Multi space reduced basis preconditioners for 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

Multi space reduced basis preconditioners for parametrized Stokes equations

N. Dal Santo[a,*], S. Deparis[a], A. Manzoni[b], A. Quarteroni[a,b]

[a] *CMCS-MATH-SB, École Polytechnique Fédérale de Lausanne (EPFL), Station 8, CH-1015 Lausanne, Switzerland.*
[b] *MOX, Department of Mathematics, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133 Milano, Italy.*

Abstract

We introduce a two-level preconditioner for the efficient solution of large scale saddle-point linear systems arising from the finite element (FE) discretization of parametrized Stokes equations. This preconditioner extends the Multi Space Reduced Basis (MSRB) preconditioning method proposed in [1]; it combines an approximated block (fine grid) preconditioner with a reduced basis (RB) solver which plays the role of coarse component. A sequence of RB spaces, constructed either with an enriched velocity formulation or a Petrov-Galerkin projection, is built. Each RB coarse component is defined to perform a single iteration of the iterative method at hand. The flexible GMRES (FGMRES) algorithm is employed to solve the resulting preconditioned system and targets small tolerances with a very small iteration count and in a very short time. Numerical test cases for Stokes flows in three dimensional parameter-dependent geometries are considered to assess the numerical properties of the proposed technique in different large scale computational settings.

Keywords: Preconditioning techniques, reduced basis method, finite element method, parametrized Stokes equations, computational fluid dynamics
2010 MSC: 76D07, 65N12, 65N30, 65F08, 65F10

1. Introduction

This work is concerned with the efficient numerical solution of parametrized saddle-point systems arising from the FE discretization of partial differential equations (PDEs). We meet this kind of problems in several contexts, e.g., in the mixed formulations of elliptic PDEs, incompressible elasticity, optimal control problems for elliptic PDEs and incompressible fluid flow problems. Here we focus on parametrized incompressible Stokes equations, describing viscous incompressible stationary flows in the limit $Re \to 0$, where Re is the the flow Reynolds number. Denote 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 encoding physical and/or geometrical properties. The apex $\boldsymbol{\mu}$ means that a variable depends on the parameter $\boldsymbol{\mu}$. Given a $\boldsymbol{\mu}-$dependent domain $\Omega^{\boldsymbol{\mu}} \subset \mathbb{R}^d$, $d = 2, 3$, such that, for any $\boldsymbol{\mu} \in \mathcal{D}$, $\partial\Omega^{\boldsymbol{\mu}} = \Gamma^{\boldsymbol{\mu}}_{out} \cup \Gamma^{\boldsymbol{\mu}}_{in} \cup \Gamma^{\boldsymbol{\mu}}_{w}$ and $\mathring{\Gamma}^{\boldsymbol{\mu}}_{out} \cap \mathring{\Gamma}^{\boldsymbol{\mu}}_{in} = \mathring{\Gamma}^{\boldsymbol{\mu}}_{w} \cap \mathring{\Gamma}^{\boldsymbol{\mu}}_{in} = \mathring{\Gamma}^{\boldsymbol{\mu}}_{out} \cap \mathring{\Gamma}^{\boldsymbol{\mu}}_{w} = \emptyset$, the Stokes equations read

$$\begin{cases} -\nu^{\boldsymbol{\mu}}\Delta\vec{u}^{\boldsymbol{\mu}} + \nabla p^{\boldsymbol{\mu}} = \vec{f}^{\boldsymbol{\mu}} & \text{in } \Omega^{\boldsymbol{\mu}} \\ \nabla \cdot \vec{u}^{\boldsymbol{\mu}} = 0 & \text{in } \Omega^{\boldsymbol{\mu}} \\ \vec{u} = \vec{g}^{\boldsymbol{\mu}}_D & \text{on } \Gamma^{\boldsymbol{\mu}}_{in} \\ \vec{u} = \vec{0} & \text{on } \Gamma^{\boldsymbol{\mu}}_{w} \\ -p^{\boldsymbol{\mu}}\vec{n}^{\boldsymbol{\mu}} + \nu^{\boldsymbol{\mu}}\dfrac{\partial\vec{u}^{\boldsymbol{\mu}}}{\partial\vec{n}^{\boldsymbol{\mu}}} = \vec{g}^{\boldsymbol{\mu}}_N & \text{on } \Gamma^{\boldsymbol{\mu}}_{out}, \end{cases} \tag{1}$$

[*] Corresponding author
Email address: niccolo.dalsanto@epfl.ch (N. Dal Santo)

where $(\vec{u}^{\boldsymbol{\mu}}, p^{\boldsymbol{\mu}})$ are the velocity and pressure fields describing a fluid with viscosity $\nu^{\boldsymbol{\mu}}$, respectively, while $\vec{f}^{\boldsymbol{\mu}}$ encodes distributed sources. Problem (1) can be written under mixed form, yielding a noncoercive variational problem, whose well-posedness is ensured according to the general theory on saddle-point problems [2, 3].

Numerical methods based on (Petrov-)Galerkin projection onto a finite dimensional subspace, as the FE or spectral element methods, are viable strategies for the numerical solution of (1), see e.g. [4, 5]. However, when they are employed, an *inf-sup* condition must be satisfied at the finite dimensional level to ensure the well-posedness of the numerical problem. Such a condition poses strict constraints on the choice of the FE spaces where the approximate solution of problem (1) is sought. In this paper, we use an *inf-sup* stable FE couple of spaces, for instance those based on $\mathbb{P}^2 - \mathbb{P}^1$ (Taylor-Hood) polynomial subspaces for the discretization of the velocity and pressure fields, respectively. The resulting FE approximation yields the solution of a parametrized saddle-point linear system

$$\begin{bmatrix} \mathbf{D}_h^{\boldsymbol{\mu}} & (\mathbf{B}_h^{\boldsymbol{\mu}})^T \\ \mathbf{B}_h^{\boldsymbol{\mu}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_h^{\boldsymbol{\mu}} \\ \mathbf{p}_h^{\boldsymbol{\mu}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h^{\boldsymbol{\mu}} \\ \mathbf{0} \end{bmatrix}, \tag{2}$$

of (possibly very large) dimension N_h, which is given by the sum of the velocity and pressure degrees of freedom, see e.g. [5, 6].

A linear system involving a matrix of the form (2) occurs in many different contexts; in addition to the Stokes equations, we recall optimal control for elliptic and incompressible fluid flows problems and incompressible elasticity. Consequently, the numerical solution of this kind of linear systems has been largely addressed over the years, and several techniques involving domain decomposition and multilevel methods have been proposed, [7, 5]. Among these techniques, algebraic multgrid (AMG) methods have demonstrated to be very efficient when a large number of unknowns is considered for the Stokes equations, see e.g. [8, 9, 10]. For further references on the topic we refer to [11, 12, 13].
Another important class of preconditioners for the Stokes equations is then given by block-preconditioners, which are designed to explicitly exploit the block structure in (2). Let us start from the following factorization of the saddle-point matrix:

$$\begin{bmatrix} \mathbf{D}_h^{\boldsymbol{\mu}} & (\mathbf{B}_h^{\boldsymbol{\mu}})^T \\ \mathbf{B}_h^{\boldsymbol{\mu}} & 0 \end{bmatrix} = \mathcal{L}_h^{\boldsymbol{\mu}} \mathcal{D}_h^{\boldsymbol{\mu}} \mathcal{U}_h^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{I}_{N_u} & 0 \\ \mathbf{B}_h^{\boldsymbol{\mu}} (\mathbf{D}_h^{\boldsymbol{\mu}})^{-1} & \mathbf{I}_{N_p} \end{bmatrix} \begin{bmatrix} \mathbf{D}_h^{\boldsymbol{\mu}} & 0 \\ 0 & \mathbf{S}_h^{\boldsymbol{\mu}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N_u} & (\mathbf{D}_h^{\boldsymbol{\mu}})^{-1} (\mathbf{B}_h^{\boldsymbol{\mu}})^T \\ 0 & \mathbf{I}_{N_p} \end{bmatrix} \tag{3}$$

where

$$\mathbf{S}_h^{\boldsymbol{\mu}} = -\mathbf{B}_h^{\boldsymbol{\mu}} (\mathbf{D}_h^{\boldsymbol{\mu}})^{-1} (\mathbf{B}_h^{\boldsymbol{\mu}})^T$$

is the Schur complement matrix, which plays an essential role in designing an efficient preconditioner for the Stokes problem. The cost of inverting the blocks in (3) would be equivalent to invert the Stokes matrix, therefore the approximated factorization $\widetilde{\mathcal{L}}_h^{\boldsymbol{\mu}} \widetilde{\mathcal{D}}_h^{\boldsymbol{\mu}} \widetilde{\mathcal{U}}_h^{\boldsymbol{\mu}} \approx \mathcal{L}_h^{\boldsymbol{\mu}} \mathcal{D}_h^{\boldsymbol{\mu}} \mathcal{U}_h^{\boldsymbol{\mu}}$, is employed in practice, where two surrogate operators $\widetilde{\mathbf{D}}_h^{\boldsymbol{\mu}}$ and $\tilde{\mathbf{S}}_h^{\boldsymbol{\mu}}$ are used to replace $\mathbf{D}_h^{\boldsymbol{\mu}}$ and $\mathbf{S}_h^{\boldsymbol{\mu}}$ in (3), respectively. This block-preconditioning strategy has been particularly exploited in the field of fluid dynamics and optimal control problems. Relevant examples are the least-squares commutator (LSC) preconditioner [14, 15], the pressure-convection-diffusion preconditioner [16, 17] and the pressure mass matrix (PMM) preconditioner [18], recent works on this topic include [19, 20]. Finally, SIMPLE type preconditioners are obtained from (3), where $\mathbf{D}_h^{\boldsymbol{\mu}}$ is substituted with its diagonal when building the Schur complement and for the update of the velocity, see e.g. [21, 22, 23, 24]. As a matter of fact, SIMPLE preconditioners are particularly effective when dealing with problems diagonally dominant, as in the case of unsteady problems with relative small timesteps. Finally, we point to [25, 26, 27] for a broad review on numerical methods for saddle-point systems.

All the methods discussed are developed for a single instance of the parameter and do not take advantage of any underlying $\boldsymbol{\mu}$-dependence of the PDE in case a parameter-dependent problem is considered. In this paper, we are interested in the efficient solution of (2) for many (say, hundreds or thousands) instances of $\boldsymbol{\mu}$. This may be an issue, for instance, when dealing with uncertainty quantification, sensitivity analysis or PDE-constrained optimization, to mention some remarkable scenarios. With this goal in mind, in this paper we develop a two-level preconditioner which exploits the underlying parametrization of the Stokes equations by means of a coarse component based on reduced order modeling (ROM) techniques. This latter is aimed at speeding the iterative solution of the FE linear system and is combined with a fine grid preconditioner which can be chosen as any preconditioner at disposal.

In the last decade, ROM techniques emerged as a convenient strategy when dealing with parametrized problems; several methods to address the approximation of parametrized (Navier-)Stokes equations have been designed, see e.g. [28, 29, 30, 31, 32]. In this work, we exploit a particular case of projection-based ROM techniques, the RB method, to build a coarse correction in a two level preconditioner for the efficient solution of large-scale parameter-dependent Stokes equations.

The RB method aims at computing an approximated (reduced) solution of the parameter dependent PDE as a linear combination of few, global problem-dependent, basis functions. These latter are obtained from a set of FE solutions (or snapshots) corresponding to different values of the parameters. Such a method is built in two phases. In the *offline* phase, we construct a RB space of dimension $N \ll N_h$ whose basis is obtained by (properly orthonormalized) linear combinations of FE solutions of the parametrized PDE. In the *online* phase, the RB problem is obtained by either Petrov-Galerkin, see [33, 34], or *enriched* Galerkin projection, resulting in a small linear system which is solved at the place of the large FE problem, usually with direct methods.A crucial assumption that allows to speed up the RB method is made by requiring that the matrix and the right hand side in (2) depend affinely on the parameter, allowing to assemble the RB problem independently on the FE dimension. Should this assumption not be verified, the Empirical Interpolation Method (EIM) or Discrete EIM (DEIM) can be used to construct an approximated affine decomposition, which however can lead to a critical overhead during both the offline and online phases. For an extensive review of RB methods for parameter dependent PDEs see, e.g., [35, 36].

The RB method for parametrized elliptic PDEs has been used to define the coarse correction in the Multi Space Reduced Basis MSRB preconditioning strategy proposed in [1], and further analyzed in [37]. Such a technique relies on the multiplicative combination of a fine grid, nonsingular operator $\mathbf{P}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ with an iteration (k-)dependent coarse correction $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ built upon the RB method. The preconditioner exploits the parameter dependence of the PDE by projecting the error equation at step k onto a k-dependent RB space tailored to provide a very accurate approximation of the k-th error equation. As a result, the number of iterations required by the iterative solver (in our case the flexible GMRES [38]) to reach a desired accuracy is very small. The MSRB preconditioning technique has demonstrated to be particularly helpful when dealing with strongly nonaffine problems, since it depends in a milder way (if compared to standards RB methods) on the approximated affine decomposition constructed with (D)EIM. Also in this work we provide numerical evidence of the advantages of using the MSRB preconditioners instead of standard RB methods when the problem is strongly nonaffine.

