbf{V}_{N_p}\mathbf{p}_{N_p}\|_{\mathbf{X}_p}} \geq \tilde{\beta}_N^{min} > 0 \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{17}$$

Unfortunately, as explained above, condition (17) is not automatically satisfied when the RB spaces $V_{N_u}$ and $Q_{N_p}$ are constructed by POD, or by greedy algorithms, by considering basis functions extracted from velocity and pressure snapshots only. Consequently, we consider an "enriched" velocity space formulation, as proposed in [4], where the velocity space $V_{N_u}$ is augmented to guarantee the well-posedness of the resulting RB approximation. This method has been proposed for both POD and greedy RB space construction, and, even if empirically it works properly, it does not rigorously ensure the well-posedness of the resulting RB problem. Furthermore, when using POD it is unclear how large the augmenting space should be.

Algebraically, this enriching strategy is pursued by building a matrix $\mathbf{V}_{N_s} \in \mathbb{R}^{N_h^u \times N_s}$ whose columns form a basis for the enriching RB velocity space.

Then, the GRB approximation is built by considering $\mathbf{V} = \mathbf{W}^{\boldsymbol{\mu}} = \tilde{\mathbf{V}}$ in (15), where

$$\tilde{\mathbf{V}} = \begin{bmatrix} \mathbf{V}_{N_u} & \mathbf{V}_{N_s} & 0 \\ 0 & 0 & \mathbf{V}_{N_p} \end{bmatrix}.$$

The enriching strategy is based upon the use of the pressure-supremizing operator $T_p^{\boldsymbol{\mu}} : Q_h^{\boldsymbol{\mu}} \to V_h^{\boldsymbol{\mu}}$. For any given $q_h \in Q_h^{\boldsymbol{\mu}}$, $T_p^{\boldsymbol{\mu}}(q_h)$ is the solution of the following problem

$$(T_p^{\boldsymbol{\mu}}(q_h), \vec{v}_h)_{V^{\boldsymbol{\mu}}} = b^{\boldsymbol{\mu}}(\vec{v}_h, q_h) \qquad \forall \vec{v}_h \in V_h^{\boldsymbol{\mu}}. \tag{18}$$

Equation (18) corresponds to a FE problem whose algebraic formulation yields the linear system

$$\mathbf{X}_u^{\boldsymbol{\mu}} \mathbf{t}_p^{\boldsymbol{\mu}}(\mathbf{q}_h) = (\mathbf{B}_h^{\boldsymbol{\mu}})^T \mathbf{q}_h, \tag{19}$$

where $\mathbf{q}_h \in \mathbb{R}^{N_h^p}$ is the FE vector representation of $q_h \in Q_h^{\boldsymbol{\mu}}$. Two strategies have been developed to build a well-posed GRB approximation for a new parameter $\boldsymbol{\mu}$:

- build for each pressure basis $\{\boldsymbol{\xi}_i\}_{i=N_u+1}^{N}$ the supremizing functions $\{\mathbf{t}_p^{\boldsymbol{\mu}}(\boldsymbol{\xi}_i)\}_{i=N_u+1}^{N}$ and define

$$\mathbf{V}_{N_s} = [\mathbf{t}_p^{\boldsymbol{\mu}}(\boldsymbol{\xi}_{N_u+1})|\dots|\mathbf{t}_p^{\boldsymbol{\mu}}(\boldsymbol{\xi}_{N_u+1})],$$

  leading to a RB formulation which by definition satisfies (17). However, in this way the construction of the supremizing enriching functions is not computationally feasible, because it entails (online) the solution of $N_p$ FE linear system for each new value of $\boldsymbol{\mu}$;

- compute a set of supremizing snapshots $\left\{\mathbf{t}_p^{\boldsymbol{\mu}_i}(\mathbf{p}_h^{\boldsymbol{\mu}_i})\right\}_{i=1}^{n_s}$ corresponding to the pressure snapshots $\left\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\right\}_{i=1}^{n_s}$ by solving (19) $n_s$ times, and then build the matrix $\mathbf{V}_{N_s}$ through POD

$$\mathbf{V}_{N_s} = \mathrm{POD}\Big(\left\{\mathbf{t}_p^{\boldsymbol{\mu}_i}(\mathbf{p}_h^{\boldsymbol{\mu}_i})\right\}_{i=1}^{n_s}, \mathbf{X}_u, \delta_{\mathrm{POD}}\Big).$$

  This option does not ensure that condition (17) (or any equivalent one) is satisfied. Moreover, the number $N_s$ of basis functions for $\mathbf{V}_{N_s}$ is chosen, with a rule of thumb, equal to $N_u$, doubling the size of the RB velocity space. This looks like a reliable option which yields a stable RB problem for the steady Navier-Stokes equations, see [4].

### 3.2.2 LSRB method

Instead of performing a Galerkin projection onto properly enriched RB spaces, the Petrov-Galerkin (PG)RB method uses a different test space $\mathbf{W}^{\boldsymbol{\mu}}$ and naturally builds an *inf-sup* stable RB problem. The PGRB method has been firstly analyzed for the affinely parametrized Stokes equations in [1] where the RB space is built upon a greedy algorithm. In this work we deepen the analysis carried out in [1], propose several strategies to make this method computationally efficient and use instead the POD method for the construction of the RB space. Moreover, we do not assume to have an analytical function which maps the reference domain $\Omega^0$ to the physical domain $\Omega^{\boldsymbol{\mu}}$; the main consequence is that we consider the more general case where recasting the problem on a reference, parameter-independent domain $\Omega^0$ is not possible. We restrict ourselves to the case of PGRB method built through the least-squares (LS) method, which automatically guarantees to obtain an *inf-sup* stable problem. With this aim, we introduce a global supremizing operator $T^{\boldsymbol{\mu}} : X_h^{\boldsymbol{\mu}} \to X_h^{\boldsymbol{\mu}}$, such that

$$(T^{\boldsymbol{\mu}}(\vec{z}_h), \vec{w}_h)_{X^{\boldsymbol{\mu}}} = \mathcal{A}^{\boldsymbol{\mu}}(\vec{z}_h, \vec{w}_h) \qquad \forall \vec{w}_h \in X_h^{\boldsymbol{\mu}}. \tag{20}$$

With respect to the definition (18) of $T_p^{\boldsymbol{\mu}}$, both velocity and pressure appearing in (20), together with the full Stokes operator at the right hand side. Given $\vec{z}_h \in X_h^{\boldsymbol{\mu}}$, problem (20) is a $\boldsymbol{\mu}-$dependent FE problem which needs to be solved to determine $T^{\boldsymbol{\mu}}(\vec{z}_h)$. Then, the LSRB problem reads as (13), where the test space is chosen as

$$W_N^{\boldsymbol{\mu}} = span\big\{T^{\boldsymbol{\mu}}(\vec{\xi}_i),\ i = 1, \dots, N\big\},$$

while the trial RB space is chosen as in (12) with the corresponding matrix $\mathbf{V}$ as in (16). From an algebraic standpoint, given $\mathbf{z}_h \in \mathbb{R}^{N_h}$, the supremizing function $\mathbf{t}^{\mu}(\mathbf{z}_h)$ is obtained by solving the linear system

$$\mathbf{X}_h^{\boldsymbol{\mu}}\mathbf{t}^{\mu}(\mathbf{z}_h) = \mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{z}_h. \tag{21}$$

The projection matrix $\mathbf{W}^{\boldsymbol{\mu}}$, whose columns are supremizers of type (21) and form a basis for the ($\boldsymbol{\mu}$-dependent) test space, is then given by

$$\mathbf{W}^{\boldsymbol{\mu}} = (\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}, \tag{22}$$

where $\mathbf{X}_h^{\boldsymbol{\mu}}$ is the $\boldsymbol{\mu}-$dependent norm matrix (8). Finally, the linear system (14) representing the LSRB problem is recovered with

$$\mathbf{A}_N^{\boldsymbol{\mu}} = \mathbf{V}^T(\mathbf{A}_h^{\boldsymbol{\mu}})^T(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V} \qquad \mathbf{g}_N^{\boldsymbol{\mu}} = \mathbf{V}^T(\mathbf{A}_h^{\boldsymbol{\mu}})^T(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}\mathbf{g}_h^{\boldsymbol{\mu}}. \tag{23}$$

The following results hold, see also [1, 32] for the proof.

**Lemma 3.1.** *Assume that condition* (9) *holds and* $\mathbf{W}^{\boldsymbol{\mu}}$ *is defined as in* (22)*. Then, the LSRB problem* (14) *is* $\boldsymbol{\mu}-$*uniformly inf-sup stable, i.e., there exists* $\beta^{min} > 0$ *independent of* $\boldsymbol{\mu}$ *such that*

$$\beta_N^{\boldsymbol{\mu}} = \inf_{\mathbf{z}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T \mathbf{A}_N^{\boldsymbol{\mu}} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \|\mathbf{W}^{\boldsymbol{\mu}}\mathbf{w}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \beta^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}.$$

*Moreover, it has a unique solution* $\mathbf{z}_N^{\boldsymbol{\mu}} \in \mathbb{R}^N$ *for any* $\boldsymbol{\mu} \in \mathcal{D}$*, which satisfies*

$$\|\mathbf{z}_N^{\boldsymbol{\mu}}\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \leq \frac{1}{\beta_N^{\boldsymbol{\mu}}} \|\mathbf{g}_h^{\boldsymbol{\mu}}\|_{(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}}.$$

**Remark 3.1.** *The solution* $\mathbf{z}_N^{\boldsymbol{\mu}} \in \mathbb{R}^N$ *of problem* (14) *solves the following minimization problem*

$$\mathbf{z}_N^{\boldsymbol{\mu}} = \arg \min_{\mathbf{v}_N \in \mathbb{R}^N} \|\mathbf{g}_h^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{V} \mathbf{v}_N\|^2_{(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}}, \tag{24}$$

*that is, the RB solution minimizes the square of the residual in the norm induced by the symmetric positive definite matrix* $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$*, see [32] for further details.*

# 4 Algebraic LSRB method for the parametrized Stokes equations

The LSRB method described in Section 3.2.2 requires to build the $\boldsymbol{\mu}$-dependent matrix $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$ or to solve approximately the $N$ linear systems (21) associated with the matrix $\mathbf{X}_h^{\boldsymbol{\mu}}$ to construct a stable RB problem for any new parameter instances $\boldsymbol{\mu} \in \mathcal{D}$ considered online. In this section we propose a purely algebraic PGRB method which can be viewed as an algebraic LSRB (aLSRB) method for problem (6). Compared to the approximate enrichment of the velocity space described in section (3.2.1), the aLSRB method allows to build a RB problem which is automatically and rigorously *inf-sup* stable and henceforth it does not require to enrich the velocity space doubling the degrees of freedom of the velocity.

## 4.1 Construction of algebraic LSRB method

The underlying idea to build the aLSRB approximation is to substitute the matrix $\mathbf{X}_h^{\boldsymbol{\mu}}$ appearing in the definition of the test space (22) by a properly chosen surrogate $\mathbf{P}_X \in \mathbb{R}^{N_h \times N_h}$. To this aim, we suppose the following assumption to hold.