The main contribution of our paper consists in extending the MSRB preconditioning framework to saddle-point problems. To this aim, the structure of the preconditioner for the Stokes problem is as the one for elliptic problems, that is a multiplicative combination of a fine component and an iteration-dependent RB coarse component, where the latter is trained to accurately solve the error equation appearing at iteration k of FGMRES to speed up the solution of the original linear system (2). The saddle-point nature of the Stokes FE matrix prevents from using the RB coarse components developed in [1] for the MSRB preconditioner. Hence, in this paper, different aspects are investigated to extend the MSRB preconditioning framework to the Stokes equations, dealing with *i)* the *construction* of RB coarse components suited when facing a saddle-point system, *ii)* the *well-posedness* of the resulting preconditioner and *iii)* the practical implementation algorithms to construct the preconditioner. Finally, the MSRB preconditioner is employed on problem of practical interest. We assessed numerically the performance by considering the 3-D Stokes equations defined in parameter-dependent domains, for which a mapping from a reference domain is not necessarily known analytically. We employ the PMM predconditioner as fine component $\mathbf{P}_h^{\boldsymbol{\mu}}$ for the construction of the MSRB preconditioner and we compare the results obtained with the ones computed by using the PMM preconditioner alone. This allows us to quantify the computational gain resulting by combining the RB coarse component with the PMM preconditioner. We highlight that any known preconditioner can be employed as fine component; we choose the PMM preconditioner since it is one of the most popular and credited preconditioners used for solving the Stokes FE linear system, thanks to the spectral properties of the preconditioned FE matrix. Secondly, we consider a patient-specific geometry of the carotid bifurcation and compare with the standard RB method for the Stokes equations. The underlying complex computational domain and parametrization don't allow for an affine decomposition of the FE matrix and right hand side. On the other hand, DEIM is used to build an approximated affine decomposition, which, however, strongly affects the speed up gained with the RB method. On the other hand, the MSRB preconditioner is shown to depend in a milder way on the approximated affine decomposition, providing a significantly improved performance compared to the standard RB method.

The structure of the paper is as follows. In Section 2 we introduce the Stokes equations, their FE approximation and the methods required to solve the saddle-point FE linear system (2) and in Section 3 we recall the RB method for the Stokes equations. In Section 4 we present in a general setting the MSRB preconditioner and in Section 5 we build the MSRB preconditioner for the parametrized Stokes equations, relying on either a PG-RB or an *enriched* G-RB coarse correction and highlighting the assumptions required to guarantee the well-posedness of the resulting preconditioner operator. In Section 6 we outline the algorithmic procedures employed to build the MSRB preconditioner and in Section 7 we present numerical results obtained with the MSRB preconditioner. Finally, in Section 8 we draw some conclusions and we report in AppendixA the details on how to construct a stable RB Stokes problem, for those readers less familiar with this topic.

For the sake of notation, hereon we denote scalar field functions by lower case letters, as $a(\vec{x}) \in \mathbb{R}$, vector field functions with an arrow, as $\vec{a}(\vec{x}) \in \mathbb{R}^d$, for $d > 1$, algebraic vectors by bold lower case letters, as $\mathbf{a} \in \mathbb{R}^n$, and matrices by bold capital letters, as $\mathbf{A} \in \mathbb{R}^{n\times n}$. Moreover, given a symmetric and positive definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, we denote by $\|\cdot\|_{\mathbf{A}}$ the norm and by $(\cdot,\cdot)_{\mathbf{A}}$ the scalar product defined as

$$\|\mathbf{a}\|_{\mathbf{A}} = \sqrt{\mathbf{a}^T\mathbf{A}\mathbf{a}} \qquad \forall\, \mathbf{a} \in \mathbb{R}^n, \qquad (\mathbf{a},\mathbf{b})_{\mathbf{A}} = \mathbf{a}^T\mathbf{A}\mathbf{b} \qquad \forall\, \mathbf{a},\, \mathbf{b} \in \mathbb{R}^n.$$

2. Parametrized Stokes equations: settings and preliminaries

In this section we introduce the weak formulation of the Stokes equations (1), together with the resulting FE approximation. We introduce a 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

$$V^{\boldsymbol{\mu}} = \left\{\vec{v} \in \left(H^1(\Omega^{\boldsymbol{\mu}})\right)^d : \vec{v}\big|_{\Gamma_w^{\boldsymbol{\mu}}} = \vec{0},\ \vec{v}\big|_{\Gamma_{in}^{\boldsymbol{\mu}}} = \vec{0}\right\}, \qquad Q^{\boldsymbol{\mu}} = L^2(\Omega^{\boldsymbol{\mu}}),$$

equipped with scalar products (and the corresponding induced norms) $(\cdot,\cdot)_{V^{\boldsymbol{\mu}}} = (\cdot,\cdot)_{H_0^1(\Omega^{\boldsymbol{\mu}})}$ 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 (1) reads: find $(\vec{u}^{\boldsymbol{\mu}}, p^{\boldsymbol{\mu}}) \in V^{\boldsymbol{\mu}} \times Q^{\boldsymbol{\mu}}$

such that

$$\begin{cases} a^{\boldsymbol{\mu}}(\vec{u}^{\boldsymbol{\mu}},\vec{v}) + b^{\boldsymbol{\mu}}(\vec{v},p^{\boldsymbol{\mu}}) = f^{\boldsymbol{\mu}}(\vec{v}) - a^{\boldsymbol{\mu}}(\vec{r}^{\boldsymbol{\mu}}_{\vec{g}_D},\vec{v}) & \forall \vec{v} \in V^{\boldsymbol{\mu}} \\ b^{\boldsymbol{\mu}}(\vec{u}^{\boldsymbol{\mu}},q) = -b^{\boldsymbol{\mu}}(\vec{r}^{\boldsymbol{\mu}}_{\vec{g}_D},q) & \forall q \in Q^{\boldsymbol{\mu}}, \end{cases} \tag{4}$$

where for any $\vec{u},\vec{v} \in V^{\boldsymbol{\mu}}$ and $q \in Q^{\boldsymbol{\mu}}$ we define the forms in (4) as

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

2.1. Finite element approximation of the Stokes equations

After using a FE approximation method with a stable FE couple (e.g. $\mathbb{P}^2 - \mathbb{P}^1$ finite elements, for velocity and pressure, respectively), an approximation to $(\vec{u}^{\boldsymbol{\mu}},\, p^{\boldsymbol{\mu}})$ is obtained by solving a parametrized saddle-point linear system under the form

$$\mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{z}^{\boldsymbol{\mu}}_h = \mathbf{g}^{\boldsymbol{\mu}}_h, \tag{5}$$

where

$$\mathbf{A}^{\boldsymbol{\mu}}_h = \begin{bmatrix} \mathbf{D}^{\boldsymbol{\mu}}_h & (\mathbf{B}^{\boldsymbol{\mu}}_h)^T \\ \mathbf{B}^{\boldsymbol{\mu}}_h & 0 \end{bmatrix} \in \mathbb{R}^{N_h \times N_h}, \qquad \mathbf{z}^{\boldsymbol{\mu}}_h = \begin{bmatrix} \mathbf{u}^{\boldsymbol{\mu}}_h \\ \mathbf{p}^{\boldsymbol{\mu}}_h \end{bmatrix} \in \mathbb{R}^{N_h} \qquad \text{and} \qquad \mathbf{g}^{\boldsymbol{\mu}}_h = \begin{bmatrix} \mathbf{f}^{\boldsymbol{\mu}}_h \\ \mathbf{r}^{\boldsymbol{\mu}}_h \end{bmatrix} \in \mathbb{R}^{N_h}, \tag{6}$$

with $N_h = N^u_h + N^p_h$ and N^u_h, N^p_h the FE dimensions for the velocity and pressure fields, respectively. The matrix $\mathbf{D}^{\boldsymbol{\mu}}_h \in \mathbb{R}^{N^u_h \times N^u_h}$ corresponds to the bilinear form $a^{\boldsymbol{\mu}}(\cdot,\cdot)$ and is positive definite, while the matrix $\mathbf{B}^{\boldsymbol{\mu}}_h \in \mathbb{R}^{N^p_h \times N^u_h}$ corresponds to the bilinear form $b^{\boldsymbol{\mu}}(\cdot,\cdot)$. The resulting block matrix $\mathbf{A}^{\boldsymbol{\mu}}_h$ is indefinite and to guarantee its nonsingularity one must ensure that there exists $\beta > 0$ such that the following *inf-sup* condition is fulfilled, uniformly across the parameter space,

$$\beta^{\boldsymbol{\mu}}_h = \inf_{\mathbf{z}_h \in \mathbb{R}^{N_h}} \sup_{\mathbf{w}_h \in \mathbb{R}^{N_h}} \frac{\mathbf{w}^T_h \mathbf{A}^{\boldsymbol{\mu}}_h \mathbf{z}_h}{\|\mathbf{z}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_h} \|\mathbf{w}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_h}} \geq \beta \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{7}$$

where the symmetric and positive definite matrix $\mathbf{X}^{\boldsymbol{\mu}}_h \in \mathbb{R}^{N_h \times N_h}$ algebraically encodes the scalar product $(\cdot,\cdot)_{V^{\boldsymbol{\mu}} \times Q^{\boldsymbol{\mu}}}$ and is built as a block diagonal matrix

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

here $\mathbf{X}^{\boldsymbol{\mu}}_u \in \mathbb{R}^{N^u_h \times N^u_h}$ and $\mathbf{X}^{\boldsymbol{\mu}}_p \in \mathbb{R}^{N^p_h \times N^p_h}$ encode the scalar products over the spaces $V^{\boldsymbol{\mu}}$ and $Q^{\boldsymbol{\mu}}$ at the FE level, respectively. We highlight that one could alternatively ensure the well-posedness of (5) in terms of the matrix $\mathbf{B}_h$, by requiring the existence of $\beta_p > 0$

$$\beta^{\boldsymbol{\mu}}_{hp} = \inf_{\mathbf{q}_h \in \mathbb{R}^{N^p_h}} \sup_{\mathbf{v}_h \in \mathbb{R}^{N^u_h}} \frac{\mathbf{v}^T_h \mathbf{B}^{\boldsymbol{\mu}}_h \mathbf{q}_h}{\|\mathbf{v}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_u} \|\mathbf{q}_h\|_{\mathbf{X}^{\boldsymbol{\mu}}_p}} \geq \beta_p \qquad \forall \boldsymbol{\mu} \in \mathcal{D}; \tag{9}$$

notice that (9) together with the positive definiteness of $\mathbf{D}^{\boldsymbol{\mu}}_h$ is equivalent to (7).

Many effective preconditioning techniques have been proposed for solving the linear system (5), among which we mention multilevel methods, domain decomposition preconditioners and block preconditioners [21, 22, 14, 15, 18, 7, 5]. In this paper we take into account block-triangular preconditioners of the form $\mathcal{D}_h^{\boldsymbol{\mu}}\mathcal{U}_h^{\boldsymbol{\mu}}$ which arise from the factorization (3), however everything can be devised also for $\mathcal{L}_h^{\boldsymbol{\mu}}\mathcal{D}_h^{\boldsymbol{\mu}}$ and $\mathcal{L}_h^{\boldsymbol{\mu}}, \mathcal{D}_h^{\boldsymbol{\mu}}\mathcal{U}_h^{\boldsymbol{\mu}}$-type preconditioners. The product $\mathcal{D}_h^{\boldsymbol{\mu}}\mathcal{U}_h^{\boldsymbol{\mu}}$ takes the following form

$$\mathbf{P}_t^{\boldsymbol{\mu}} = \mathcal{D}_h^{\boldsymbol{\mu}}\mathcal{U}_h^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{D}_h^{\boldsymbol{\mu}} & (\mathbf{B}_h^{\boldsymbol{\mu}})^T \\ 0 & \mathbf{S}_h^{\boldsymbol{\mu}} \end{bmatrix} \tag{10}$$

and if used as preconditioner within the preconditioned GMRES method, it allows to reach convergence (in exact arithmetic) in 2 iterations. However, at each iteration of the chosen Krylov method the inverse of $\mathbf{P}_t^{\boldsymbol{\mu}}$ needs to be applied to a Krylov basis function $\mathbf{v}_k$; this shall involve the inverse matrix of the Schur complement $\mathbf{S}_h^{\boldsymbol{\mu}}$ and the inverse of $\mathbf{D}_h^{\boldsymbol{\mu}}$, which are both extremely demanding to apply. Approximated block-triangular preconditioners are developed by approximating the inverse matrices of $\mathbf{S}_h^{\boldsymbol{\mu}}$ and $\mathbf{D}_h^{\boldsymbol{\mu}}$ with proper surrogates $\tilde{\mathbf{S}}_h^{\boldsymbol{\mu}}$ and $\tilde{\mathbf{D}}_h^{\boldsymbol{\mu}}$, respectively, e.g. with two corresponding preconditioners or inner iterations.

As iterative solver for (5), we employ the flexible GMRES method [39], see algorithm (1). This method provides a variant of the GMRES method able to deal with an iteration-dependent preconditioner, such as the one defined in (10) when inner iterations are employed (instead of computing exactly the inverse matrices of $\mathcal{D}_h^{\boldsymbol{\mu}}$ and $\mathbf{S}_h^{\boldsymbol{\mu}}$). This also proves to be necessary in view of the application of the proposed MSRB preconditioner, since this latter relies on an iteration dependent RB coarse correction. In algorithm 1, $\mathbf{v}_k$ represents the $k-$th Krylov basis and at line 4 the preconditioning step is reported.

Algorithm 1 Flexible GMRES [38]

```
 1: procedure FGMRES($\mathbf{A}, \mathbf{b}, \mathbf{x}_0, \{\mathbf{M}_k\}_k$)
 2:     Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$
 3:     for $k = 1, \dots, m$ do
 4:         Compute $\mathbf{z}_k = \mathbf{M}_k^{-1}\mathbf{v}_k$
 5:         Compute $\mathbf{w} = \mathbf{A}\mathbf{z}_k$
 6:         for $j = 1, \dots, k$ do
 7:             $h_{j,k} = (\mathbf{w}, \mathbf{v}_j)$
 8:             $\mathbf{w} = \mathbf{w} - h_{j,k}\mathbf{v}_j$
 9:         end for
10:         Compute $h_{k+1,k} = \|\mathbf{w}\|$ and $\mathbf{v}_{k+1} = \mathbf{w}/h_{k+1,k}$
11:         Define $\mathbf{Z}_m = [\mathbf{z}_1, \dots, \mathbf{z}_m]$, $\tilde{\mathbf{H}}_m = \{h_{j,k}\}_{1 \le j \le k+1;\, 1 \le k \le m}$
12:     end for
13:     Compute $\mathbf{y}_m = \arg\min_{\mathbf{y} \in \mathbb{R}^m} \|\beta\mathbf{e}_1 - \tilde{\mathbf{H}}_m\mathbf{y}\|_2$ and $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{Z}_m\mathbf{y}_m$
14:     If satisfied Stop, else set $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ and GoTo 2.
15: end procedure
Output: $\mathbf{x}_m$
```

Here $\mathbf{M}_k$ denotes the preconditioner operator, which possibly varies at each iteration k and is used in algorithm (1) to approximate the solution of the system

$$\mathbf{A}\mathbf{c}_k = \mathbf{v}_k. \tag{11}$$

Should the linear system (11), which in our case is $\boldsymbol{\mu}-$dependent, be solved exactly, the FGMRES converges to the exact solution at iteration k.

3. RB methods for the parametrized Stokes equations

For the MSRB preconditioners we propose in this paper, a key ingredient is represented by the RB method, which is exploited as a coarse component in a two-level preconditioner. In the following, we will briefly recall the RB method for the parametrized Stokes equations. For a more extensive outlook on the subject, we refer to [33] for RB techniques for the parametrized Stokes equations and to [35, 36] for parametrized PDEs in general.

The RB method relies on the idea that the solution $\mathbf{z}_h^{\boldsymbol{\mu}}$ of the parametrized system (5), for a certain value of the parameter $\boldsymbol{\mu}$, can be well approximated as a linear combination of $N \ll N_h$ global, problem-dependent basis functions $\{\boldsymbol{\xi}_i\}_{i=1}^N$ obtained by orthonormalizing FE solutions of the same problem computed for selected values of the parameter. The basis functions are collected in a matrix $\mathbf{V} = [\boldsymbol{\xi}_1|\dots|\boldsymbol{\xi}_N] \in \mathbb{R}^{N_h \times N}$. The RB space, which is formally obtained by the span of the columns of $\mathbf{V}$, is usually built during an *offline* phase with a greedy algorithm or employing proper orthogonal decomposition (POD). Specifically, we use this latter approach. Once the RB space has been built, during the *online* phase the solution of the PDE for a new parameter $\boldsymbol{\mu}$ is computed by solving a RB system, instead of (5). The RB problem is constructed by introducing a test space represented by a matrix $\mathbf{W}^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N}$, generally different from $\mathbf{V}$ and possibly $\boldsymbol{\mu}$-dependent. If $\mathbf{W}^{\boldsymbol{\mu}} \neq \mathbf{V}$ we end up with a more general PG-RB problem, otherwise, if $\mathbf{W}^{\boldsymbol{\mu}} = \mathbf{V}$, we come up with a G-RB problem. Here, for the sake of generality, we consider the more general PG-RB problem, which leads to the following RB problem

$$\mathbf{A}_N^{\boldsymbol{\mu}} \mathbf{z}_N^{\boldsymbol{\mu}} = \mathbf{g}_N^{\boldsymbol{\mu}}. \tag{12}$$

The latter is a linear system 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

$$\mathbf{A}_N^{\boldsymbol{\mu}} = (\mathbf{W}^{\boldsymbol{\mu}})^T \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{V}, \qquad \mathbf{g}_N^{\boldsymbol{\mu}} = (\mathbf{W}^{\boldsymbol{\mu}})^T \mathbf{g}_h^{\boldsymbol{\mu}}, \tag{13}$$

respectively. Finally, the FE representation $\mathbf{V}\mathbf{z}_N^{\boldsymbol{\mu}}$ of the RB approximation is recovered as

$$\mathbf{V}\mathbf{z}_N^{\boldsymbol{\mu}} = \mathbf{V}(\mathbf{A}_N^{\boldsymbol{\mu}})^{-1}\mathbf{g}_N^{\boldsymbol{\mu}} = \mathbf{V}(\mathbf{A}_N^{\boldsymbol{\mu}})^{-1}(\mathbf{W}^{\boldsymbol{\mu}})^T \mathbf{g}_h^{\boldsymbol{\mu}}. \tag{14}$$

We highlight that problem (12) is obtained by enforcing the projection of the FE residual evaluated for the RB solution $\mathbf{V}\mathbf{z}_N^{\boldsymbol{\mu}}$ onto $\mathbf{W}^{\boldsymbol{\mu}}$ to vanish, that is by requiring

$$(\mathbf{W}^{\boldsymbol{\mu}})^T \Big(\mathbf{g}_h^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{V}\mathbf{z}_N^{\boldsymbol{\mu}}\Big) = 0. \tag{15}$$

In the Stokes case, the matrix $\mathbf{V}$ is such that

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{N_u} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{N_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_1^u & | & \dots & | & \boldsymbol{\varphi}_{N_u}^u & | & \mathbf{0} & | & \dots | & \mathbf{0} \\ \mathbf{0} & | & \dots & | & \mathbf{0} & | & \boldsymbol{\varphi}_1^p & | & \dots | & \boldsymbol{\varphi}_{N_p}^p \end{bmatrix}, \tag{16}$$

where $\mathbf{V}_{N_u} = \big[\boldsymbol{\varphi}_1^u|\dots|\boldsymbol{\varphi}_{N_u}^u\big] \in \mathbb{R}^{N_h^u \times N_u}$ and $\mathbf{V}_{N_p} = \Big[\boldsymbol{\varphi}_1^p|\dots|\boldsymbol{\varphi}_{N_p}^p\Big] \in \mathbb{R}^{N_h^p \times N_p}$ are specifically used to find an approximation for the velocity $\mathbf{u}_h^{\boldsymbol{\mu}}$ and the pressure $\mathbf{p}_h^{\boldsymbol{\mu}}$. The RB spaces are built from a set of snapshots $\big\{\mathbf{u}_h^{\boldsymbol{\mu}_i}\big\}_{i=1}^{n_s}$, $\big\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\big\}_{i=1}^{n_s}$ computed for different instances (properly sampled) of the parameters $\big\{\boldsymbol{\mu}_i\big\}_{i=1}^{n_s}$, by performing POD on the two sets of snapshots separately

$$\mathbf{V}_{N_u} = \mathrm{POD}\Big(\big\{\mathbf{u}_h^{\boldsymbol{\mu}_i}\big\}_{i=1}^{n_s}, \mathbf{X}_u, \varepsilon_{\mathrm{POD}}\Big), \qquad \mathbf{V}_{N_p} = \mathrm{POD}\Big(\big\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\big\}_{i=1}^{n_s}, \mathbf{X}_p, \varepsilon_{\mathrm{POD}}\Big).$$

Indeed, the vector spaces spanned by the columns of $\mathbf{V}_{N_u}$ (resp. $\mathbf{V}_{N_p}$) approximate up to a certain tolerance $\varepsilon_{\text{POD}} > 0$ the space spanned by the snapshots $\{\mathbf{u}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ (resp. $\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$). The matrices $\mathbf{V}_{N_u}$ and $\mathbf{V}_{N_p}$ are indeed constructed by selecting the largest $N_u = N_u(\varepsilon_{\text{POD}})$ and $N_p = N_p(\varepsilon_{\text{POD}})$ eigenmodes respectively, see [35]; a priori, $N_u \neq N_p$. 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 order of magnitudes; for this reason the RB system (12) is usually solved by direct methods.

Remark 3.1. *Instead of prescribing a tolerance to POD, one can provide as input the dimensions N_u and N_p. In this case, POD retrieves the first N_u (resp. N_p) modes, approximating the snapshots subspace up to a tolerance $\varepsilon^u_{\text{POD}} = \varepsilon^u_{\text{POD}}(N)$ (resp. $\varepsilon^p_{\text{POD}} = \varepsilon^p_{\text{POD}}(N)$).*

Remark 3.2. *POD computes an approximation space by minimizing the distance with respect to a prescribed norm. In our case we employ for velocity and pressure the norms induced by the matrices $\mathbf{X}_u$ and $\mathbf{X}_p$, respectively.*

An important issue concerns the stability of the resulting RB approximation, since a stable couple of reduced subspaces for velocity and pressure, fulfilling an equivalent *inf-sup* condition at the reduced level, must be used to ensure that the RB Stokes problem (12) is well-posed. More precisely, there must exist $\beta_N^{min} > 0$ such that

$$\beta_N^{\boldsymbol{\mu}} = \inf_{\mathbf{z}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T \mathbf{A}_N^{\boldsymbol{\mu}} \mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \|\mathbf{W}^{\boldsymbol{\mu}} \mathbf{w}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \beta_N^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{17}$$

This property is not automatically guaranteed if a G-RB method is used, that is, in the case where the RB problem is constructed by Galerkin projection onto an RB space made of orthonormalized solutions of (2) for different values of parameters. Therefore, different strategies have been designed to ensure the stability of the RB problem by fulfilling (17). One possibility consists in augmenting the velocity space by means of a set of "enriching" basis functions computed through the so-called pressure supremizing operator, leading to a reduced problem with roughly as twice as many velocity degrees of freedom compared to the pressure, see [40, 41] for the details. Another possibility to automatically build a stable RB problem exploits PG-RB methods [33, 34, 35], such as the least-squares (LS) method. The LS-RB method relies on a test space which is obtained as the image of the RB space through a global supremizer operator which involves both velocity and pressure fields. These strategies are detailed in AppendixA for the sake of completeness. In the following, the MSRB preconditioning method will be built by relying on either one of these options.

4. MSRB preconditioners

The MSRB preconditioning method has been firstly presented in [1] for elliptic parametrized problems; a numerical investigation on parametrized advection-diffusion PDEs has been carried out in [37]. The goal of this technique is to build a preconditioner to efficiently solve parametrized linear systems which arise from the FE discretization of parameter-dependent PDEs. Computational efficiency is pursued by combining multiplicatively a nonsingular fine grid preconditioner $\mathbf{P}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ with an efficient coarse correction built upon the RB method, leading to a two-level preconditioning method. Following the strategy introduced in [1], we define the MSRB preconditioner as

$$\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k} = (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1} + \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big), \qquad k = 1, 2, \ldots, \tag{18}$$

where $\mathbf{Q}^{\boldsymbol{\mu}}_{N_k}$ is the iteration- (k-) dependent RB coarse component. The preconditioner $\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}$ is used at iteration k, and yields a coarse correction tailored for the error equation (11) and corresponding to the $k-$th iteration. In particular, $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}}$ is an RB solver which is trained on the following equation

$$\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{y}_k^{\boldsymbol{\mu}} = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big)\mathbf{v}_k^{\boldsymbol{\mu}}, \qquad k = 1, 2, \dots, \tag{19}$$

where $\mathbf{v}_k^{\boldsymbol{\mu}}$ is the k-th Krylov basis; as a matter of fact, $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}}$ provides an accurate approximation of the solution of the parametrized linear system (19). By using an accurate RB coarse component, the FGMRES algorithm converges in only few iterations. To ease the notation, in the following we denote

$$\mathbf{v}_{k+\frac{1}{2}}^{\boldsymbol{\mu}} = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big)\mathbf{v}_k^{\boldsymbol{\mu}}.$$

As mentioned above, the core contribution of this work lies in extending the MSRB preconditioning framework to saddle-point problems, and in particular the Stokes equations. To this aim, we keep the structure of the preconditioner as in (18) and train the RB coarse components to accurately solve the error equation (19). We remark that $\mathbf{A}_h^{\boldsymbol{\mu}}$ is the FE saddle-point matrix of the Stokes system, hence the RB coarse components must comply with the saddle-point structure of equation (19) and cannot be constructed as in [1]. In the next section, the following aspects are investigated to build the MSRB preconditioner for the Stokes equations.

a) The *costruction* of the MSRB preconditioner: the RB coarse components employed in [1] are obtained by relying on a RB solver for equation (19) obtained by Galerkin projection of the FE arrays (matrix and right hand side) onto the RB spaces. This procedure cannot be employed when dealing with a saddle-point system, because it would lead to a RB solver for the error equation which includes a singular RB matrix. This problem is overcome by introducing a new RB coarse component obtained by either enriched G-RB method (that is by enriching the velocity space with supremizer functions and by employing a Galerkin projection) or a PG-RB method. This is done in Subsections 5.1 and 5.2.

b) The *well-posedness* of the MSRB preconditioner constructed as explained at point a) is then studied. If the enriched G-RB method is employed to construct the RB coarse components, then the results in [1] hold, ensuring the nonsingularity of the MSRB preconditioner. On the other hand, if a PG-RB method is employed the results in [1] do not hold. Consequently, in Subsection 5.3 this latter case is investigated.

c) The *practical implementation* of the MSRB preconditioner for Stokes equations also needs to be revised, since its construction requires either to compute the supremizer functions (if the enriched Galerkin approach is employed) or to perform a projection using a PG-RB approach, in the case this latter formulation is used.