**Assumption 4.1.** *The matrix* $\mathbf{P}_X \in \mathbb{R}^{N_h \times N_h}$ *is symmetric and positive definite and induces a norm* $\|\mathbf{x}\|^2_{\mathbf{P}_X} = (\mathbf{x}, \mathbf{x})_{\mathbf{P}_X} = \mathbf{x}^T \mathbf{P}_X \mathbf{x}$ *for any* $\mathbf{x} \in \mathbb{R}^{N_h}$*. Moreover, there exist two positive constants* $c$ *and* $C$ *independent of* $\boldsymbol{\mu}$ *such that*

$$c\|\mathbf{x}\|_{\mathbf{P}_X} \leq \|\mathbf{x}\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \leq C\|\mathbf{x}\|_{\mathbf{P}_X} \qquad \forall \mathbf{x} \in \mathbb{R}^{N_h}. \tag{25}$$

Next, we introduce a slightly modified supremizing operator $T_{\mathbf{P}_X}^{\boldsymbol{\mu}} : V_h^{\boldsymbol{\mu}} \times V_h^{\boldsymbol{\mu}} \to V_h^{\boldsymbol{\mu}}$ defined by the following problem

$$(T_{\mathbf{P}_X}^{\boldsymbol{\mu}}(\vec{z}_h), \vec{w}_h)_{\mathbf{P}_X} = \mathcal{A}^{\boldsymbol{\mu}}(\vec{z}_h, \vec{w}_h) \qquad \forall \vec{w}_h \in V_h^{\boldsymbol{\mu}}, \tag{26}$$

where the difference with respect to (20) is the choice of the scalar product $(\cdot,\cdot)_{\mathbf{P}_X}$ with respect to which the operator is built. Reasoning as in the previous section, we introduce a PG problem under the form: find $\vec{z}_N \in V_N$ such that

$$\mathcal{A}^{\boldsymbol{\mu}}(\vec{z}_N, \vec{w}_N) = \mathcal{F}^{\boldsymbol{\mu}}(\vec{w}_N) \qquad \forall \vec{w}_N \in W_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}, \tag{27}$$

where now the test space is chosen as

$$W_{N,\mathbf{P}_X}^{\boldsymbol{\mu}} = span\big\{T_{\mathbf{P}_X}^{\boldsymbol{\mu}}(\vec{\xi}_i),\ i = 1, \ldots, N\big\},$$

with $\{\vec{\xi}_i\}_{i=1}^N$ denoting the RB functions of the space $V_N$ defined in (12). Problem (26) is algebraically equivalent to solving

$$\mathbf{P}_X \mathbf{t}^{\mu}_{\mathbf{P}_X}(\mathbf{z}_h) = \mathbf{A}_h \mathbf{z}_h, \tag{28}$$

and yields a projection matrix of the following form

$$\mathbf{W}^{\boldsymbol{\mu}}_{\mathbf{P}_X} = \mathbf{P}_X^{-1} \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{V}. \tag{29}$$

Finally, the corresponding RB system is

$$\mathbf{A}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} \mathbf{z}^{\boldsymbol{\mu}}_N = \mathbf{g}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X}, \tag{30}$$

where the RB matrix $\mathbf{A}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} \in \mathbb{R}^{N\times N}$ and the RB right hand side $\mathbf{g}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} \in \mathbb{R}^N$ are defined as

$$\mathbf{A}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} = \mathbf{V}^T (\mathbf{A}^{\boldsymbol{\mu}}_h)^T \mathbf{P}_X^{-1} \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{V} \qquad \mathbf{g}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} = \mathbf{V}^T (\mathbf{A}^{\boldsymbol{\mu}}_h)^T \mathbf{P}_X^{-1} \mathbf{g}^{\boldsymbol{\mu}}_h. \tag{31}$$

**Remark 4.1.** *Equations* (31) *are similar to the ones in* (23)*, provided* $\mathbf{X}^{\boldsymbol{\mu}}_h$ *is substituted with* $\mathbf{P}_X$*.*

## 4.2 Stability of algebraic LSRB method

In the following we provide results showing the stability of system (30) and the optimality properties satisfied by the solution $\mathbf{z}^{\boldsymbol{\mu}}_N$ of (30).

**Proposition 4.1.** *Assume that condition* (9) *holds,* $\mathbf{W}^{\boldsymbol{\mu}}$ *is taken as in* (29)*, and let assumption 4.1 hold. Then problem* (30) *is inf-sup stable, more precisely*

$$\beta^{\boldsymbol{\mu}}_{\mathbf{P}_X,N} = \inf_{\mathbf{z}_N\in\mathbb{R}^N} \sup_{\mathbf{w}_N\in\mathbb{R}^N} \frac{\mathbf{w}_N^T \mathbf{A}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \|\mathbf{W}^{\boldsymbol{\mu}}_{\mathbf{P}_X} \mathbf{w}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h}} \geq \frac{c}{C} \beta^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{32}$$

*Moreover, problem* (30) *has a unique solution* $\mathbf{z}^{\boldsymbol{\mu}}_N \in \mathbb{R}^N$ *for any* $\boldsymbol{\mu} \in \mathcal{D}$*, which satisfies*

$$\|\mathbf{z}^{\boldsymbol{\mu}}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \leq \frac{1}{\beta^{\boldsymbol{\mu}}_{\mathbf{P}_X,N}} \|\mathbf{g}^{\boldsymbol{\mu}}_h\|_{(\mathbf{X}^{\boldsymbol{\mu}}_h)^{-1}}.$$

*Proof.* Starting from (28), it holds

$$\mathbf{w}_h^T \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{z}_h = \mathbf{w}_h^T \mathbf{P}_X \mathbf{t}^{\mu}_{\mathbf{P}_X}(\mathbf{z}_h) \leq \|\mathbf{t}^{\mu}_{\mathbf{P}_X}(\mathbf{z}_h)\|_{\mathbf{P}_X} \|\mathbf{w}_h\|_{\mathbf{P}_X} \qquad \forall \mathbf{w}_h \in \mathbb{R}^{N_h},$$

where the equality is reached for $\mathbf{w}_h = \mathbf{t}^{\mu}_{\mathbf{P}_X}(\mathbf{z}_h)$. Consequently, using (25) we have

$$\begin{aligned}
\beta^{\boldsymbol{\mu}}_{\mathbf{P}_X,N} &= \inf_{\mathbf{z}_N\in\mathbb{R}^N} \sup_{\mathbf{w}_N\in\mathbb{R}^N} \frac{\mathbf{w}_N^T \mathbf{A}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \|\mathbf{W}^{\boldsymbol{\mu}}_{\mathbf{P}_X} \mathbf{w}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h}} \geq \frac{1}{C} \inf_{\mathbf{z}_N\in\mathbb{R}^N} \sup_{\mathbf{w}_N\in\mathbb{R}^N} \frac{\mathbf{w}_N^T \mathbf{A}^{\boldsymbol{\mu}}_{N,\mathbf{P}_X} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \|\mathbf{W}^{\boldsymbol{\mu}}_{\mathbf{P}_X} \mathbf{w}_N\|_{\mathbf{P}_X}} \\
&= \frac{1}{C} \inf_{\mathbf{z}_N\in\mathbb{R}^N} \frac{\|\mathbf{t}^{\mu}_{\mathbf{P}_X}(\mathbf{V}\mathbf{z}_N)\|_{\mathbf{P}_X}}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}^{\boldsymbol{\mu}}_h}} \geq \frac{1}{C} \inf_{\mathbf{z}_h\in\mathbb{R}^{N_h}} \frac{\|\mathbf{t}^{\mu}_{\mathbf{P}_X}(\mathbf{z}_h)\|_{\mathbf{P}_X}}{\|\mathbf{z}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_h}} = \frac{1}{C} \inf_{\mathbf{z}_h\in\mathbb{R}^{N_h}} \sup_{\mathbf{w}_h\in\mathbb{R}^{N_h}} \frac{\mathbf{w}_h^T \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{z}_h}{\|\mathbf{z}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \|\mathbf{w}_h\|_{\mathbf{P}_X}} \\
&\geq \frac{c}{C} \inf_{\mathbf{z}_h\in\mathbb{R}^{N_h}} \sup_{\mathbf{w}_h\in\mathbb{R}^{N_h}} \frac{\mathbf{w}_h^T \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{z}_h}{\|\mathbf{z}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \|\mathbf{w}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_h}} = \frac{c}{C} \beta^{\boldsymbol{\mu}}_h \geq \frac{c}{C} \beta^{min}.
\end{aligned}$$

Applying the Babuška theorem for non-coercive problems satisfying an *inf-sup* stability property, see [2], allows to conclude the proof. □

**Proposition 4.2.** *Let assumption 4.1 hold, then problem* (30) *corresponds to solving the minimization problem*

$$\mathbf{z}^{\boldsymbol{\mu}}_N = \arg\min_{\mathbf{v}_N\in\mathbb{R}^N} \|\mathbf{g}^{\boldsymbol{\mu}}_h - \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{V}\mathbf{v}_N\|^2_{\mathbf{P}_X^{-1}}. \tag{33}$$

*Proof.* We consider the quadratic functional

$$J(\mathbf{v}_N) = \|\mathbf{g}_h^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{v}_N\|^2_{\mathbf{P}_X^{-1}}, \quad \mathbf{v}_N \in \mathbb{R}^N,$$

which has a unique minimum in $\mathbf{u}_N \in \mathbb{R}^N$ thanks to the nonsingularity of the matrices $\mathbf{P}_X$ and $\mathbf{A}_h^{\boldsymbol{\mu}}$. We impose its gradient with respect to $\mathbf{v}_N$, and evaluated at $\mathbf{u}_N$, to vanish. By employing the definition of the norm $\|\cdot\|_{\mathbf{P}_X^{-1}}$ we obtain

$$\begin{aligned}\mathbf{0} &= \frac{\partial J\{\mathbf{v}_N\}}{\partial \mathbf{v}_N}(\mathbf{u}_N) = \frac{\partial}{\partial \mathbf{v}_N}\Big\{(\mathbf{g}_h^{\boldsymbol{\mu}})^T\mathbf{P}_X^{-1}\mathbf{g}_h^{\boldsymbol{\mu}} + \mathbf{v}_N^T\mathbf{V}^T(\mathbf{A}_h^{\boldsymbol{\mu}})^T\mathbf{P}_X^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{v}_N - 2(\mathbf{g}_h^{\boldsymbol{\mu}})^T\mathbf{P}_X^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{v}_N\Big\}(\mathbf{u}_N)\\ &= 2\mathbf{V}^T(\mathbf{A}_h^{\boldsymbol{\mu}})^T\mathbf{P}_X^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{u}_N - 2(\mathbf{g}_h^{\boldsymbol{\mu}})^T\mathbf{P}_X^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{u}_N = 2\mathbf{A}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}\mathbf{u}_N - 2\mathbf{g}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}.\end{aligned}$$

Therefore, $\mathbf{u}_N$ is such that

$$\mathbf{A}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}\mathbf{u}_N = \mathbf{g}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}},$$

hence it coincides with the RB solution $\mathbf{z}_N^{\boldsymbol{\mu}}$, since the matrix $\mathbf{A}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}$ is invertible. □

**Remark 4.2.** *As a matter of fact, compared to the LSRB methods in [1, 32], the main feature of aLSRB methods consists in employing a parameter-independent metric in* (33)*, which is obtained by substituting the matrix* $\mathbf{X}_h^{\boldsymbol{\mu}}$ *by the* $\boldsymbol{\mu}$*-independent* $\mathbf{P}_X$ *in the minimization criterion for the residual.*

## 4.3 On the choice of $\mathbf{P}_X$

A natural question arising in this context regards the choice of the matrix $\mathbf{P}_X$, since this directly affects the values of the constants $c$ and $C$; (see (32)). These constants play indeed a relevant role in the conditioning of the aLSRB approximation. Moreover, it is clear that by taking $\mathbf{P}_X = \mathbf{X}_h^{\boldsymbol{\mu}}$, we would have the optimal case $c/C = 1$, hence recovering the standard LSRB method. Therefore, $\mathbf{P}_X$ should be chosen as close as possible to $\mathbf{X}_h^{\boldsymbol{\mu}}$, however it has to be $\boldsymbol{\mu}$-independent. The following results give some insights on how to properly choose the matrix $\mathbf{P}_X$. Their proofs are reported in the Appendix.