Notice that, as in [1], we opt here for a multiplicative combination of fine and coarse components, however different combinations of $\mathbf{P}_h^{\boldsymbol{\mu}}$ and $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}}$, e.g. additive or symmetric, are also possible.

5. MSRB preconditioners for the Stokes equations

This section is devoted to build a MSRB preconditioner when dealing with parametrized Stokes equations (4) and analyze its well-posedness. To set up the MSRB preconditioner for the Stokes equations in a fairly general way, we consider the PG-RB method to build the k-dependent coarse components $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}}$. To this aim, we introduce the matrices $\mathbf{V}_k \in \mathbb{R}^{N_h \times N_k}$, $k = 1, 2, \dots$ such that

$$\mathbf{V}_k = \left[\boldsymbol{\xi}_1^k | \dots | \boldsymbol{\xi}_N^k\right],$$

where the basis $\{\boldsymbol{\xi}_i^k\}_i^{N_k}$ is tailored to provide a RB approximation $\mathbf{y}_{N_k}^{\boldsymbol{\mu}}$ to the solution $\mathbf{y}_k^{\boldsymbol{\mu}}$ of problem (19); here $k = 1, 2, \dots$ is the iteration counter of the FGMRES method and N_k, $k = 1, 2, \dots$ is the dimension of the k-th RB space. We remark that the RB coarse component for the MSRB preconditioner is obtained, similarly to (15), by enforcing the projection of the FE residual of (19) evaluated for the RB coarse correction $\mathbf{V}_k\mathbf{y}_{N_k}^{\boldsymbol{\mu}}$ onto $\mathbf{W}_k^{\boldsymbol{\mu}}$ to vanish, that is by requiring

$$(\mathbf{W}_k^{\boldsymbol{\mu}})^T\Big(\mathbf{v}_{k+\frac{1}{2}}^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}\mathbf{y}_{N_k}^{\boldsymbol{\mu}}\Big) = 0. \tag{20}$$

Notice that $\mathbf{W}_k^{\boldsymbol{\mu}}$ depends on both k and $\boldsymbol{\mu}$. If $\mathbf{W}_k^{\boldsymbol{\mu}} \neq \mathbf{V}_k$, we build a PG-RB coarse correction; whereas, by choosing $\mathbf{W}_k^{\boldsymbol{\mu}} = \mathbf{V}_k$, we employ a G-RB coarse correction. We then obtain the following RB problem, to be solved at iteration k, for any $\boldsymbol{\mu}$

$$(\mathbf{W}_k^{\boldsymbol{\mu}})^T\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}_k\mathbf{y}_{N_k}^{\boldsymbol{\mu}} = (\mathbf{W}_k^{\boldsymbol{\mu}})^T\Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big)\mathbf{v}_k^{\boldsymbol{\mu}}, \qquad k = 1, 2, \dots, \tag{21}$$

whose solution $\mathbf{y}_{N_k}^{\boldsymbol{\mu}} \in \mathbb{R}^{N_k}$ is the RB approximation of the solution $\mathbf{y}_k^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h}$ of (19). Accordingly with the construction in Section 3, the RB matrices $\mathbf{A}_{N_k}^{\boldsymbol{\mu}} \in \mathbb{R}^{N_k \times N_k}$, $k = 1, 2, \dots$ are built as

$$\mathbf{A}_{N_k}^{\boldsymbol{\mu}} = (\mathbf{W}_k^{\boldsymbol{\mu}})^T\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}_k. \tag{22}$$

The FE representation $\mathbf{V}_k\mathbf{y}_{N_k}^{\boldsymbol{\mu}}$ of the RB approximation is then recovered as in equation (14), that is

$$\mathbf{V}_k\mathbf{y}_{N_k}^{\boldsymbol{\mu}} = \mathbf{V}_k(\mathbf{A}_{N_k}^{\boldsymbol{\mu}})^{-1}(\mathbf{W}_k^{\boldsymbol{\mu}})^T\Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big)\mathbf{v}_k^{\boldsymbol{\mu}},$$

from which we set the coarse correction as $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}} = \mathbf{V}_k(\mathbf{A}_{N_k}^{\boldsymbol{\mu}})^{-1}(\mathbf{W}_k^{\boldsymbol{\mu}})^T$.

In the case of the parametrized Stokes equations, the solution of equation (19) is made of both velocity and pressure components, that is, $\mathbf{y}_k^{\boldsymbol{\mu}} = [(\mathbf{y}_{uk}^{\boldsymbol{\mu}})^T, (\mathbf{y}_{pk}^{\boldsymbol{\mu}})^T]^T$, $k = 1, \dots$. Consequently, we build the RB spaces for these two variables separately by setting

$$\mathbf{V}_{N_k^u}^u = \mathrm{POD}\Big(\{\mathbf{y}_{uk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}, \mathbf{X}_u, \delta_{RB,k}\Big), \qquad \mathbf{V}_{N_k^p}^p = \mathrm{POD}\Big(\{\mathbf{y}_{pk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}, \mathbf{X}_p, \delta_{RB,k}\Big), \tag{23}$$

where $\delta_{RB,k} > 0$ is a prescribed tolerance (possibly dependeing on k). Here $\{\mathbf{y}_{uk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and $\{\mathbf{y}_{pk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ are error snapshots for the velocity and the pressure, respectively, such that $\mathbf{y}_k^{\boldsymbol{\mu}} = [\mathbf{y}_{uk}^{\boldsymbol{\mu}}, \mathbf{y}_{pk}^{\boldsymbol{\mu}}]^T$ is the solution of problem (19), for properly chosen instances of the parameters. Notice that POD on velocities $\{\mathbf{y}_{uk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, $k = 1, \dots$ is performed with respect to the scalar product induced by the norm matrix $\mathbf{X}_u$. On the other hand, POD on pressures $\{\mathbf{y}_{pk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ is performed with respect to the scalar product induced by the norm matrix $\mathbf{X}_p$. Finally, the matrix $\mathbf{V}_k$ has the following form

$$\mathbf{V}_k = \begin{bmatrix} \mathbf{V}_{N_k^u}^u & 0 \\ 0 & \mathbf{V}_{N_k^p}^p \end{bmatrix}. \tag{24}$$

Remark 5.1. *An inf-sup condition similar to* (17) *must hold in order to guarantee the nonsingularity of the matrices* $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$ *for* $k = 1, 2, \dots$, *that is, for any* $k = 1, 2, \dots$ *there must exist* $\beta_{N_k}^{min} > 0$ *such that*

$$\beta_{N_k}^{\boldsymbol{\mu}} = \inf_{\mathbf{z}_N \in \mathbb{R}^N} \sup_{\mathbf{w}_N \in \mathbb{R}^N} \frac{\mathbf{w}_N^T\mathbf{A}_{N_k}^{\boldsymbol{\mu}}\mathbf{z}_N}{\|\mathbf{V}\mathbf{z}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}\|\mathbf{W}_k^{\boldsymbol{\mu}}\mathbf{w}_N\|_{\mathbf{X}_h^{\boldsymbol{\mu}}}} \geq \beta_{N_k}^{min} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{25}$$

Remark 5.2. *Instead of providing the tolerances* $\delta_{RB,k}$, *we could prescribe the dimensions* N_k^u *and* N_k^p

of the RB spaces for the velocity and the pressure, respectively, at each iteration.

In the following we devise two alternative techniques to build a well-posed RB coarse correction, according to two different choices of $\mathbf{W}_k^{\boldsymbol{\mu}}$, $k = 1, 2 \dots$. These two options reflect the choice between a G-RB or an algebraic LS-RB method.

5.1. MSRB preconditioners with enriched G-RB coarse corrections

A G-RB approximation to build the $k-$th coarse correction is obtained by choosing $\mathbf{W}_k^{\boldsymbol{\mu}} = \mathbf{V}_k$, $k = 1, 2, \dots$. However, the resulting RB approximation is not guaranteed to fulfill (25). Consequently, we consider an *enriched velocity space formulation*, where the velocity space spanned by the columns of $\mathbf{V}_{N_k^u}^u$ is augmented by a set of N_k^s enriching basis functions. Given the pressure snapshots $\{\mathbf{y}_{pk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, we build the *pressure supremizing snapshots* by solving the following problems

$$\mathbf{X}_u^{\boldsymbol{\mu}} \mathbf{y}_{sk}^{\boldsymbol{\mu}_i} = (\mathbf{B}_h^{\boldsymbol{\mu}_i})^T \mathbf{y}_{pk}^{\boldsymbol{\mu}_i} \qquad i = 1, \dots, n_s. \tag{26}$$

Next, we run POD on the set of pressure supremized snapshots $\{\mathbf{y}_{sk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and obtain $\mathbf{V}_{N_k^s}^s \in \mathbb{R}^{N_h \times N_k^s}$ as

$$\mathbf{V}_{N_k^s}^s = \mathrm{POD}\Big(\{\mathbf{y}_{sk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}, \mathbf{X}_u, \varepsilon_{\mathrm{POD}}\Big).$$

The columns of $\mathbf{V}_{N_k^s}^s$ form a N_k^s-dimensional space employed to augment the velocity space. We introduce

$$\tilde{\mathbf{V}}_k = \begin{bmatrix} \mathbf{V}_{N_k^u}^u & \mathbf{V}_{N_k^s}^s & 0 \\ 0 & 0 & \mathbf{V}_{N_k^p}^p \end{bmatrix}, \qquad k = 1, 2, \dots,$$

and obtain a well-posed G-RB coarse correction by choosing $\mathbf{W}_k^{\boldsymbol{\mu}} = \mathbf{V}_k = \tilde{\mathbf{V}}_k$, $k = 1, \dots$, in (22), leading to the following definition

$$\mathbf{A}_{N_k}^{\boldsymbol{\mu}} = \tilde{\mathbf{V}}_k^T \mathbf{A}_h^{\boldsymbol{\mu}} \tilde{\mathbf{V}}_k, \qquad k = 1, \dots.$$

Notice that a velocity enrichment is required for every coarse correction, leading to solve n_s additional problems of the form of (26) for each coarse correction $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}}$, $k = 1, 2, \dots$ which has to be built. The velocity enrichment allows to obtain a couple of RB spaces which proves to be numerically stable, even though a rigorous stability result cannot be proven, see e.g. [41, 33].

5.2. MSRB preconditioners with PG-RB coarse corrections

A purely algebraic PG-RB method, recently proposed in [33], yields a stable RB approximation to problem (19). This method can be viewed as an algebraic least-squares RB (we call it aLS-RB) method for parametrized noncoercive problems as (5). Compared to the approximate enrichment of the velocity space described in Section 5.1, the aLS-RB method features a smaller dimension of the RB spaces (i.e. the number of RB functions is lower), since in this case the velocity space is not augmented. This yields a remarkable advantage when the RB coarse corrections and the inverse matrices of $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$, $k = 1, 2, \dots$, are constructed for a new parameter. Furthermore, the resulting RB formulation is automatically *inf-sup* stable, i.e. (25) is fulfilled.

To build an aLSRB approximation, we introduce a symmetric and positive definite matrix $\mathbf{P}_X \in \mathbb{R}^{N_h \times N_h}$, and we assume the existence of two positive constants $C \geq c$ such that

$$c\|\mathbf{x}\|_{\mathbf{P}_X} \leq \|\mathbf{x}\|_{\mathbf{X}_h^{\boldsymbol{\mu}}} \leq C\|\mathbf{x}\|_{\mathbf{P}_X} \qquad \forall \mathbf{x} \in \mathbb{R}^{N_h}. \tag{27}$$

The aLS-RB coarse correction is constructed by selecting $\mathbf{W}_k^{\boldsymbol{\mu}}$ as $\mathbf{W}_k^{\boldsymbol{\mu}} = \mathbf{P}_X^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}_k$ in (22), leading to the following definition

$$\mathbf{A}_{N_k}^{\boldsymbol{\mu}} = \mathbf{V}_k^T(\mathbf{A}_h^{\boldsymbol{\mu}})^T\mathbf{P}_X^{-1}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}_k, \qquad k = 1, \dots. \tag{28}$$

In our numerical experiments, $\mathbf{P}_X$ is chosen as $\mathbf{P}_X = \mathbf{X}_h^0$, i.e. as the norm matrix in the reference domain, or as a block diagonal preconditioner $\mathbf{P}_{\mathbf{X}_h^0} \in \mathbb{R}^{N_h \times N_h}$ 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 $\mathbf{X}_p^0$, respectively.

Remark 5.3. *The standard LSRB method relies on formulation* (28) *where the matrix* $\mathbf{X}_h^{\boldsymbol{\mu}}$ *plays the role of* $\mathbf{P}_X$*. However, when the computational domain depends on the parameter, and especially when the mapping from* Ω^0 *to* $\Omega^{\boldsymbol{\mu}}$ *is not known a priori, the* $\boldsymbol{\mu}-$*dependence of* $\mathbf{X}_h^{\boldsymbol{\mu}}$ *could lead to huge assembling costs for* $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$*. On the other hand, by choosing a* $\boldsymbol{\mu}-$*independent matrix* $\mathbf{P}_X$*, this overhead is no longer there.*

5.3. Nonsingularity of the preconditioner

When a G-RB approximation is employed to build the coarse corrections, as in the case where an augmented velocity space is used, the MSRB preconditioner operator $\mathbf{Q}_{\mathrm{MSRB},k}^{\boldsymbol{\mu}}$ has been shown to be invertible, with proper assumptions on $\mathbf{P}_h^{\boldsymbol{\mu}}$ and the basis $\mathbf{V}_k$, in [1]. In this section we extend these results, showing that $\mathbf{Q}_{\mathrm{MSRB},k}^{\boldsymbol{\mu}}$ is invertible when a more general PG-RB approach is used to build the RB coarse correction, as in section 5.2.

Let $W_1 = span\{\mathbf{w}_j^1\}_{j=1}^M$ and $W_2 = span\{\mathbf{w}_j^2\}_{j=1}^M \subset \mathbb{R}^{N_h}$ be two subspaces such that $dim(W_1) = dim(W_2) = M$. We denote by $W_1^\perp$ and $W_2^\perp$ the orthogonal complement of W_1 and W_2, respectively, and by $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{N_h \times M}$ the matrices of basis vectors such that $\mathbf{W}_1 = [\mathbf{w}_1^1, \dots, \mathbf{w}_M^1]$, $\mathbf{W}_2 = [\mathbf{w}_1^2, \dots, \mathbf{w}_M^2]$. Moreover, given a subspace $W \subset \mathbb{R}^{N_h}$ and a nonsingular matrix $\mathbf{B} \in \mathbb{R}^{N_h \times N_h}$, we define the following spaces

$$\mathbf{B}W = \Big\{\mathbf{x} \in \mathbb{R}^{N_h} : \ \mathbf{B}^{-1}\mathbf{x} \in W \ \ \Big\} = \Big\{\mathbf{x} \in \mathbb{R}^{N_h} : \ \ \mathbf{x} = \mathbf{B}\mathbf{z}, \, \mathbf{z} \in W\Big\},$$
$$\mathbf{B}W^\perp = \Big\{\mathbf{x} \in \mathbb{R}^{N_h} : \ \mathbf{B}^{-1}\mathbf{x} \in W^\perp\Big\} = \Big\{\mathbf{x} \in \mathbb{R}^{N_h} : \ \ \mathbf{x} = \mathbf{B}\mathbf{z}, \, \mathbf{z} \in W^\perp\Big\}.$$

We remark that $\mathbb{R}^{N_h} = \mathbf{B}W \oplus \mathbf{B}W^\perp$, because of the nonsingularity of $\mathbf{B}$.

Lemma 5.1. *Let* W_1 *and* W_2 *be two* M*-dimensional subspaces of* $\mathbb{R}^{N_h}$*,* $\{\mathbf{w}_j^1\}_{j=1}^M$ *and* $\{\mathbf{w}_j^2\}_{j=1}^M$ *their basis and* $\mathbf{W}_1 = [\mathbf{w}_1^1, \dots, \mathbf{w}_M^1] \in \mathbb{R}^{N_h \times M}$*,* $\mathbf{W}_2 = [\mathbf{w}_1^2, \dots, \mathbf{w}_M^2] \in \mathbb{R}^{N_h \times M}$*. Moreover, let* $\mathbf{B}$ *be a nonsingular* $N_h \times N_h$ *matrix and assume that* $\mathbf{W}_2^T\mathbf{B}\mathbf{W}_1$ *is nonsingular. Then the following implication holds:*

$$\mathbf{x} \in \mathbf{B}W_1 \quad \textit{and} \quad \mathbf{W}_2^T\mathbf{x} = \mathbf{0} \quad \Rightarrow \quad \mathbf{x} = \mathbf{0}.$$

Proof. We take $\mathbf{x} \in \mathbf{B}W_1$ such that $\mathbf{W}_2^T\mathbf{x} = \mathbf{0}$ and show that it must be $\mathbf{x} = \mathbf{0}$. By definition of $\mathbf{B}W_1$, $\mathbf{B}^{-1}\mathbf{x} = \mathbf{W}_1\mathbf{z}_M$ for some $\mathbf{z}_M \in \mathbb{R}^M$. Thanks to the nonsingularity of $\mathbf{B}$, we obtain

$$\mathbf{0} = \mathbf{W}_2^T\mathbf{x} = \mathbf{W}_2^T\mathbf{B}\mathbf{B}^{-1}\mathbf{x} = \mathbf{W}_2^T\mathbf{B}\mathbf{W}_1\mathbf{z}_M,$$

which implies $\mathbf{z}_M = \mathbf{0}$, due to the nonsingularity of $\mathbf{W}_2^T\mathbf{B}\mathbf{W}_1 \in \mathbb{R}^{M \times M}$. Finally, we have

$$\mathbf{0} = \mathbf{W}_1\mathbf{z}_M = \mathbf{B}^{-1}\mathbf{x},$$

which, thanks to the nonsingularity of $\mathbf{B}$, ends the proof. □

In the following we employ Lemma 5.1 by taking $\mathbf{W}_1 = \mathbf{V}_k$, $\mathbf{W}_2 = \mathbf{W}_k^{\boldsymbol{\mu}}$, $\mathbf{B} = \mathbf{P}_h^{\boldsymbol{\mu}}$ in order to prove that $\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}$ is nonsingular. To this aim, we define

$$V_{N_k}^{\mathbf{P}_h /\!/} = \Big\{ \mathbf{x} \in \mathbb{R}^{N_h} : \quad (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x} \in V_{N_k} \Big\}, \qquad V_{N_k}^{\mathbf{P}_h \perp} = \Big\{ \mathbf{x} \in \mathbb{R}^{N_h} : \quad (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x} \in V_{N_k}^{\perp} \Big\}.$$

Theorem 5.1. *For any $\boldsymbol{\mu} \in \mathcal{D}$, assume that $\mathbf{P}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h \times N_h}$ is a nonsingular matrix such that the matrix $(\mathbf{W}_k^{\boldsymbol{\mu}})^T \mathbf{P}_h^{\boldsymbol{\mu}} \mathbf{V}_k$ is nonsingular. Then the matrix $\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}$ is nonsingular.*

Proof. The proof is similar to the one outlined in [1]. Given $\mathbf{x} = \mathbf{x}_{/\!/} + \mathbf{x}_{\perp}$, where $\mathbf{x}_{/\!/} \in V_{N_k}^{\mathbf{P}_h /\!/}$, $\mathbf{x}_{\perp} \in V_{N_k}^{\mathbf{P}_h \perp}$, such that $\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}\mathbf{x} = \mathbf{0}$, then it must be $\mathbf{x} = \mathbf{0}$. Then we have

$$\begin{aligned} \mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}\mathbf{x}_{/\!/} &= (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x}_{/\!/} + \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big)\mathbf{x}_{/\!/} \\ &= \mathbf{V}_k \mathbf{z}_N^{\boldsymbol{\mu}} + \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\mathbf{x}_{/\!/} - \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{V}_k\mathbf{z}_N^{\boldsymbol{\mu}} = \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\mathbf{x}_{/\!/}, \end{aligned}$$

where $(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x}_{/\!/} = \mathbf{V}_k\mathbf{z}_N^{\boldsymbol{\mu}}$ for some $\mathbf{z}_{N_k}^{\boldsymbol{\mu}} \in \mathbb{R}^{N_k}$. Then

$$\begin{aligned} \mathbf{0} = \mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}\mathbf{x} &= \mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}\mathbf{x}_{/\!/} + \mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}\mathbf{x}_{\perp} \\ &= \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\mathbf{x}_{/\!/} + (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x}_{\perp} + \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\Big(\mathbf{I}_{N_h} - \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\Big)\mathbf{x}_{\perp} \end{aligned}$$

which leads to

$$\mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\Big(\mathbf{x}_{/\!/} + \mathbf{x}_{\perp} + \mathbf{A}_h^{\boldsymbol{\mu}}(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x}_{\perp}\Big) = -(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x}_{\perp}. \tag{29}$$

The left hand side is an element of V_{N_k}, the right hand side is an element of $V_{N_k}^{\perp}$, therefore the only way for them to be equal is when they are both zero. Being $(\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{x}_{\perp} = \mathbf{0}$, implies $\mathbf{x}_{\perp} = \mathbf{0}$ thanks to the nonsingularity of $\mathbf{P}_h^{\boldsymbol{\mu}}$, leading to

$$\mathbf{0} = \mathbf{Q}^{\boldsymbol{\mu}}_{N_k}\mathbf{x}_{/\!/} = \mathbf{V}_k(\mathbf{A}^{\boldsymbol{\mu}}_{N_k})^{-1}(\mathbf{W}_k^{\boldsymbol{\mu}})^T\mathbf{x}_{/\!/} \tag{30}$$

which, thanks to linear independence of the columns of $\mathbf{V}_k$ and the non singularity of $\mathbf{A}^{\boldsymbol{\mu}}_{N_k}$ yields

$$(\mathbf{W}_k^{\boldsymbol{\mu}})^T\mathbf{x}_{/\!/} = \mathbf{0}.$$

Finally, by applying Lemma 5.1 with $W_1 = V_{N_k}$, $\mathbf{W}_1 = \mathbf{V}_k$, $\mathbf{W}_2 = \mathbf{W}_k^{\boldsymbol{\mu}}$ and $\mathbf{B} = \mathbf{P}_h^{\boldsymbol{\mu}}$, we obtain that $\mathbf{x}_{/\!/} = \mathbf{0}$. □

Being the matrix $\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k}$ invertible, we can define the MSRB preconditioner as

$$\mathbf{P}^{\boldsymbol{\mu}}_{\text{MSRB},k} = (\mathbf{Q}^{\boldsymbol{\mu}}_{\text{MSRB},k})^{-1}.$$

6. Algorithmic procedure

In this section we detail the procedures required to build and use the MSRB preconditioner, by splitting the computation in an *offline* (typically expensive) and an *online* phase, where the FE problem (5) is solved for a new instance of $\boldsymbol{\mu}$.

6.1. Offline phase

During the offline phase, we build the structures required by (18) to handle any new possible instance of the parameter online, namely the RB spaces $\mathbf{V}_k$, $k = 1, 2, \ldots$ and the corresponding coarse corrections.

6.1.1. Building the RB spaces

In order to build the RB spaces as in (23), we first solve the FE problem (5) for n_s instances of $\boldsymbol{\mu}$ to build the snapshots for velocity $\{\mathbf{u}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and pressure $\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, and set

$$\mathbf{y}_{u0}^{\boldsymbol{\mu}_i} = \mathbf{u}_h^{\boldsymbol{\mu}_i}, \quad \mathbf{y}_{p0}^{\boldsymbol{\mu}_i} = \mathbf{p}_h^{\boldsymbol{\mu}_i}, \qquad i = 1, \dots, n_s.$$

The sets of snapshots $\{\mathbf{y}_{u0}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and $\{\mathbf{y}_{p0}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ are used to build the spaces $\mathbf{V}_{N_0^u}^u$ and $\mathbf{V}_{N_0^p}^p$, respectively. These are used to provide the initial guess to the FGMRES algorithm. For each new RB space $\mathbf{V}_k$, $k = 1, 2, \dots$, the new snapshots $\{\mathbf{y}_{uk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and $\{\mathbf{y}_{pk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, $k = 1, 2, \dots$, solution of (19) for particular instances of $\boldsymbol{\mu}$, need to be considered. An option to compute them is to solve problem (19), for any k and for each snapshot; this is however impractical, especially when the dimension N_h of the FE problem largely increases. On the other hand, one can alternatively take advantage of the following relations

$$\begin{cases} \gamma^{\mu} = \|\mathbf{g}_h^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{z}_h^0\|_2, \\ \mathbf{y}_1^{\boldsymbol{\mu}} = \frac{1}{\gamma^{\mu}}(\mathbf{z}_h^{\boldsymbol{\mu}} - \mathbf{z}_h^0) - (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{v}_1^{\boldsymbol{\mu}}, \\ \mathbf{z}_k^{\boldsymbol{\mu}} = \mathbf{Q}_{\mathrm{MSRB},k}^{\boldsymbol{\mu}} \mathbf{v}_k \quad \mathbf{w}^{\boldsymbol{\mu}} = \mathbf{A}\mathbf{z}_k^{\boldsymbol{\mu}} \\ h_{j,k}^{\boldsymbol{\mu}} = (\mathbf{w}^{\boldsymbol{\mu}}, \mathbf{v}_j^{\boldsymbol{\mu}}), \ \ \mathbf{w}^{\boldsymbol{\mu}} = \mathbf{w}^{\boldsymbol{\mu}} - h_{j,k}^{\boldsymbol{\mu}}\mathbf{v}_j^{\boldsymbol{\mu}} \qquad j = 1, \dots, k \\ h_{k+1,k}^{\boldsymbol{\mu}} = \|\mathbf{w}^{\boldsymbol{\mu}}\| \\ \mathbf{y}_k^{\boldsymbol{\mu}} = \frac{1}{h_{k,k-1}^{\boldsymbol{\mu}}}\Big[\mathbf{z}_k^{\boldsymbol{\mu}} - \sum_{j=1}^{k-1} h_{j,k-1}^{\boldsymbol{\mu}}\big(\mathbf{y}_j^{\boldsymbol{\mu}} + (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{v}_j^{\boldsymbol{\mu}}\big)\Big] - (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{v}_k^{\boldsymbol{\mu}}, \quad k \geq 2, \end{cases} \tag{31}$$

which do not involve the solution of any other FE linear system and hold by construction when FGMRES is employed and started with $\mathbf{z}_h^0$ as the initial guess, see [1].