**Lemma 4.1.** *Let assumption 4.1 hold. The optimal value for the constants* $C \geq c$ *satisfying* (25) *are*

$$C = \|\mathbf{P}_X^{-1/2}(\mathbf{X}_h^{\boldsymbol{\mu}})^{1/2}\|_{\mathbf{P}_X}, \qquad c = 1/\|(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1/2}\mathbf{P}_X^{1/2}\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}. \tag{34}$$

From now, we consider $C$, $c$ as their optimal values (34).

**Lemma 4.2.** *Let assumption 4.1 hold. The two constants* $C \geq c > 0$ *satisfying* (25) *and* (34) *are such that*

$$\frac{c}{C} = \Big[\mathcal{K}_{\mathbf{X}_h}(\mathbf{P}_X^{-1}\mathbf{X}_h^{\boldsymbol{\mu}})\Big]^{-1/2} = \Big[\mathcal{K}_2(\mathbf{P}_X^{-1/2}\mathbf{X}_h\mathbf{P}_X^{-1/2})\Big]^{-1/2}. \tag{35}$$

It is clear from Lemma 4.2 that the matrix $\mathbf{P}_X$ should be chosen so that the condition number of the preconditioned matrix $\mathbf{P}_X^{-1}\mathbf{X}_h^{\boldsymbol{\mu}}$ does not depend on the mesh size $h$, that is, $\mathbf{P}_X$ should be an optimal preconditioner for $\mathbf{X}_h^{\boldsymbol{\mu}}$. If this is not the case, the value of the stability constant of the RB approximation $\beta_{\mathbf{P}_X,N}^{\boldsymbol{\mu}}$ may depend on $h$. Furthermore, if we set up our RB approximation in a HPC environment, employing a mesh partitioner to divide the computational domain among the processors, it is also advisable to choose $\mathbf{P}_X$ such that $\frac{c}{C}$ does not depend on the size $H$ of the subdomains, that is, $\mathbf{P}_X$ should be a scalable preconditioner for $\mathbf{X}_h^{\boldsymbol{\mu}}$.

In our numerical experiments, $\mathbf{P}_X$ is chosen either as $\mathbf{P}_X = \mathbf{X}_h^0$, that is, as the norm matrix in the reference domain, or as a block diagonal preconditioner of $\mathbf{X}_h^0$, where the two blocks are generated as symmetric and positive definite preconditioners $\mathbf{P}_{\mathbf{X}_u} \in \mathbb{R}^{N_h^u \times N_h^u}$ of $\mathbf{X}_u^0$ and $\mathbf{P}_{\mathbf{X}_p} \in \mathbb{R}^{N_h^p \times N_h^p}$ of $\mathbf{X}_u^0$ and $\mathbf{X}_p^0$, respectively.

### 4.4 Assembling the RB problem

When building a RB approximation, it is essential to assume the affine dependence on $\boldsymbol{\mu}$ in the FE arrays (6), that is, the possibility to express them as

$$\mathbf{A}_h^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_h^q, \qquad \mathbf{g}_h^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu}) \mathbf{g}_h^q, \tag{36}$$

where $\Theta_a^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_a$ and $\Theta_g^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_g$ are $\boldsymbol{\mu}$-dependent functions, while the matrices $\mathbf{A}_h^q \in \mathbb{R}^{N_h \times N_h}$ and the vectors $\mathbf{g}_h^q \in \mathbb{R}^{N_h}$ are $\boldsymbol{\mu}$-independent. If assumption (36) is verified, then the RB algebraic structures can be written, for the aLSRB case, as

$$\mathbf{A}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}} = \sum_{q_1,q_2=1}^{Q_a} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_a^{q_2}(\boldsymbol{\mu}) \mathbf{V}^T (\mathbf{A}_h^{q_1})^T \mathbf{P}_X^{-1} \mathbf{A}_h^{q_2} \mathbf{V} = \sum_{q_1,q_2=1}^{Q_a} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_a^{q_2}(\boldsymbol{\mu}) \mathbf{A}_N^{q_1,q_2} \tag{37}$$

$$\mathbf{g}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}} = \sum_{q_1=1}^{Q_a} \sum_{q_2=1}^{Q_g} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_g^{q}(\boldsymbol{\mu}) \mathbf{V}^T (\mathbf{A}_h^{q_1})^T \mathbf{P}_X^{-1} \mathbf{g}_h^{q_2} = \sum_{q_1=1}^{Q_a} \sum_{q_2=1}^{Q_g} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_g^{q_2}(\boldsymbol{\mu}) \mathbf{g}_N^{q_1,q_2}. \tag{38}$$

In the GRB case, the algebraic RB structures can be instead obtained as

$$\mathbf{A}_N^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{A}_h^q \mathbf{V} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_N^q \tag{39}$$

$$\mathbf{g}_N^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{g}_h^q = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu}) \mathbf{g}_N^q. \tag{40}$$

The matrices $\mathbf{A}_N^q$, $q = 1, \dots, Q_a$, $\mathbf{A}_N^{q_1,q_2} \in \mathbb{R}^{N \times N}$, $q_1, q_2 = 1, \dots, Q_a$, and the vectors $\mathbf{g}_N^q \in \mathbb{R}^N$, $q = 1, \dots, Q_g$, $\mathbf{g}_N^{q_1,q_2} \in \mathbb{R}^N$, $q_1 = 1, \dots, Q_a$, $q_2 = 1, \dots, Q_g$ can be precomputed and stored during the offline phase. During the online phase, only the sums in (37)–(38) and (39)–(40) must be calculated to assemble the RB problem.

Notice that the construction of $\mathbf{A}_N^{\boldsymbol{\mu}}$ and $\mathbf{g}_N^{\boldsymbol{\mu}}$ in (39)–(40) depends linearly on the number of affine terms $Q_a$ and $Q_g$ for the GRB method. On the other hand, the corresponding aLSRB structures $\mathbf{A}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}$ and $\mathbf{g}_{N,\mathbf{P}_X}^{\boldsymbol{\mu}}$ in (37)–(38) depend quadratically $Q_a$ and $Q_g$. Practically, by employing the GRB method softens the dependence on the number of affine terms, since less RB structures must be assembled and stored with respect to the aLSRB method. This advantage is also visible in the online phase, since the construction of (39)–(40) scale linearly with respect to $Q_a$ and $Q_g$. However, the aLSRB matrices and right hand sides have a smaller dimension, since the velocity basis is not augmented, entailing a lower cost for computing and storing each array and for computing the solution of the RB system. Finally, notice that the affine decomposition (36) would not be exploitable in the case of standard LSRB method, due to the $\boldsymbol{\mu}-$dependence of the matrix $\mathbf{X}_h^{\boldsymbol{\mu}}$. Indeed, one would need also an affine decomposition of $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$, which is generally not available since it is never explicitly assembled and its application is performed by solving a linear system where $\mathbf{X}_h^{\boldsymbol{\mu}}$ is at the left hand side.

In the numerical examples considered in this work, as well as in almost every problem of applied interest, the geometrical dependence of the computational domain on the parameter $\boldsymbol{\mu}$ is generally nonaffine, therefore an affine representation of $\mathbf{A}_h^{\boldsymbol{\mu}}$ and $\mathbf{g}_h^{\boldsymbol{\mu}}$ cannot be computed. To circumvent this problem, both the empirical interpolation method (EIM) or its discrete variants DEIM and Matrix-DEIM [5, 15, 28] offer the possibility to recover an *approximate* affine decomposition. When such techniques are employed, the relations (36) are satisfied up to a certain tolerance,

$$\mathbf{A}_h^{\boldsymbol{\mu}} \approx \sum_{q=1}^{Q_a} \widetilde{\Theta}_a^q(\boldsymbol{\mu}) \mathbf{A}_h^q, \qquad \mathbf{g}_h^{\boldsymbol{\mu}} \approx \sum_{q=1}^{Q_g} \widetilde{\Theta}_g^q(\boldsymbol{\mu}) \mathbf{g}_h^q. \tag{41}$$

$Q_a$ and $Q_g$ are the number of selected basis computed by the corresponding algorithms. In the case of DEIM (resp. MDEIM), the basis are again built by applying POD on a set of $n_s$ vector (resp. matrix) snapshots, where Algorithm 1 is employed with $\mathbf{X}$ the identity matrix and a tolerance $\delta_{\text{POD}} = \delta_{\text{deim}}$. Then, for a new value of the parameter $\boldsymbol{\mu}$, the coefficients $\widetilde{\Theta}_g^q : \mathcal{D} \to \mathbb{R}\, q = 1, \dots, Q_a$ (resp. $\widetilde{\Theta}_a^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_g$) are computed by solving an interpolation problem.

# 5 Coarse algebraic LSRB method for the parametrized Stokes equations

The cost of assembling the aLSRB problem (30) depends quadratically on the number of affine terms $Q_a$ for the RB matrix and $Q_g$ for the right hand side; this represents a serious bottleneck when a large number affine contributions are needed to affinely approximate the RB arrays. In order to overcome it, we introduce a *coarse* aLSRB (caLSRB) method, which allows to perform the construction of the RB problem by relying on $O(Q_a)$ (instead than $O(Q_a^2)$) terms. This entails the same complexity of a Galerkin RB method, however ending up with an intrinsically stable RB problem, without the need of enriching the velocity RB space.

To start with, we introduce a new supremizing operator: given $\mathbf{z}_h \in \mathbb{R}^{N_h}$, the new supremizing solution $\mathbf{t}^\mu(\mathbf{z}_h)$ is obtained by solving the linear system

$$\mathbf{P}_X \mathbf{t}^\mu(\mathbf{z}_h) = \widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \mathbf{z}_h \tag{42}$$

where $\mathbf{P}_X$ satisfies Assumption 4.1 and $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ can be regarded as a convenient approximation of $\mathbf{A}_h^{\boldsymbol{\mu}}$. Here we assume that $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ fulfills the following *inf-sup* condition.