6.1.2. Assembling the RB coarse corrections

When building the RB coarse correction for the k-th iteration of FGMRES and for a new instance of the parameter, the matrix $\mathbf{Q}_{N_k}^{\boldsymbol{\mu}}$ is not explicitly assembled; indeed, $\mathbf{W}_k^{\boldsymbol{\mu}}$, $(\mathbf{A}_{N_k}^{\boldsymbol{\mu}})^{-1}$, which is computed and stored as LU factorization of $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$ and $\mathbf{V}_k$ are applied consecutively to the right hand side of (19).

Here, we specifically focus on the construction of the matrix $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$, a task which would normally require to project the matrix $\mathbf{A}_h^{\boldsymbol{\mu}}$ as in (22). To avoid this operation, we require the FE matrix $\mathbf{A}_h^{\boldsymbol{\mu}}$ to feature an affine parameter dependence, that is,

$$\mathbf{A}_h^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_h^q, \tag{32}$$

where $\Theta_a^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_a$ are $\boldsymbol{\mu}$-dependent functions, and the matrices $\mathbf{A}_h^q \in \mathbb{R}^{N_h \times N_h}$ are $\boldsymbol{\mu}$-independent. If assumption (32) is verified, then the RB matrix $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$ in the G-RB case can be constructed as

$$\mathbf{A}_{N_k}^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \tilde{\mathbf{V}}_k^T \mathbf{A}_h^q \tilde{\mathbf{V}}_k = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_{N_k}^q. \tag{33}$$

On the other hand, in the aLS-RB case, the RB matrix can be built as

$$\mathbf{A}_{N_k}^{\boldsymbol{\mu}} = \sum_{q_1,q_2=1}^{Q_a} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_a^{q_2}(\boldsymbol{\mu}) \mathbf{V}_k^T (\mathbf{A}_h^{q_1})^T \mathbf{P}_X^{-1} \mathbf{A}_h^{q_2} \mathbf{V}_k = \sum_{q_1,q_2=1}^{Q_a} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_a^{q_2}(\boldsymbol{\mu}) \mathbf{A}_{N_k}^{q_1,q_2}. \tag{34}$$

The matrices $\mathbf{A}^q_{N_k}$, $q = 1, \dots, Q_a$, $\mathbf{A}^{q_1,q_2}_{N_k} \in \mathbb{R}^{N \times N}$, $q_1, q_2 = 1, \dots, Q_a$, depending on the chosen RB approximation, can be precomputed and stored once the RB spaces $\mathbf{V}_k$ (and $\tilde{\mathbf{V}}_k$ in the G-RB case) are constructed. Then, given a new value $\boldsymbol{\mu}$ of parameter, only the sum in (34) or (33) must be carried out to build $\mathbf{A}^{\boldsymbol{\mu}}_{N_k}$. Notice that an affine decomposition as (32) is hard to be found as a built-in property of the original $\boldsymbol{\mu}$-dependent problem. For instance, in the numerical results shown in this work, the computational domain depends nonaffinely on the parameter $\boldsymbol{\mu}$, because of the geometrical nature of the parametrization. Therefore, we rely on EIM or its discrete variant suited for sparse matrices Matrix-Discrete-EIM (MDEIM), see [42, 43]. To recover an *approximate* affine decomposition, such that the relation (32) is satisfied up to a certain tolerance δ_{mdeim} provided to the MDEIM algorithm:

$$\mathbf{A}^{\boldsymbol{\mu}}_h \approx \sum_{q=1}^{Q_a} \tilde{\Theta}^q_a(\boldsymbol{\mu}) \mathbf{A}^q_h, \tag{35}$$

where Q_a is the number of selected basis computed by MDEIM. Once a new value of $\boldsymbol{\mu}$ is considered, the coefficients $\tilde{\Theta}^q_a : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_a$ are computed by solving an interpolation problem. In practice, we run separately MDEIM on the matrices $\mathbf{D}^{\boldsymbol{\mu}}_h$ and $\mathbf{B}^{\boldsymbol{\mu}}_h$, meaning that the following relations hold

$$\mathbf{D}^{\boldsymbol{\mu}}_h \approx \sum_{q=1}^{Q_d} \tilde{\Theta}^q_d(\boldsymbol{\mu}) \mathbf{D}^q_h, \qquad \mathbf{B}^{\boldsymbol{\mu}}_h \approx \sum_{q=1}^{Q_b} \tilde{\Theta}^q_b(\boldsymbol{\mu}) \mathbf{B}^q_h, \tag{36}$$

where the functions $\tilde{\Theta}^q_d : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_d$ and $\tilde{\Theta}^q_b : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_b$ are $\boldsymbol{\mu}-$dependent and the matrices $\mathbf{D}^q_h \in \mathbb{R}^{N^u_h \times N^u_h}$, $q = 1, \dots, Q_d$ and $\mathbf{B}^q_h \in \mathbb{R}^{N^p_h \times N^u_h}$, $q = 1, \dots, Q_b$ are $\boldsymbol{\mu}$-independent.

The standard RB method detailed in Section 3 also exploits the affine parameter dependence (32) of the matrix $\mathbf{A}^{\boldsymbol{\mu}}_h$, or an approximated one as in (36), to boost its efficiency. In addition, it similarly employs the affine dependence property of the FE right hand side $\mathbf{g}^{\boldsymbol{\mu}}_h$, and if such an assumption is not met, it can be recovered approximately with EIM or DEIM [42, 44]. The accuracy of the resulting RB solution is significantly affected by the accuracy of the affine decomposition of $\mathbf{A}^{\boldsymbol{\mu}}_h$ and $\mathbf{g}^{\boldsymbol{\mu}}_h$, which is known to be a bottleneck for the efficiency of the RB approximation. See AppendixA.3 for further details.

Furthermore we remark that, when the MSRB preconditioning strategy is employed, an affine decomposition of $\mathbf{A}^{\boldsymbol{\mu}}_h$ is not strictly required; however, in the case it is available, it proves to be useful to cut the (possibly large) costs entailed by building the RB matrices by projection through (22). On the other hand, the affine decomposition of $\mathbf{g}^{\boldsymbol{\mu}}_h$ is not needed in any case: in opposition with the classic RB method, with the MSRB preconditioning strategy we aim at solving the full FE problem exploiting directly the FE right hand side and the FE residual, in other words, the FE problem is not substituted online with a smaller problem as in the standard RB case.

6.1.3. *Offline algorithms*

The offline construction of the MSRB preconditioner is outlined in algorithm 2 for the G-RB case and in algorithm 3 for the aLS-RB case. We provide a set of snapshots parameter $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$, a final tolerance ε_r and the tolerances to construct each RB space $\{\delta_{RB,k}\}_k$; then, at first we compute an affine decomposition $\{\mathbf{A}^q_h\}_{q=1}^{Q_a}$ of the matrix $\mathbf{A}^{\boldsymbol{\mu}}_h$ with MDEIM algorithm [43] (step 2), and we construct the snapshots required to build the first space (step 3). Then, we iteratively build the necessary RB spaces through POD (steps 5-8) and the affine RB decomposition matrices $\{\mathbf{A}^{q_1,q_2}_{N_k}\}_{q_1,q_2=1}^{Q_a}$ (step 9). The final number of RB spaces constructed is L. In the G-RB case, the construction of the snapshots is more demanding, since it requires to build also the supremizer snapshots and an additional POD for each RB space, which also leads to RB coarse components of larger dimension due to the enrichment of the velocity space. However, the number of affine structures to be computed and stored is Q_a in the G-RB case, but increases to Q_a^2 in the aLS-RB case.

Algorithm 2 MSRB Preconditioner with G-RB coarse correction - Offline phase

```
procedure MSRB-PRECONDITIONER-G-RB-OFFLINE($\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}, \varepsilon_r, \{\delta_{RB,k}\}_k, \delta_{\mathrm{MDEIM}}$)
    Compute an affine approximation $\{\mathbf{A}_h^q\}_{q=1}^{Q_a}$
    Compute the FE solutions $\{\mathbf{z}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and pressure supremizers $\{\mathbf{t}_p^{\boldsymbol{\mu}_i}(\mathbf{p}_h^{\boldsymbol{\mu}_i})\}_{i=1}^{n_s}$
    Set $\mathbf{S}_{\vec{u}}^{(0)} = [\vec{u}_h^{\boldsymbol{\mu}_1}, \dots, \vec{u}_h^{\boldsymbol{\mu}_{n_s}}]$, $\mathbf{S}_p^{(0)} = [p_h^{\boldsymbol{\mu}_1}, \dots, p_h^{\boldsymbol{\mu}_{n_s}}]$, $\mathbf{S}_{\vec{t}}^{(0)} = [\mathbf{t}_p^{\boldsymbol{\mu}_1}, \dots, \mathbf{t}_p^{\boldsymbol{\mu}_{n_s}}]$ and $k = 0$
    while $\prod_k \delta_{RB,k} > \varepsilon_r$ do
        Build $\mathbf{V}_{N_k^u}^u = POD(\mathbf{S}_{\vec{u}}^{(k)}, \delta_{RB,k})$, $\mathbf{V}_{N_k^p}^p = POD(\mathbf{S}_p^{(k)}, \delta_{RB,k})$, $\mathbf{V}_{N_k^s}^s = POD(\mathbf{S}_{\vec{t}}^{(k)}, \frac{\delta_{RB,k}}{10})$
        Build $\mathbf{A}_{N_k}^q = \mathbf{V}_k^T \mathbf{A}_h^q \mathbf{V}_k$, $q = 1, \dots, Q_a$
        Compute new snapshots $\{\mathbf{y}_{uk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ and $\{\mathbf{y}_{pk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ with (31) and $\{\mathbf{y}_{sk}^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ with (26)
        Set $\mathbf{S}_{\vec{u}}^{(k+1)} = [\mathbf{y}_{uk}^{\boldsymbol{\mu}_1}, \dots, \mathbf{y}_{uk}^{\boldsymbol{\mu}_{n_s}}]$, $\mathbf{S}_p^{(k+1)} = [\mathbf{y}_{pk}^{\boldsymbol{\mu}_1}, \dots, \mathbf{y}_{pk}^{\boldsymbol{\mu}_{n_s}}]$, $\mathbf{S}_{\vec{t}}^{(k+1)} = [\mathbf{y}_{sk}^{\boldsymbol{\mu}_1}, \dots, \mathbf{y}_{sk}^{\boldsymbol{\mu}_{n_s}}]$ and $k = k+1$
    end while
end procedure
```

Algorithm 3 MSRB Preconditioner with aLS-RB coarse correction - Offline phase

```
procedure MSRB-PRECONDITIONER-ALS-RB-OFFLINE($\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}, \varepsilon_r, \{\delta_{RB,k}\}_k, \delta_{\mathrm{MDEIM}}$)
    Compute an affine approximation $\{\mathbf{A}_h^q\}_{q=1}^{Q_a}$
    Compute the FE solutions $\{\mathbf{z}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$
    Set $\mathbf{S}_{\vec{u}}^{(0)} = [\vec{u}_h^{\boldsymbol{\mu}_1}, \dots, \vec{u}_h^{\boldsymbol{\mu}_{n_s}}]$, $\mathbf{S}_p^{(0)} = [p_h^{\boldsymbol{\mu}_1}, \dots, p_h^{\boldsymbol{\mu}_{n_s}}]$ and $k = 0$
    while $\prod_k \delta_{RB,k} > \varepsilon_r$ do
        Build the new basis $\mathbf{V}_{N_k^u}^u = POD(\mathbf{S}_{\vec{u}}^{(k)}, \delta_{RB,k})$, $\mathbf{V}_{N_k^p}^p = POD(\mathbf{S}_p^{(k)}, \delta_{RB,k})$
        Build $\mathbf{A}_{N_k}^{q_1,q_2} = \mathbf{V}_k^T \mathbf{A}_h^{q_1} \mathbf{P}_X^{-1} \mathbf{A}_h^{q_2} \mathbf{V}_k$, $q_1, q_2 = 1, \dots, Q_a$
        Compute new snapshots $\{\mathbf{y}_k^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ with (31)
        Set $\mathbf{S}_{\vec{u}}^{(k+1)} = [\mathbf{y}_{uk}^{\boldsymbol{\mu}_1}, \dots, \mathbf{y}_{uk}^{\boldsymbol{\mu}_{n_s}}]$, $\mathbf{S}_p^{(k+1)} = [\mathbf{y}_{pk}^{\boldsymbol{\mu}_1}, \dots, \mathbf{y}_{pk}^{\boldsymbol{\mu}_{n_s}}]$ and $k = k + 1$
    end while
end procedure
```

Notice that instead of providing a set of tolerances $\{\delta_{RB,k}\}_k$, we can also provide a set of dimensions $\{N_k\}_k$. Indeed, we specifically devised two different strategies to build in practice the RB coarse corrections:

- *fixed accuracy* : all the tolerances $\{\delta_{RB,k}\}_k$ are chosen equal to the same value δ_{RB}, that is $\delta_{RB,k} = \delta_{RB}$ for any k. This choice leads to RB coarse corrections which provide a constant accuracy, and let the norm of the error decrease at a fixed rate at each iteration. However, the dimension of the RB spaces increases with k, leading to a larger computational time to assemble and solve the resulting RB system. If a G-RB method approach is employed, then the tolerance provided to POD for the construction of the enriching basis functions $\mathbf{V}_{N_k^s}^s$ is $\delta_{RB,k}/10$, which empirically results in a well-posed RB approximation;

- *fixed dimension*: the dimensions $\{N_k^u\}_k$ and $\{N_k^p\}_k$ (and $\{N_k^s\}_k$ if G-RB is employed) of the RB spaces are set to a fixed value N, that is $N_k^u = N_k^p = N (= N_k^s)$ for any k. This choice is specifically more convenient when we are dealing with problems showing less regular dependence on the parameter $\boldsymbol{\mu}$, since the number of RB functions in each space is fixed and cannot excessively increase.