**Assumption 5.1.** *There exists $\widetilde{\beta}^{min} > 0$ such that for any $\boldsymbol{\mu} \in \mathcal{D}$ is such that*

$$\widetilde{\beta}_h^{\boldsymbol{\mu}} = \inf_{\vec{z}_h \in \mathbb{R}^{N_h}} \sup_{\vec{w}_h \in \mathbb{R}^{N_h}} \frac{\mathbf{w}_h^T \widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \mathbf{z}_h}{\|\mathbf{z}_h\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \|\mathbf{w}_h\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \widetilde{\beta}^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{43}$$

The caLSRB approximation is obtained by selecting the test projection matrix $\mathbf{W}^{\boldsymbol{\mu}}$ as

$$\mathbf{W}^{\boldsymbol{\mu}} = \widetilde{\mathbf{W}}^{\boldsymbol{\mu}} = \mathbf{P}_X^{-1} \widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \mathbf{V}, \tag{44}$$

that is, the columns of $\widetilde{\mathbf{W}}^{\boldsymbol{\mu}}$ are supremizers fulfilling (42) by taking as $\mathbf{z}_h$ in this case each column of $\mathbf{V}$ – and form a basis for the ($\boldsymbol{\mu}$-dependent) test space. At algebraic level, the corresponding caLSRB formulation reads

$$\widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}} \widetilde{\mathbf{z}}_N^{\boldsymbol{\mu}} = \widetilde{\mathbf{g}}_N^{\boldsymbol{\mu}}, \tag{45}$$

where $\widetilde{\mathbf{z}}_N^{\boldsymbol{\mu}} \in \mathbb{R}^N$ is the caLSRB approximation and

$$\widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}} = (\widetilde{\mathbf{W}}^{\boldsymbol{\mu}})^T \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{V} = \mathbf{V}^T (\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}})^T \mathbf{P}_X^{-1} \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{V} \in \mathbb{R}^{N \times N}, \qquad \widetilde{\mathbf{g}}_N^{\boldsymbol{\mu}} = \mathbf{V}^T (\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}})^T \mathbf{P}_X^{-1} \mathbf{g}_h^{\boldsymbol{\mu}} \in \mathbb{R}^N.$$

The main difference with the aLSRB approximation (30) lies in the presence of $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$ in the definition of the RB matrix and right hand side. When employing either approximation (30) or (45), the RB problem is practically assembled by exploiting the approximated affine decompositions (41) of $\mathbf{A}_h^{\boldsymbol{\mu}}$ and $\mathbf{g}_h^{\boldsymbol{\mu}}$ as explained in Section 4.4. In the aLSRB approximation the assembling phase depends quadratically on the number of affine terms $Q_a$ and $Q_g$; contrarily, when employing the caLSRB formulation (45), the choice of the matrix $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$ is arbitrary. In the numerical experiments, $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$ will be selected as a *coarse* affine approximation of $\mathbf{A}_h^{\boldsymbol{\mu}}$, where only the first (and most relevant) mode of MDEIM is retained, that is,

$$\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} = \widetilde{\Theta}_a^1(\boldsymbol{\mu}) \mathbf{A}_h^1. \tag{46}$$

On one hand, this choice guarantees to efficiently assemble the RB problem (45), since the resulting assembling phase depends linearly on $Q_a$ for $\widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}}$ and $Q_g$ for $\widetilde{\mathbf{g}}_N^{\boldsymbol{\mu}}$. On the other hand, using a coarse affine approximation only for defining the test space does not affect the overall accuracy of the resulting method. Furthermore, we introduce the matrix

$$\bar{\mathbf{A}}_N^{\boldsymbol{\mu}} = (\widetilde{\mathbf{W}}^{\boldsymbol{\mu}})^T \widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \mathbf{V} = \mathbf{V}^T (\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}})^T \mathbf{P}_X^{-1} \widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}} \mathbf{V} \in \mathbb{R}^{N \times N},$$

which is obtained by algebraic least squares projection of the matrix $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$. The following result ensures the invertibility of $\bar{\mathbf{A}}_N^{\boldsymbol{\mu}}$.

**Proposition 5.1.** *Assume that condition* (43) *holds, then the following inf-sup condition holds*

$$\bar{\beta}_N^{\boldsymbol{\mu}} = \inf_{\mathbf{z}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T \bar{\mathbf{A}}_N^{\boldsymbol{\mu}} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \|\widetilde{\mathbf{W}}^{\boldsymbol{\mu}} \mathbf{w}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \frac{c}{C} \widetilde{\beta}^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}.$$

*Proof.* The proof is carried out as the one for Proposition 4.1, by properly replacing $\mathbf{A}_h^{\boldsymbol{\mu}}$ with $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$ and $\beta^{min}$ with $\widetilde{\beta}^{min}$. □

**Remark 5.1.** *We remark that we cannot prove a priori the invertibility of the matrix $\widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}}$, however this property is observed experimentally. Furthermore, if $\widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}}$ is assumed to be invertible, then the following inf-sup stability holds*

$$\widetilde{\beta}_N^{\boldsymbol{\mu}} = \inf_{\mathbf{z}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T \widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \|\widetilde{\mathbf{W}}^{\boldsymbol{\mu}} \mathbf{w}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \frac{c}{C} C_A \beta^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}.$$

*Indeed, we can evaluate the stability factor of the caLSRB problem* (45) *as*

$$\begin{aligned}
\widetilde{\beta}_N^{\boldsymbol{\mu}} &= \inf_{\mathbf{z}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T \widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h} \|\widetilde{\mathbf{W}}^{\boldsymbol{\mu}} \mathbf{w}_N\|_{\mathbf{X}_h}} \\
&\geq \inf_{\mathbf{y}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T \bar{\mathbf{A}}_N^{\boldsymbol{\mu}} \mathbf{y}_N}{\|\mathbf{V}\mathbf{y}_N\|_{\mathbf{X}_h} \|\widetilde{\mathbf{W}}^{\boldsymbol{\mu}} \mathbf{w}_N\|_{\mathbf{X}_h}} \inf_{\mathbf{z}_N \in \mathbb{R}^N} \frac{\|\mathbf{V}(\bar{\mathbf{A}}_N^{\boldsymbol{\mu}})^{-1} \widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}} \mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \frac{c}{C} C_A \widetilde{\beta}^{min}
\end{aligned}$$

*where*

$$C_A = \inf_{\mathbf{z}_N \in \mathbb{R}^N} \frac{\|\mathbf{V}(\bar{\mathbf{A}}_N^{\boldsymbol{\mu}})^{-1} \widetilde{\mathbf{A}}_N^{\boldsymbol{\mu}} \mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} > 0.$$

*Note that $C_A = 1$ in the case where $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$ is used in place of its approximation $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$; as a matter of fact, $C_A$ is as close to 1 as $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$ is close to $\mathbf{A}_h^{\boldsymbol{\mu}}$.*

# 6 Numerical experiments

We show the results obtained with the RB methods presented in Section 3, 4 and 5 implemented within the LifeV library[2], see [9]. We compare the GRB method (with velocity space enrichment) and the aLSRB method in the case of large-scale Stokes flows in a cylindrical domain which is nonaffinely parametrized. The deformation is not analitically known, since it is retrieved as the solution of an additional FE problem which harmonically extends in the interior of the domain the datum prescribed on a Dirichlet boundary.

## 6.1 Test case setting: Stokes problem in a parametrized cylinder

We consider the Stokes equations in a parameter dependent domain $\Omega^{\boldsymbol{\mu}} \subset \mathbb{R}^3$, which is obtained by deforming a reference domain

$$\Omega^0 = \{\vec{x} \in \mathbb{R}^3 : \ x_1^2 + x_2^2 < 0.25, \, x_3 \in (0, 5)\}$$

by means of a displacement $\vec{d}^{\boldsymbol{\mu}}$ obtained as the harmonic extension of a boundary deformation. More specifically, we set

$$\Omega^{\boldsymbol{\mu}} = \{\vec{x}^{\boldsymbol{\mu}} \in \mathbb{R}^3 : \ \vec{x}^{\boldsymbol{\mu}} = \vec{x} + \vec{d}^{\boldsymbol{\mu}}\},$$

where $\vec{d}^{\boldsymbol{\mu}}$ solves the following PDE

[2] www.lifev.org

$$\begin{cases} -\Delta \vec{d}^{\boldsymbol{\mu}} = \vec{0} & \text{in } \Omega^0 \\ \vec{d}^{\boldsymbol{\mu}} = \vec{h}^{\boldsymbol{\mu}} & \text{on } \partial\Omega^0. \end{cases} \tag{47}$$

In our numerical experiments we take $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D} = [-0.5, 0.5] \times [1.8, 3.2]$ and a Dirichlet datum of the form

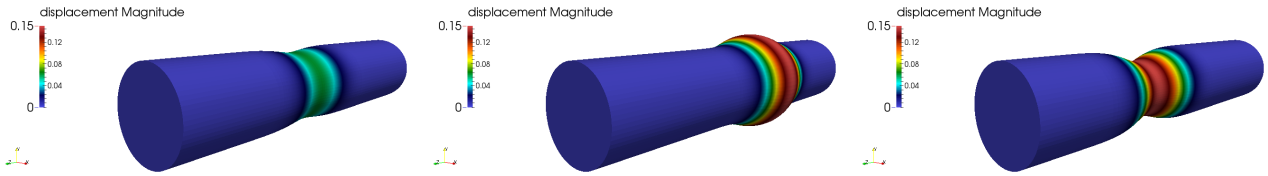
$$\vec{h}^{\boldsymbol{\mu}} = \begin{bmatrix} -x_1 \mu_1 \exp\{-5(x_3 - \mu_2)^2\} \\ -x_2 \mu_1 \exp\{-5(x_3 - \mu_2)^2\} \\ 0 \end{bmatrix},$$

entailing a deformation of the cylinder by narrowing or enlarging (according to the sign of $\mu_2$) its section in different positions along the coordinate $x_3$ (according to the value of $\mu_1$). Notice that the solution $\vec{d}^{\boldsymbol{\mu}}$ of (47) is not known a-priori, therefore we compute its numerical approximation $\vec{d}_h^{\boldsymbol{\mu}}$ by writing the variational form of problem (47) and by employing the FE method. We denote by $\mathbf{d}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^d}$ the solution of the corresponding FE linear system.

Moreover, once the computational domain has been deformed, the lifting function $\vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}}$ is computed similarly by solving the following problem

$$\begin{cases} -\Delta \vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}} = \vec{0} & \text{in } \Omega^{\boldsymbol{\mu}} \\ \vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}} = \vec{g}_D^{\boldsymbol{\mu}} & \text{on } \Gamma_{in}^{\boldsymbol{\mu}} \\ \vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}} = \vec{0} & \text{on } \Gamma_{w}^{\boldsymbol{\mu}} \\ \dfrac{\partial \vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}}}{\partial \vec{n}^{\boldsymbol{\mu}}} = \vec{0} & \text{on } \Gamma_{out}^{\boldsymbol{\mu}}, \end{cases} \tag{48}$$

which is an harmonic extension of the Dirichlet data in (2). Here $\vec{g}_D^{\boldsymbol{\mu}}$ is a parabolic profile such that the flow rate at the inlet is equal to 1. The FE method with second order polynomials ($\mathbb{P}^2$) basis functions is employed to approximate the solution of problem (48); this leads to a parametrized linear system whose solution $\mathbf{r}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^u}$ is the approximated lifting functions. In Fig. 1, the deformation $\mathbf{d}_h^{\boldsymbol{\mu}}$ is reported for three different values of $\boldsymbol{\mu} \in \mathcal{D}$. In the numerical experiments we




(a) $\boldsymbol{\mu} = (2.7, 0.12)$ (b) $\boldsymbol{\mu} = (2, -0.3)$ (c) $\boldsymbol{\mu} = (3, 0.3)$

Figure 1: Displacement for different values of $\boldsymbol{\mu}$.

present, Taylor-Hood FE ($\mathbb{P}^2 - \mathbb{P}^1$), with a mesh leading to $N_h = N_h^u + N_h^p = 1'503'280 + 64'943 = 1'568'223$ degrees of freedom, are employed for the Stokes equations. The algebraic problem is run on the Piz-Daint cluster with Cray XC40 machines, at the Swiss National Supercomputing Center (CSCS) in Lugano. The computation has been carried out with 256 processors.

### 6.1.1 FE simulation setup

For any parameter $\boldsymbol{\mu}$ considered, we solve the FE problems to approximate the deformation $\vec{d}^{\boldsymbol{\mu}}$ of problem (47) and the lifting function $\vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}}$ of problem (48). Next, we employ a move-mesh tool to shape the computational domain and assemble the FE Stokes arrays. This ensures that the meshes for different instances of the parameter $\boldsymbol{\mu}$ are topologically equivalent and there is a one-to-one correspondence between degrees of freedom. The FE linear system (6) is solved with the preconditioned flexible GMRES (FGMRES) method, where the preconditioner is the Pressure Mass Matrix (PMM) operator, which exploits the block structure of (7) and employs the mass matrix in pressure to approximate the Schur complement, see [30]. Finally, in order to compute the FE solution with the FGMRES method, up to a final tolerance of $10^{-8}$, our solver requires on average of 48.2 seconds, which also accounts for the time for deforming the domain, building the lifting function, the PMM preconditioner and the FE solution.

### 6.1.2 RB simulation setup

During the *offline* phase, we explore the parameter domain $\mathcal{D}$ for building our RB approximation. In particular we perform the following steps:

- we randomly choose a set of $n_s$ parameters $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s} \subset \mathcal{D}$; then we compute the corresponding velocity snapshots $\{\mathbf{u}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and pressure snapshots $\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ by solving the linear system (6). Next, we build the RB space $V_N$ by separately computing a basis $\mathbf{V}_{N_u}$ for the velocity and $\mathbf{V}_{N_p}$ the pressure, by plugging in the POD the same tolerance $\delta_{\text{POD}} = \delta_{RB}$ in both cases. If the GRB method with velocity enrichment is employed, we also compute $n_s$ supremizer snapshots $\{\mathbf{t}_p^{\boldsymbol{\mu}_i}(\mathbf{p}_h^{\boldsymbol{\mu}_i})\}_{i=1}^{n_s}$. Since in general we do not take the same number of basis functions for the velocity and pressure RB spaces, we use a tolerance also for computing the pressure supremizer basis functions. With this aim, we employ POD with $\delta_{\text{POD}} = \frac{\delta_{RB}}{10}$ to build the supremizer basis $\mathbf{V}_{N_s}$, which numerically confirmed to provide a stable GRB problem. The number of snapshots $n_s$ for the construction of the RB spaces has to be chosen in order to properly sample the parameter space $\mathcal{D}$ and construct a basis which provides a uniform accuracy across $\mathcal{D}$. In Fig. 2 the singular value decay for different values of $n_s$ for velocity (left) and pressure (right). By changing $n_s$, the decay approaches the one obtained with $n_s = 500$. Furthermore, in Table 1 the number of RB functions selected by POD is reported as function of $n_s$ when using a tolerance $\delta_{RB} = \delta_{\text{POD}} = 10^{-6}$ in Algorithm 1. As a matter of fact, the number of selected RB functions stabilizes as $n_s$ approaches $n_s = 500$, which is then considered a reasonable sample size for properly sampling the parameter space, yet avoiding a too heavy offline computation.
- we compute a basis to affinely approximate $\mathbf{f}_h^{\boldsymbol{\mu}}$, $\mathbf{r}_h^{\boldsymbol{\mu}}$ (by DEIM) and $\mathbf{D}_h^{\boldsymbol{\mu}}$, $\mathbf{B}_h^{\boldsymbol{\mu}}$ (by MDEIM), by taking $n_s = 250$ snapshots for each of these quantities and a tolerance $\delta_{\text{deim}}$ for the POD embedded in the (M)DEIM computation for the sake of basis selection, cf. Section 4.4.

In the *online* phase, we consider a test set made by $N_{onl} = 100$ new parameter instances and solve the corresponding RB problem (by means of the GRB, aLSRB or caLSRB method). We remark the parameters chosen during the online phase are different from the ones belonging to the training set $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$. We perform an analysis of the GRB and (c)aLSRB methods with respect to the tolerances $\delta_{RB}$ (or equivalently the number of basis functions $N$) and $\delta_{\text{deim}}$, by choosing $\delta_{RB} = 10^{-l}$, $l = 3, 4, 5, 6$ and $\delta_{\text{deim}} = 10^{-5}, 10^{-7}$. We evaluate the accuracy of the RB solutions $\mathbf{z}_N^{\boldsymbol{\mu}}$ in terms of the rescaled RB residual

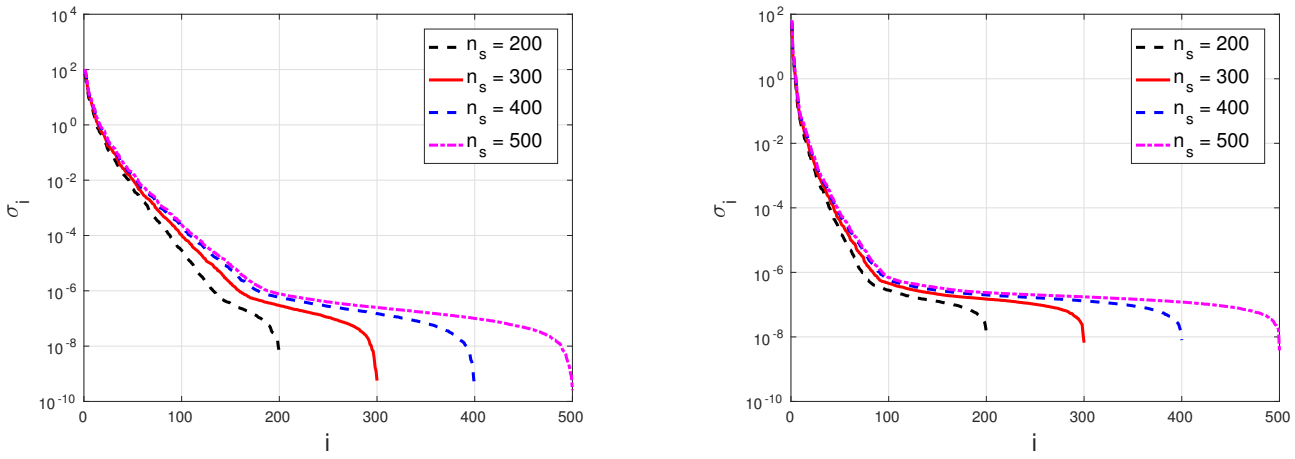
$$r_{RB} = \frac{\|\mathbf{g}_h^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{z}_N^{\boldsymbol{\mu}}\|_{(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}}}{\|\mathbf{g}_h^{\boldsymbol{\mu}}\|_{(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}}},$$

averaging the results obtained by solving the RB problem for the parameters of the test set. For the aLSRB and caLSRB methods, we present the results for the choice of the matrix $\mathbf{P}_X = \mathbf{P}_{\mathbf{X}_h^0}$, that is, we take the preconditioner $\mathbf{P}_{\mathbf{X}_h^0}$ of $\mathbf{X}_h^0$, which is the matrix norm on the reference domain $\Omega^0$. $\mathbf{P}_{\mathbf{X}_h^0}$ has a diagonal $2 \times 2$ block structure, where $\mathbf{P}_{\mathbf{X}_u} \in \mathbb{R}^{N_h^u \times N_h^u}$ (resp. $\mathbf{P}_{\mathbf{X}_p} \in \mathbb{R}^{N_h^p \times N_h^p}$) is a symmetric and positive definite AMG preconditioner of $\mathbf{X}_u^0$ (resp. $\mathbf{X}_p^0$), leading to a matrix $\mathbf{P}_X$ which does not depend on $\boldsymbol{\mu}$.

Table 1: Number of basis functions for velocity and pressure selected by POD with $\delta_{RB} = 10^{-6}$ as function of $n_s$.

| $n_s$ | 200 | 300 | 400 | 500 |
|---|---|---|---|---|
| $N_u$ | 96 | 110 | 119 | 122 |
| $N_p$ | 49 | 54 | 57 | 58 |

During the online phase, for any new $\boldsymbol{\mu}$, we solve the FE linear system for computing the deformation $\mathbf{d}_h^{\boldsymbol{\mu}}$ and the lifting function $\mathbf{r}_h^{\boldsymbol{\mu}}$. Alternatively, we could approximate both $\mathbf{d}_h^{\boldsymbol{\mu}}$ and $\mathbf{r}_h^{\boldsymbol{\mu}}$ by the RB method, similarly to what has been proposed in [25] on a simpler scalar problem. However, this goes beyond the scope of this paper and will be the subject of further research.




(a) Velocity (b) Pressure

Figure 2: Decay of singular values computed by performing POD on velocity and pressure snapshots.

## 6.2 Numerical results

### 6.2.1 Offline phase

In Table 2 we show the offline time and the corresponding number of affine terms $Q_a$ and $Q_g$ required to build an affine approximation of the FE matrices and right hand sides when using $\delta_{\text{deim}} = 10^{-7}$; these times are shared by both the GRB and aLSRB methods. In Table 3 the total computational time employed for the offline phase with $\delta_{RB} = 10^{-6}$ and $\delta_{\text{deim}} = 10^{-7}$ are reported, together with the details of its three main stages: (i) snapshots computation, (ii) POD and (iii) RB affine arrays construction. Comparable results hold when bigger tolerances are used, since the most part of the computational time is employed for computing the snapshots.