In the numerical experiments, we will show results for both these options.

6.1.4. Sequential RB coarse correction construction

The offline phase, and especially the computation of the set of snapshots $\{\mathbf{z}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$ in step 3 of Algorithms 2 and 3 can be particularly expensive. In order to speed up the process, we can alternatively opt for a sequential construction of the RB coarse components.
With this aim, we introduce M subsets $\mathcal{Z}_m$, $m = 1, \ldots, M$, of $\{\mathbf{z}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$, of dimension n_s^m, respectively, and such that

$$\{\mathbf{z}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s} = \bigcup_{m=1}^{M} \mathcal{Z}_m, \qquad n_s = \sum_{m=1}^{M} n_s^m, \qquad \mathcal{Z}_m = \{\mathbf{z}_h^{\boldsymbol{\mu}_i}\}_{1+i_{m-1}}^{i_m},$$

where $i_m = \sum\limits_{l=1}^{m} n_s^l$. Then, the k-th RB matrix $\mathbf{V}_k$ is built using $\bigcup_{m=1}^{k} \mathcal{Z}_m$ as snapshots set. Exploiting only part of the snapshots allows to use the MSRB preconditioner developed up to iteration k for the computation of the new snapshots $\mathcal{Z}_j$, $j > k$, which will be employed to construct the RB spaces $\mathbf{V}_j$, $j > k$. This technique yields a reduction of the overall time required by the snapshot computation, since the speed up provided by the MSRB preconditioner is sequentially used to build part of the snapshots. M and $\mathcal{Z}_m$, $m = 1, \ldots, M$ are empirically chosen such that the accuracies obtained by the RB coarse corrections is not significantly impacted if compared with the ones obtained with the RB coarse corrections built with the complete set of snapshot. In the numerical experiments, we will employ $M = 3$ stages.

6.2. Online phase

In the online phase, we aim at computing the solutions of (5) for new instances of the parameter $\boldsymbol{\mu}$, which have not been considered during the offline phase. We thus need to compute the weights $\{\Theta_a^q(\mu)\}_{q=1}^{Q_a}$ of the affine decomposition of $\mathbf{A}_h^{\boldsymbol{\mu}}$, and build the coarse corrections $\{\mathbf{Q}_{N_k}^{\mu}\}_k$. Finally, we apply the the FGMRES algorithm relying on $\mathbf{M}_k = \mathbf{Q}_{\text{MSRB},k}^{\boldsymbol{\mu}}$ in the preconditioning step. The operations required by the matrix-vector multiplication $\mathbf{Q}_{\text{MSRB},k}^{\boldsymbol{\mu}}\mathbf{v}_k^{\boldsymbol{\mu}}$ are detailed in algorithm 4; step 3 corresponds to solving the RB problem

$$\mathbf{A}_{N_k}^{\boldsymbol{\mu}}\mathbf{y}_{N_k}^{\boldsymbol{\mu}} = (\mathbf{W}_k^{\boldsymbol{\mu}})^T\mathbf{v}_{k+\frac{1}{2}} \tag{37}$$

and build $\mathbf{w}_{N_k,k+\frac{1}{2}} = \mathbf{V}_k\mathbf{y}_{N_k}^{\boldsymbol{\mu}}$. If the number of iterations required to reach a certain tolerance in the FGMRES method exceeds the number of RB coarse corrections constructed, one can either continue to use the last coarse correction in the remaining operations or drop steps 2-3 of Algorithm 4.

Algorithm 4 Computation of $\mathbf{Q}_{\text{MSRB},k}^{\boldsymbol{\mu}}\mathbf{v}_k$

```
1: apply the inverse of the fine component $\mathbf{P}_h^{\boldsymbol{\mu}}$: $\mathbf{w}_k = (\mathbf{P}_h^{\boldsymbol{\mu}})^{-1}\mathbf{v}_k$;
2: build the residual $\mathbf{v}_{k+\frac{1}{2}} = \mathbf{v}_k - \mathbf{A}_h^{\boldsymbol{\mu}}\mathbf{w}_k$;
3: apply the RB coarse component $\mathbf{w}_{k+\frac{1}{2}} = \mathbf{Q}_{N_k}^{\boldsymbol{\mu}}\mathbf{v}_{k+\frac{1}{2}}$;
4: build the preconditioned Kylov basis $\mathbf{z}_k = \mathbf{w}_k + \mathbf{w}_{k+\frac{1}{2}}$.
```

7. Numerical results

In this section we show numerical results where the proposed MSRB preconditioner, based on either on a G-RB or an aLS-RB method, is employed to solve Stokes equations in parametrized geometries. Parameter dependent domains are obtained by considering a map from a reference domain to the physical domain which can be provided either analytically (test case I) or by computing the solution of an additional FE problem (test case II), e.g. when a solid extension mesh moving technique is employed, see [45]. Furthermore, we highlight that the proposed strategy is applicable also to the case where physical parameters are considered. As fine component $\mathbf{P}_h^{\boldsymbol{\mu}}$ we use the PMM preconditioner defined as

$$\mathbf{P}_h^{\boldsymbol{\mu}} = \mathbf{P}_{\mathbf{M}}^{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{D}_h^{\boldsymbol{\mu}} & (\mathbf{B}_h^{\boldsymbol{\mu}})^T \\ 0 & -\frac{1}{\nu^{\boldsymbol{\mu}}}\mathbf{X}_p^{\boldsymbol{\mu}} \end{bmatrix}, \tag{38}$$

where the Schur complement $\mathbf{S}_h^{\boldsymbol{\mu}}$ is approximated with the rescaled pressure mass matrix, that is $\tilde{\mathbf{S}}_h^{\boldsymbol{\mu}} = \frac{1}{\nu^{\boldsymbol{\mu}}}\mathbf{X}_p^{\boldsymbol{\mu}}$ (which is spectrally equivalent to $\mathbf{S}_h^{\boldsymbol{\mu}}$ at least for two-dimensional problems). The application of $(\mathbf{P}_{\mathbf{M}}^{\boldsymbol{\mu}})^{-1}$ to the $k-$th Krylov basis function $\mathbf{v}_k$ (at step k of the Krylov method) is summarized in algorithm 5.

Algorithm 5 Computation of $(\mathbf{P}_{\mathbf{M}}^{\boldsymbol{\mu}})^{-1}\mathbf{v}_k$

```
1: solve the pressure problem $-\frac{1}{\nu^{\boldsymbol{\mu}}}\mathbf{X}_p^{\boldsymbol{\mu}}\mathbf{z}_{kp} = \mathbf{v}_{kp}$ (solved inexactly by inner iterations);
2: update the velocity $\mathbf{v}_{ku} = \mathbf{v}_{ku} - (\mathbf{B}_h^{\boldsymbol{\mu}})^T\mathbf{z}_{kp}$;
3: solve the velocity problem $\mathbf{D}_h^{\boldsymbol{\mu}}\mathbf{z}_{ku} = \mathbf{v}_{ku}$ (solved inexactly by inner iterations).
```

The PMM preconditioner (38) allows to obtain extremely satisfactory results both in terms of optimality and scalability, see e.g. [18] and results therein. Specifically, the application of $\mathbf{P}_{\mathbf{M}}^{\boldsymbol{\mu}}$ is detailed in algorithm 5, where steps 1 and 3 are solved inexactly by inner iterations up to a tolerance of 10^{-5} on the Euclidean norm of the residual rescaled with the Euclidean norm of the right hand side. An algebraic multigrid (AMG) preconditioner from the ML package of Trilinos [46] is employed for the inner iterations.

We employ Taylor-Hood ($\mathbb{P}^2 - \mathbb{P}^1$) finite element spaces for velocity and pressure, respectively, which provide an *inf-sup* stable discretization. In the following, we compare the results obtained with the MSRB preconditioner with the ones obtained by using only the PMM preconditioner $\mathbf{P}_{\mathbf{M}}^{\boldsymbol{\mu}}$. The lifting function $\vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}}$ is computed as harmonic extension of the Dirichlet data $\vec{g}_D^{\boldsymbol{\mu}}$ in (1), which is chosen as a parabolic profile such that the flow rate at the inlet is equal to 1. An approximation of $\vec{r}_{\vec{g}_D}^{\boldsymbol{\mu}}$ is computed by employing the FE method, with second order polynomials basis functions. 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 computed with the preconditioned conjugate gradient (PCG) method, exploiting an AMG preconditioner from the ML package of Trilinos [46].

All the results have been obtained with the FE library LifeV [47]. Our tests have been run by employing the Swiss National Supercomputing Center (CSCS) facilities on Cray XC40 compute nodes.

7.1. Test case I: parametrized cylinder

The first test case concerns a Stokes flow in a three-dimensional cylinder whose shape varies according to a set of parameters. We introduce a reference domain

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

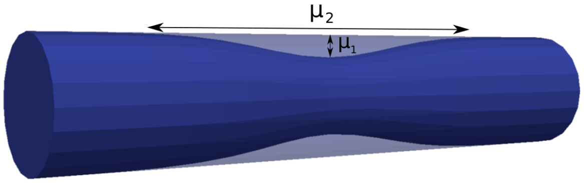


Figure 1: Deformation of the domain for test case I.

and obtain the computational domain $\Omega^{\boldsymbol{\mu}}$ as

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

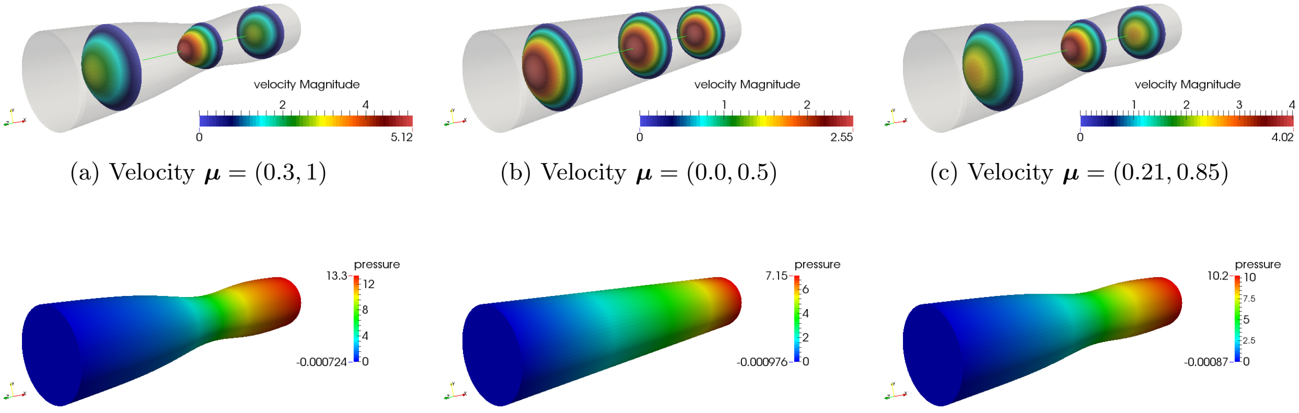
where $\vec{d^{\boldsymbol{\mu}}}$ is an analytical displacement

$$\vec{d^{\boldsymbol{\mu}}} = \begin{bmatrix} -x_1\mu_1 \exp\{-\frac{(x_3-2.5)^2}{\mu_2}\} \\ -x_2\mu_1 \exp\{-\frac{(x_3-2.5)^2}{\mu_2}\} \\ 0 \end{bmatrix}.$$

Here the parameter $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D} = (0, 0.3) \times (0.5, 1)$. The cylinder is narrowed in the central section by a factor $\mu_1/2$, whereas μ_2 determines how the narrowing effect propagates towards the inlet and outlet sections. An example of deformation is shown in Fig. 1.