Table 2: Computational time (seconds) and number of affineterms to build (M)DEIM affine basis with $\delta_{\text{deim}} = 10^{-7}$.

| | **MDEIM - $\mathbf{D}_h^{\boldsymbol{\mu}}$** | **MDEIM - $\mathbf{B}_h^{\boldsymbol{\mu}}$** | **DEIM - $\mathbf{f}_h^{\boldsymbol{\mu}}$** | **DEIM - $\mathbf{r}_h^{\boldsymbol{\mu}}$** | **Total (M)DEIM** |
|---|---|---|---|---|---|
| **Offline time** | 998.9 | 549.2 | 721.8 | 721.5 | 2991.4 |
| **Affine terms** | 39 | 29 | 28 | 97 | 193 |

Table 3: Computational time (seconds) to build RB approximation with $\delta_{RB} = 10^{-6}$ and $\delta_{\text{deim}} = 10^{-7}$.

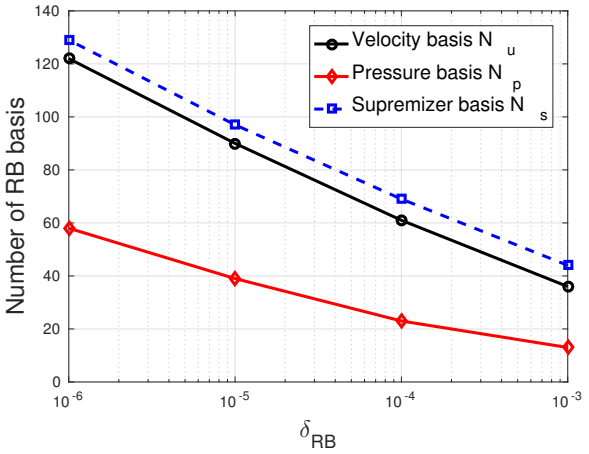
| | **GRB** | **aLSRB ($\mathbf{P}_{\mathbf{X}_h^0}$)** | **caLSRB ($\mathbf{P}_{\mathbf{X}_h^0}$)** |
|---|---|---|---|
| **Snapshots computation** | 24290.0 | 24105.0 | 24105.0 |
| **POD** | 25.6 | 16.4 | 16.4 |
| **Affine arrays construction** | 171.1 | 1997.7 | 77.9 |
| **Total (M)DEIM** | 2991.4 | 2991.4 | 2991.4 |
| **Total offline phase** | 27478.2 | 29110.5 | 27190.7 |

We recall that $\delta_{RB}$ is used within POD to build the velocity and pressure RB spaces, while $\delta_{\text{deim}}$ for building an affine approximation of the FE blocks of $\mathbf{A}_h^{\boldsymbol{\mu}}$ and $\mathbf{g}_h$ in the (M)DEIM algorithm. Snapshots computation is the most demanding phase, and is more expensive if the GRB method is employed, since it entails the additional computation of $n_s$ pressure supremizer snapshots $\left\{\mathbf{t}_p^{\boldsymbol{\mu}_i}(\mathbf{p}_h^{\boldsymbol{\mu}_i})\right\}_{i=1}^{n_s}$. The second phase, involving the POD to build the RB spaces, only requires a tiny percentage of the offline time for all the three methods considered, however also in this case the two variants of the aLSRB method need a shorter time than the GRB method, because they require only the construction of velocity and pressure spaces $\mathbf{V}_{N_u}$ and $\mathbf{V}_{N_p}$ – in the GRB case the

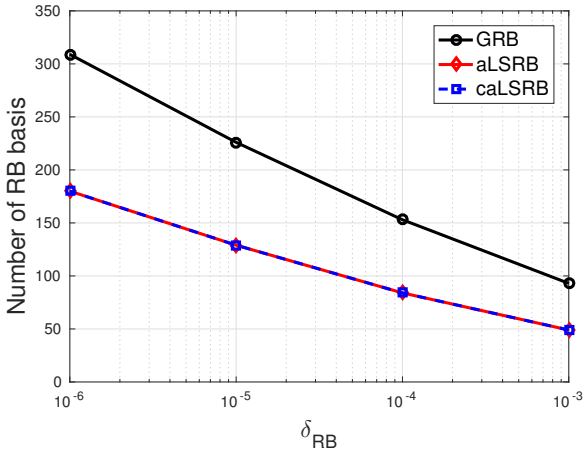
pressure supremizer space $\mathbf{V}_{N_s}$ must also be built.

Concerning the construction of the affine RB matrices and vectors, the GRB method scales linearly on the number ($Q_a$ and $Q_g$) of affine terms of the FE matrices and right hand sides, yielding a computational time which is shorter than the one obtained with the aLSRB method for this phase. Similarly, also the caLSRB method scales linearly thanks to the choice of the matrix $\widetilde{\mathbf{A}}_h^{\boldsymbol{\mu}}$; this latter is taken as the affine approximation of $\mathbf{A}_h^{\boldsymbol{\mu}}$ where only the first POD mode is retained, cf. (46). In this way, the caLSRB method shows the most efficient offline phase, since on one hand it does not require to handle the supremizer snapshots, on the other it provides an efficient construction of the RB affine terms.

In Figure 3 the number of RB functions for the velocity, supremizer and pressure spaces and the total number of RB functions for each method are reported as function of the tolerance $\delta_{RB}$. The total number of RB functions is the same for the aLSRB and caLSRB methods; for the GRB method, the supremizer RB functions need to be considered, due to the velocity enrichment required to ensure the well-posedness of the resulting GRB approximation.




(a) Number of RB functions for the velocity, pressure and supremizer spaces as function of $\delta_{RB}$.



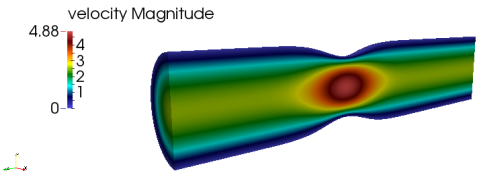

(b) Total number of RB functions as function of $\delta_{RB}$ for different methods.

Figure 3: Number of RB functions as function of $\delta_{RB}$ for each RB space (left) and for each method (right).

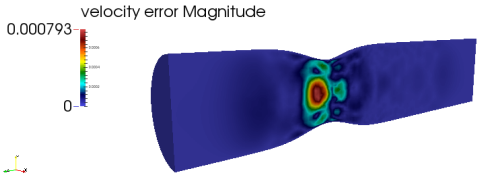
### 6.2.2 Online phase

In Fig. 4 and 5 the FE solution computed for an instance of the parameter and the corresponding errors obtained with the GRB, aLSRB and caLSRB methods are shown for velocity and pressure, respectively.

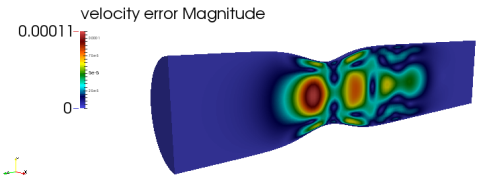
The proposed aLSRB method allows to obtain an exponential decay of the residual $r_{RB}$ with respect to the tolerance $\delta_{RB}$; the trend, in loglog scale, is reported in Fig. 6a and 7a for $\delta_{\text{deim}} = 10^{-5}$ and $\delta_{\text{deim}} = 10^{-7}$, respectively. The convergence of the RB residual $r_{RB}$ is stopped by the tolerance $\delta_{\text{deim}}$, reaching a plateau for the when $\delta_{RB} \leq \delta_{\text{deim}}$, as expected, see e.g. [15, 27]. By comparing the GRB and the new (c)aLSRB methods, when the same tolerances $\delta_{\text{deim}}$ and $\delta_{RB}$ are used, we have that the (c)aLSRB methods allow to compute a more accurate solution during the online phase of about 1 order of magnitude. Moreover, notice that by using the same $\delta_{\text{deim}}$ for the aLSRB methods and the GRB method, the latter requires a lower tolerance $\delta_{RB}$ to reach a solution with the same accuracy, yielding a much larger number of RB functions. Obtaining a more accurate solution with the aLSRB method is an expected result, since the standard LSRB method seeks a RB approximation minimizing the $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$ norm of the residual, and the aLSRB method provides a RB approximation minimizing the $\mathbf{P}_X^{-1}$ norm, where $\mathbf{P}_X^{-1} \approx (\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$, as shown in Proposition 4.2. A similar argument also holds for the caLSRB method and allows to obtain an accuracy comparable (yet slightly larger) to the one obtained with the aLSRB. This is expected since the caLSRB method has been constructed by approximating the aLSRB one.



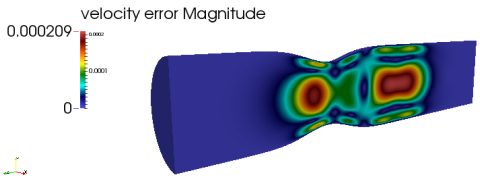

(a) FE velocity magnitude.




(b) GRB velocity error magnitude.

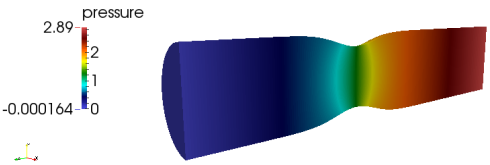



(c) aLSRB velocity error magnitude.

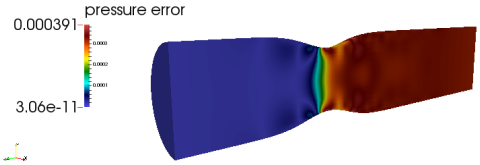



(d) caLSRB velocity error magnitude.

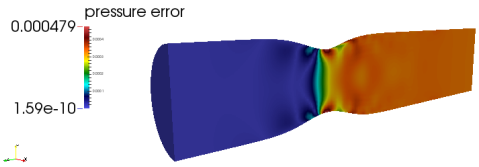
Figure 4: FE velocity solution and GRB and aLSRB errors for $\boldsymbol{\mu} = (2.92, 0.3)$.



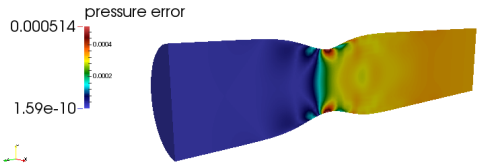

(a) FE pressure magnitude.




(b) GRB pressure error magnitude.




(c) aLSRB pressure error magnitude.

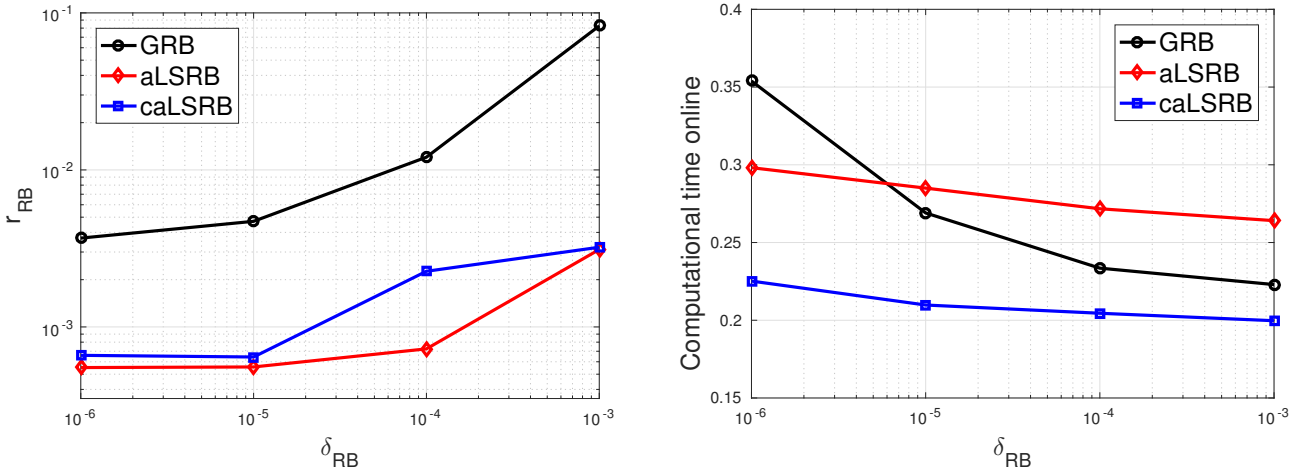



(d) caLSRB pressure error magnitude.

Figure 5: FE pressure solution and GRB, aLSRB and caLSRB errors for $\boldsymbol{\mu} = (2.92, 0.3)$.

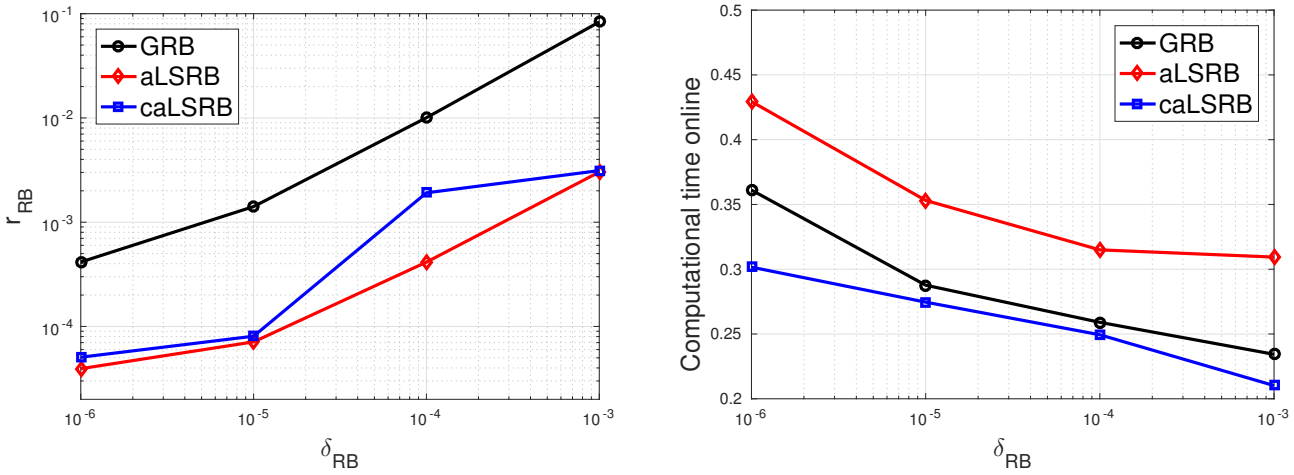
In Fig. 6b and 7b, the computational time required to assemble and solve the RB problem is reported for the three methods as function of $\delta_{RB}$ for the (M)DEIM tolerances $\delta_{\text{deim}} = 10^{-5}, 10^{-7}$, respectively. Depending on the desired level of accuracy and the RB method employed, the computational time required to solve the RB Stokes problem online ranges from 0.2 to 0.45 seconds. We remark that this computational time does not take into account the time required to solve the two additional problems for the domain displacement $\mathbf{d}_h^{\boldsymbol{\mu}}$ and the lifting $\mathbf{r}_h^{\boldsymbol{\mu}}$.

In Table 4, for the three methods examined, we compare the minimum computational time to compute a RB approximation whose residual $r_{RB}$ is lower than a fixed target accuracy (left column). The aLSRB and caLSRB methods confirm to reach a target accuracy in a lower time. In particular, the caLSRB method outperforms, by reaching the same accuracy of the aLSRB method, being significantly faster thanks to the linear dependence of the assembling phase with respect to the number of affine terms. The 'x' in the the GRB column states that the accuracy $10^{-4}$ is not reached when this method with the given tolerance values $\delta_{RB} = 10^{-l}$, $l = 3, 4, 5, 6$ and $\delta_{\text{deim}} = 10^{-5}$, $10^{-7}$. Therefore, one should further decrease $\delta_{RB}$ and $\delta_{\text{deim}}$ to compute a more accurate solution.




(a) Residual. (b) Computational time online.

Figure 6: Residuals in norm $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$ (left) and online computational time as function $\delta_{RB}$ for $\delta_{\text{deim}} = 10^{-5}$computed and averaged on $N_{onl} = 100$ parameters.




(a) Residual. (b) Computational time online.

Figure 7: Residuals in norm $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$ (left) and online computational time as function $\delta_{RB}$ for $\delta_{\text{deim}} = 10^{-7}$computed and averaged on $N_{onl} = 100$ parameters.

Table 4: Computational time (seconds) required by the RB methods to compute a solution satisfying a target accuracy.

| **Accuracy** | **GRB** | **aLSRB** $(\mathbf{P}_X^0)$ | **caLSRB** $(\mathbf{P}_X^0)$ |
|---|---|---|---|
| 1e-02 | 0.27 | 0.26 | 0.20 |
| 1e-03 | 0.36 | 0.27 | 0.21 |
| 1e-04 | x | 0.35 | 0.27 |

# 7 Conclusions

In this paper we have proposed a new algebraic PGRB method which can be generally used when the problem is parametrized with respect to the shape of the computational domain, especially when an analytical dependence of the geometry from the parameters is not known a priori. When such a case occurs, the state of the art PGRB methods are currently not exploitable, since they require an unbearable amount of computation to build the RB problem online due to the $\boldsymbol{\mu}$-dependence of

the matrix $\mathbf{X}_h^{\boldsymbol{\mu}}$.

The key idea of the proposed method relies on approximating the ($\boldsymbol{\mu}$-dependent) matrix $\mathbf{X}_h^{\boldsymbol{\mu}}$ used to build a standard LSRB approximation, with a $\boldsymbol{\mu}-$independent matrix $\mathbf{P}_X$, giving birth to an algebraic LSRB method which we called aLSRB. We have analyzed its theoretical properties by providing well-posedness results and by comparing it with the state of the art techniques to deal with the same class of problems, i.e. a GRB approximation with augmented velocity basis.

Next, we have introduced an approximation of the aLSRB method, the caLSRB method, where the definition of the test space employs a coarse affine approximation of the FE matrix. This choice allows to obtain an assembling phase which depends linearly on the number of affine term of the FE arrays, thus speeding up the construction of the RB approximation when a new instance of the parameter is queried online.

We have numerically investigated the properties of the proposed aLSRB and caLSRB methods, comparing the results with the ones obtained by means of the GRB method. The new proposed methods yield a reduced linear system with lower dimension and require in general a lower number of affine terms in the affine decomposition of the FE arrays to reach a prescribed accuracy. Finally, the results obtained with the (c)aLSRB method are more accurate and computationally cheaper (both at the offline and online stages) than the ones obtained using the GRB method with velocity enrichment.

Theoretical findings and numerical results suggest that the proposed strategies yield an efficient construction of a stable and accurate RB method for Stokes equations. In principle, they would also allow the solution of more involved systems as the Navier-Stokes equations in parametrized geometries: indeed, the efficient reduction of these problems may yield a severely large number of affine terms for the approximation of the operators, possibly hampering the efficiency of the RB approximation. In particular, we believe our approach outlined in Section 5, making use of an affinely approximated test space, and showing a linear dependence on the number of affine contributions, is extremely promising in view of enhancing the efficiency of RB methods in this case.

## Acknowledgements

The research of N. Dal Santo has been supported by the Swiss State Secretariat for Education, Research and Innovation (SERI), project No. C14.0068, in the framework of the COST action number TD1307. The second author thanks the Swiss National Science Foundation for support under the grant FNS 200021E-168311. We acknowledge the Swiss National Supercomputing Centre (CSCS) for providing the CPU resources under project IDs s635 and s796.

# References

[1] A. Abdulle and O. Budáč. A Petrov–Galerkin reduced basis approximation of the stokes equation in parameterized geometries. *Comptes Rendus Mathematique*, 353(7):641–645, 2015.

[2] I. Babuška. Error-bounds for finite element method. *Numerische Mathematik*, 16(4):322–333, 1971.

[3] E. Bache, J. Vega, and A. Velazquez. Model reduction in the back step fluid–thermal problem with variable geometry. *International Journal of Thermal Sciences*, 49(12):2376–2384, 2010.

[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*, 102(5):1136–1161, 2015.

[5] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

[6] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

[7] M. Benzi and A. J. Wathen. Some preconditioning techniques for saddle point problems. In *Model order reduction: theory, research aspects and applications*, pages 195–211. Springer Berlin Heidelberg, 2008.

[8] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust POD models. *Journal of Computational Physics*, 228(2):516 – 538, 2009.

[9] L. Bertagna, S. Deparis, L. Formaggia, D. Forti, and A. Veneziani. The LifeV library: engineering mathematics beyond the proof of concept. *arXiv:1710.06596*, 2017.

[10] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 8(R2):129–151, 1974.

[11] F. Brezzi and K.-J. Bathe. A discourse on the stability conditions for mixed finite element formulations. *Computer methods in applied mechanics and engineering*, 82(1-3):27–57, 1990.

[12] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity–pressure reduced order models for incompressible flows. *Journal of Computational Physics*, 259:598–616, 2014.

[13] C. Canuto, M. Y. Hussaini, A. Quarteroni, and A. Thomas Jr. *Spectral methods in fluid dynamics*. Springer Science & Business Media, 2012.

[14] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[15] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[16] W. Dahmen, C. Huang, C. Schwab, and G. Welper. Adaptive petrov–galerkin methods for first order transport equations. *SIAM journal on numerical analysis*, 50(5):2420–2445, 2012.

[17] S. Deparis. Reduced basis error bound computation of parameter-dependent Navier–Stokes equations by the natural norm approach. *SIAM Journal of Numerical Analysis*, 46(4):2039–2067, 2008.

[18] S. Deparis and G. Rozza. Reduced basis method for multi-parameter-dependent steady Navier-Stokes equations: Applications to natural convection in a cavity. *Journal of Computational Physics*, 228(12):4359–4378, 2009.

[19] H. C. Elman and V. Forstall. Numerical solution of the parameterized steady-state navier–stokes equations using empirical interpolation methods. *Computer Methods in Applied Mechanics and Engineering*, 317:380 – 399, 2017.

[20] H. C. Elman, D. J. Silvester, and A. J. Wathen. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics, 2005.

[21] A.-L. Gerner and K. Veroy. Certified reduced basis methods for parametrized saddle point problems. *SIAM J. Sci. Comput.*, 34(5):A2812–A2836, 2012.

[22] J. S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2016.

[23] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numerical Analysis*, 40(2):492–515, 2002.

[24] A. Manzoni. An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier-Stokes flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(4):1199–1226, 2014.

[25] A. Manzoni and F. Negri. Efficient reduction of PDEs defined on domains with variable shape. In P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban, editors, *Model Reduction of Parametrized Systems*, volume 17, pages 183–199. Springer, Cham, 2017.

[26] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670, 2012.

[27] F. Negri. *Efficient Reduction Techniques for the Simulation and Optimization of Parametrized Systems*. PhD thesis, EPFL, 2015.

[28] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.

[29] F. Negri, A. Manzoni, and G. Rozza. Reduced basis approximation of parametrized optimal flow control problems for the Stokes equations. *Computers & Mathematics with Applications*, 69:319–336, 2015.

[30] A. Quarteroni. *Numerical Models for Differential Problems*, volume 9 of *Modeling, Simulation and Applications (MS&A)*. Springer-Verlag Italia, Milano, 2nd edition, 2014.

[31] A. Quarteroni. *Numerical models for differential problems*, volume 8. Springer, 2014.

[32] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2016.

[33] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier–Stokes equations by reduced basis methods. *Numerical Methods for Partial Differential Equations*, 23(4):923–948, 2007.

[34] M. Rehman, T. Geenen, C. Vuik, G. Segal, and S. MacLachlan. On iterative methods for the incompressible Stokes problem. *International Journal for Numerical methods in fluids*, 65(10):1180–1200, 2011.

[35] G. Rozza. Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. *Computing and Visualization in Science*, 12(1):23–35, Jan 2009.

[36] G. Rozza, D. Huynh, and A. Manzoni. Reduced basis approximation and error bounds for Stokes flows in parametrized geometries: roles of the inf–sup stability constants. *Numerische Mathematik*, 125(1):115–152, 2013.

[37] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer methods in applied mechanics and engineering*, 196(7):1244–1260, 2007.

[38] A. Toselli and O. B. Widlund. *Domain decomposition methods: algorithms and theory*. Springer series in computational mathematics. Springer, Berlin, 2005.

[39] J. Weller, E. Lombardi, M. Bergmann, and A. Iollo. Numerical methods for low-order modeling of fluid flows based on POD. *Int. J. Numer. Methods Fluids*, 63(2):249–268, 2010.

# A Proofs of the results in Section 4.3

In the proofs we will omit the apex $\boldsymbol{\mu}$ for the sake of clearness, i.e. $\mathbf{X}_h = \mathbf{X}_h^{\boldsymbol{\mu}}$.

## A.1 Lemma 4.1

*Proof.* Being $\mathbf{X}_h$ and $\mathbf{P}_X$ symmetric and positive definite, for any $\mathbf{y} \in \mathbb{R}^{N_h}$ it holds

$$\begin{aligned}\|\mathbf{y}\|^2_{\mathbf{X}_h} &= \left(\mathbf{X}_h^{1/2}\mathbf{y}, \mathbf{X}_h^{1/2}\mathbf{y}\right)_2 = \left(\mathbf{P}_X^{-1/2}\mathbf{P}_X\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}, \mathbf{X}_h^{1/2}\mathbf{y}\right)_2 = \left(\mathbf{P}_X\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}, \mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}\right)_2 \\ &= \left(\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}, \mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}\right)_{\mathbf{P}_X} = \|\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}\|^2_{\mathbf{P}_X} \leq \|\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\|^2_{\mathbf{P}_X}\|\mathbf{y}\|^2_{\mathbf{P}_X},\end{aligned}$$

and there exists an element $\mathbf{y}_0 \in \mathbb{R}^{N_h}$ where equality is reached. This leads to an optimal $C = \|\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\|_{\mathbf{P}_X}$. Similarly, by inverting the roles of $\mathbf{P}_X$ and $\mathbf{X}_h$ and following the same argument, we have that $c = 1/\|\mathbf{X}_h^{-1/2}\mathbf{P}_X^{1/2}\|_{\mathbf{X}_h}$. □

## A.2 Lemma 4.2

*Proof.* We rewrite the optimal values for $C$ and $c$ as it follows

$$\begin{aligned}
C^2 &= \|\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\|^2_{\mathbf{P}_X} = \sup_{\mathbf{y}\in\mathbb{R}^{N_h},\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y}\|^2_{\mathbf{P}_X}}{\|\mathbf{y}\|^2_{\mathbf{P}_X}} \\
&= \sup_{\mathbf{y}\in\mathbb{R}^{N_h},\mathbf{y}\neq\mathbf{0}} \frac{(\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y},\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\mathbf{y})_{\mathbf{P}_X}}{(\mathbf{y},\mathbf{y})_{\mathbf{P}_X}} = \sup_{\mathbf{y}\in\mathbb{R}^{N_h},\mathbf{y}\neq\mathbf{0}} \frac{(\mathbf{X}_h^{1/2}\mathbf{y},\mathbf{X}_h^{1/2}\mathbf{y})_2}{(\mathbf{P}_X^{1/2}\mathbf{y},\mathbf{P}_X^{1/2}\mathbf{y})_2} \\
&= \sup_{\mathbf{y}\in\mathbb{R}^{N_h},\mathbf{y}\neq\mathbf{0}} \frac{(\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\mathbf{P}_X^{1/2}\mathbf{y},\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\mathbf{P}_X^{1/2}\mathbf{y})_2}{(\mathbf{P}_X^{1/2}\mathbf{y},\mathbf{P}_X^{1/2}\mathbf{y})_2} \\
&= \sup_{\mathbf{w}\in\mathbb{R}^{N_h},\mathbf{w}\neq\mathbf{0}} \frac{(\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\mathbf{w},\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\mathbf{w})_2}{(\mathbf{w},\mathbf{w})_2} = \sup_{\mathbf{w}\in\mathbb{R}^{N_h},\mathbf{w}\neq\mathbf{0}} \frac{\|\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\mathbf{w}\|_2^2}{\|\mathbf{w}\|_2^2} \\
&= \|\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\|_2^2. \qquad (49)
\end{aligned}$$

Similarly, we have that $\|\mathbf{X}_h^{-1/2}\mathbf{P}_X^{1/2}\|_{\mathbf{X}_h} = \|\mathbf{P}_X^{1/2}\mathbf{X}_h^{-1/2}\|_2$, yielding

$$\begin{aligned}
\frac{c}{C} &= \left(\|\mathbf{X}_h^{-1/2}\mathbf{P}_X^{1/2}\|_{\mathbf{X}_h}\|\mathbf{P}_X^{-1/2}\mathbf{X}_h^{1/2}\|_{\mathbf{P}_X}\right)^{-1} = \left(\|\mathbf{P}_X^{1/2}\mathbf{X}_h^{-1/2}\|_2\|\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\|_2\right)^{-1} \\
&= \left[\mathcal{K}_2(\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2})\right]^{-1} = \left[\mathcal{K}_2(\mathbf{P}_X^{1/2}\mathbf{X}_h^{-1/2})\right]^{-1},
\end{aligned}$$

where the last two relations are both used to find different equalities. Next, by recalling the definition of condition number (with respect to the Euclidean norm) $\mathcal{K}_2$ for a matrix, we obtain

$$\begin{aligned}
\mathcal{K}_2(\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}) &= \sqrt{\frac{\lambda_{\max}\left((\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2})^T\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\right)}{\lambda_{\min}\left((\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2})^T\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1/2}\right)}} = \sqrt{\frac{\lambda_{\max}\left(\mathbf{P}_X^{-1/2}\mathbf{X}_h\mathbf{P}_X^{-1/2}\right)}{\lambda_{\min}\left(\mathbf{P}_X^{-1/2}\mathbf{X}_h\mathbf{P}_X^{-1/2}\right)}} \\
&= \sqrt{\mathcal{K}_2\left(\mathbf{P}_X^{-1/2}\mathbf{X}_h\mathbf{P}_X^{-1/2}\right)},
\end{aligned}$$

which verifies the second equality of (35). On the other hand we have

$$\begin{aligned}
\mathcal{K}_2(\mathbf{P}_X^{1/2}\mathbf{X}_h^{-1/2}) &= \sqrt{\mathcal{K}_2\left(\mathbf{X}_h^{-1/2}\mathbf{P}_X\mathbf{X}_h^{-1/2}\right)} = \sqrt{\|\mathbf{X}_h^{-1/2}\mathbf{P}_X\mathbf{X}_h^{-1/2}\|_2\|\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1}\mathbf{X}_h^{1/2}\|_2} \\
&= \sqrt{\|\mathbf{X}_h^{-1}\mathbf{P}_X\|_{\mathbf{X}_h}\|\mathbf{P}_X^{-1}\mathbf{X}_h\|_{\mathbf{X}_h}} = \sqrt{\mathcal{K}_{\mathbf{X}_h}(\mathbf{P}_X^{-1}\mathbf{X}_h)},
\end{aligned}$$

where we have used that $\|\mathbf{X}_h^{-1/2}\mathbf{P}_X\mathbf{X}_h^{-1/2}\|_2 = \|\mathbf{X}_h^{-1}\mathbf{P}_X\|_{\mathbf{X}_h}$ and $\|\mathbf{X}_h^{1/2}\mathbf{P}_X^{-1}\mathbf{X}_h^{1/2}\|_2 = \|\mathbf{P}_X^{-1}\mathbf{X}_h\|_{\mathbf{X}_h}$; these latter relationships are verified similarly to (49), and their proof can therefore be omitted. □

# MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

**31/2018** Quarteroni, A.
*The role of statistics in the era of big data: A computational scientist' perspective*

**30/2018** Ieva, F.; Palma, F.; Romo, J.
*Bootstrap-based Inference for Dependence in Multivariate Functional Data*

**29/2018** Manzoni, A; Bonomi, D.; Quarteroni, A.
*Reduced order modeling for cardiac electrophysiology and mechanics: new methodologies, challenges & perspectives*

**28/2018** Gerbi, A.; Dede', L.; Quarteroni, A.
*Segregated algorithms for the numerical simulation of cardiac electromechanics in the left human ventricle*

**26/2018** Vergara, C.; Zonca, S.
*Extended Finite Elements method for fluid-structure interaction with an immersed thick non-linear structure*

**25/2018** Chave, F.; Di Pietro, D.A.; Formaggia, L.
*A Hybrid High-Order method for passive transport in fractured porous media*

**27/2018** Antonietti, P.F.; Verani, M.; Vergara, C.; Zonca, S.
*Numerical solution of fluid-structure interaction problems by means of a high order Discontinuous Galerkin method on polygonal grids*

**24/2018** Bassi, C.; Abbà, A.; Bonaventura,L.; Valdettaro,L.
*Direct and Large Eddy Simulation of three-dimensional non-Boussinesq gravity currents with a high order DG method*

**21/2018** Gervasio, P.; Dede', L.; Chanon, O.; Quarteroni, A.
*Comparing Isogeometric Analysis and Spectral Element Methods: accuracy and spectral properties*

**23/2018** Benacchio, T.;Bonaventura,L.
*A seamless extension of DG methods for hyperbolic problems to unbounded domains*