7.1.1. Simulation setup

We show numerical results obtained for three different meshes, leading to a finite element problem with dimension $N_h = 52'152$, $320'338$, $1'568'223$, respectively, computed with $N_{cpu} = 36$, 180, 900 processors, thus distributing about 1800 dofs per CPU. The FE solution for different values of the parameter $\boldsymbol{\mu}$ is reported in Fig. 2.



(a) Velocity $\boldsymbol{\mu} = (0.3, 1)$

(b) Velocity $\boldsymbol{\mu} = (0.0, 0.5)$

(c) Velocity $\boldsymbol{\mu} = (0.21, 0.85)$

(d) Pressure $\boldsymbol{\mu} = (0.3, 1)$

(e) Pressure $\boldsymbol{\mu} = (0.0, 0.5)$

(f) Pressure $\boldsymbol{\mu} = (0.21, 0.85)$

Figure 2: Test case I, numerical solution for three values of $\boldsymbol{\mu}$ obtained with the MSRB preconditioning technique.

As RB coarse component, we show results for both the *fixed accuracy* and *fixed dimension* approaches

Table 1: Configurations employed for choosing the coarse corrections.

GRB	G-RB coarse corrections
$\mathbf{aLSRB\text{-}X_h^0}$	aLS-RB coarse corrections where $\mathbf{P}_X = \mathbf{X}_h^0$, i.e. the matrix norm (8) on the reference domain
$\mathbf{aLSRB\text{-}P_{X_h^0}}$	aLS-RB coarse corrections where $\mathbf{P}_X = \mathbf{P}_{\mathbf{X}_h^0}$, where $\mathbf{P}_{\mathbf{X}_h^0}$ is a symmetric and positive definite preconditioner for $\mathbf{X}_h^0$ with a block structure $\mathbf{P}_{\mathbf{X}_h^0} = diag(\mathbf{P}_{\mathbf{X}_u^0}, \mathbf{P}_{\mathbf{X}_p^0})$, 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. $\mathbf{X}_p^0$)

Table 2: Test case I, MDEIM offline results, $\delta_{\text{mdeim}} = 10^{-6}$.

N_h	Q_d	Q_b	$\mathbf{D}_h^{\boldsymbol{\mu}}$ offline time (s)	$\mathbf{B}_h^{\boldsymbol{\mu}}$ offline time (s)
52152	7	10	24.65	5.25
320338	6	10	37.29	8.11
1568223	6	10	54.37	11.71

in the configurations outlined in Tab. 1.

For the offline phase, we take $n_s = 100$ snapshots for both the construction of the RB spaces (state reduction) and the MDEIM algorithm (system approximation). Specifically, for MDEIM we set $\delta_{\text{mdeim}} = 10^{-6}$ for the construction of the affine approximation of both $\mathbf{D}_h^{\boldsymbol{\mu}}$ and $\mathbf{B}_h^{\boldsymbol{\mu}}$. Regarding the construction of the RB spaces, we take as final tolerance $\varepsilon_r = 10^{-9}$ for all the test cases. For the fixed accuracy approach we construct $L = 4$ RB spaces, yielding $\delta_{RB,k} = \delta_{RB} = 10^{-9/4} \approx 5.6 \cdot 10^{-3}$ for any k. For the fixed dimension approach, we take $N_k = 10$ for any k.

During the online phase, we test the proposed MSRB preconditioners with the three different RB coarse corrections (**GRB**, $\mathbf{aLSRB\text{-}X_h^0}$ and $\mathbf{aLSRB\text{-}P_{X_h^0}}$) and solving the FE linear system with the FGMRES method up to a tolerance, on the Euclidean norm of the residual, rescaled with the Euclidean norm of the right hand side, of 10^{-6} on 150 online parameters different from the ones employed during the offline phase to build the RB spaces.

7.1.2. Numerical results

The computational time required to compute the approximate affine decomposition of the matrices $\mathbf{D}_h^{\boldsymbol{\mu}}$ and $\mathbf{B}_h^{\boldsymbol{\mu}}$ with the MDEIM algorithm and the number of basis functions Q_a are reported in Tab. 2. The number of required basis functions Q_a mainly depends on the parameter dependence of the PDE, consequently it does not vary with the FE dimension, and ranges from 6 to 10 to reach a tolerance $\delta_{\text{mdeim}} = 10^{-6}$.

The results obtained with the MSRB preconditioner during the online phase, i.e. for new instances of the parameter, for the fixed accuracy approach with **GRB**, $\mathbf{aLSRB\text{-}X_h^0}$ and $\mathbf{aLSRB\text{-}P_{X_h^0}}$ are reported in Tab. 4, 5 and 6, respectively. For the fixed dimension approach, the results are reported in Tab. 7, 8 and 9, respectively. For each case, we report the number of RB coarse corrections L and the total number of basis functions N_k for the space k, as the sum of the velocity, pressure and supremizer RB functions, this latter only if **GRB** is employed. We underline that the number of basis functions is larger in the **GRB** case, due to the velocity enrichment. Furthermore, the detailed results concerning the time required to compute the solution by employing the PMM preconditioner t_{PMM} and the MSRB preconditioner $t_{\text{MSRB}}^{\text{onl}}$, together with the corresponding iteration counts It_{PMM} and $It_{\text{MSRB}}^{\text{onl}}$, are reported. A summary of the reported quantities is given in Tab. 3.

The number of iterations $It_{\text{MSRB}}^{\text{onl}}$ required to reach convergence in the FGMRES algorithm is lower than or equal to 6 for all the tests carried out with the MSRB preconditioner, it does not significantly vary with the FE dimension and, depending of the simulation, it is between 5% and 15% of that obtained by using the PMM preconditioner only, see Figure 3a. The computational times $t_{\text{MSRB}}^{\text{onl}}$ required to solve the FE linear system by employing the MSRB preconditioner is reduced of about 85% with respect to the one needed by employing only the PMM preconditioner t_{PMM} for the **GRB** and $\mathbf{aLSRB\text{-}P_{X_h^0}}$ cases,

Table 3: Notations of the quantities reported in the numerical experiments.

N_h	L	N_k, $k = 0, \dots, L-1$	BEP
FE dimension	number of coarse components	dimensions of RB coarse components	break even point (39)

$t^{\text{onl}}_{\text{MSRB}}$ (sec)	$It^{\text{onl}}_{\text{MSRB}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)
online time MSRB	online iteration count MSRB	online time PMM	online iteration count PMM	offline time MSRB

and is reduced of about 70% in the **aLSRB-$\mathbf{X_h^0}$**, see Figure 3b. The additional time required by this latter approach is caused by the application of the matrix $(\mathbf{X}_h^{\boldsymbol{\mu}})^{-1}$ to the vector $\mathbf{v}_{k+\frac{1}{2}}$ at each iteration of the FGMRES method (see step 3 in Alg. 4); this is practically performed by solving the corresponding linear system where $\mathbf{X}_h^{\boldsymbol{\mu}}$ is at the left hand side and $\mathbf{v}_{k+\frac{1}{2}}$ is at the right hand side. The **GRB** and **aLSRB-$\mathbf{P}_{\mathbf{X_h^0}}$** approaches entail a cheaper computation of such a step since in the former we rely on a G-RB method, while in the latter only the (fast) application of $\mathbf{P}_X^{-1}$ is required.

The computational time t_{off} required by the offline phase is reported for all tests, together with the break even point (BEP), that is, the number of online evaluations required to repay the offline phase. Our criterion is based on the wall time comparison:

$$\text{BEP} = \frac{t_{\text{off}}}{t_{\text{PMM}} - t^{\text{onl}}_{\text{MSRB}}}, \tag{39}$$

where we indicate by t_{off} the wall time required by the offline computation, i.e. the construction of the RB coarse components. We highlight that the **GRB** case requires an offline time which is larger than the others due to the need of computing the pressure supremizer snapshots $\mathbf{S}_{\vec{t}}$ and performing an additional POD. On the other hand, the offline time in the case of **aLSRB-$\mathbf{X_h^0}$** is larger than the one obtained with **aLSRB-$\mathbf{P}_{\mathbf{X_h^0}}$** due to the construction of the RB affine matrices $\mathbf{A}_{N_k}^{q_1,q_2}$, $q_1, q_2 = 1, \dots, Q_a$, because in the former case a FE linear system needs to be solved for each combination of the N_k RB functions $\{\boldsymbol{\xi}_i\}_{i=1}^N$ and Q_a affine terms $\{\mathbf{A}_h^q\}_{q=1}^{Q_a}$, leading to $N \cdot Q_a$ FE linear systems, while by employing $\mathbf{P}_X = \mathbf{P}_{\mathbf{X}_h^0}$, only $N \cdot Q_a$ applications of $\mathbf{P}_{\mathbf{X}_h^0}^{-1}$ need to be performed, boosting the computation of the affine RB structures.

By inspecting the BEP values, it emerges that the most convenient approach is obtained by adopting the **aLSRB-$\mathbf{P}_{\mathbf{X_h^0}}$** method. Indeed, it allows to solve the problem online in a computational time comparable to the one obtained with the **GRB** approach, however entailing a cheaper offline phase, especially when the FE dimension increases.

Table 4: Test case I, fixed accuracy with **GRB**, $L = 4$, $\delta_{RB,k} \approx 5.6 \cdot 10^{-3}$, $\forall k$.

N_h	N_k	$t^{\text{onl}}_{\text{MSRB}}$ (sec)	$It^{\text{onl}}_{\text{MSRB}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
52152	9 24 50 113	0.72	3	4.70	40	1514.78	374
320338	9 24 48 118	1.30	3	11.32	42	2951.76	291
1568223	9 23 48 116	5.10	3	30.65	42	9548.40	372

Table 5: Test case I, fixed accuracy with **aLSRB-$\mathbf{X_h^0}$**, $L = 4$, $\delta_{RB,k} \approx 5.6 \cdot 10^{-3}$, $\forall k$.

N_h	N_k	$t^{\text{onl}}_{\text{MSRB}}$ (sec)	$It^{\text{onl}}_{\text{MSRB}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
52152	5 13 24 54	1.97	4	4.70	40	1493.10	535
320338	5 13 23 56	4.82	6	11.32	42	3411.82	519
1568223	5 13 23 52	11.25	6	30.65	42	8542.47	437

Table 6: Test case I, fixed accuracy with $\mathbf{aLSRB}$-$\mathbf{P}_{\mathbf{X_h^0}}$, $L = 4$, $\delta_{RB,k} \approx 5.6 \cdot 10^{-3}$, $\forall k$.

N_h	N_k	$t_{\text{MSRB}}^{\text{onl}}$ (sec)	$It_{\text{MSRB}}^{\text{onl}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
52152	5 13 24 53	1.29	4	4.70	40	1374.38	395
320338	5 13 23 55	2.57	6	11.32	42	2727.60	307
1568223	5 13 23 52	5.36	6	30.65	42	6975.20	274

Table 7: Test case I, fixed dimension with $\mathbf{GRB}$, $N_k^u = N_k^p = N_k^s = 10$, $\forall k$.

N_h	L	$t_{\text{MSRB}}^{\text{onl}}$ (sec)	$It_{\text{MSRB}}^{\text{onl}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
52152	9	0.51	2	4.70	40	2476.72	584
320338	7	1.24	3	11.32	42	4546.66	447
1568223	8	4.74	3	30.65	42	18369.68	707

Table 8: Test case I, fixed dimension with $\mathbf{aLSRB}$-$\mathbf{X_h^0}$, $N_k^u = N_k^p = 10$, $\forall k$.

N_h	L	$t_{\text{MSRB}}^{\text{onl}}$ (sec)	$It_{\text{MSRB}}^{\text{onl}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
52152	9	1.51	5	4.70	40	2507.38	776
320338	8	5.12	6	11.32	42	6770.73	1086
1568223	8	10.14	5	30.65	42	15121.68	735

Table 9: Test case I, fixed dimension with $\mathbf{aLSRB}$-$\mathbf{P}_{\mathbf{X_h^0}}$, $N_k^u = N_k^p = 10$, $\forall k$.

N_h	L	$t_{\text{MSRB}}^{\text{onl}}$ (sec)	$It_{\text{MSRB}}^{\text{onl}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
52152	9	1.17	5	4.70	40	2886.91	810
320338	8	2.74	6	11.32	42	5475.75	633
1568223	8	4.75	5	30.65	42	11489.38	442

7.2. Test case II: parametrized carotid bifurcations

In the second test case, we consider parametrized Stokes flows in a carotid bifurcation, whose shape varies according to a set of parameters. The computational domain $\Omega^{\boldsymbol{\mu}}$ is obtained by deforming a reference domain Ω^0, shown in Fig. 4a, such that $\partial\Omega^0 = \Gamma_w \cup \Gamma_{in} \cup \Gamma_{out}$. 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 the displacement $\vec{d}^{\boldsymbol{\mu}}$ is computed as the solution of the following parametrized elliptic problem

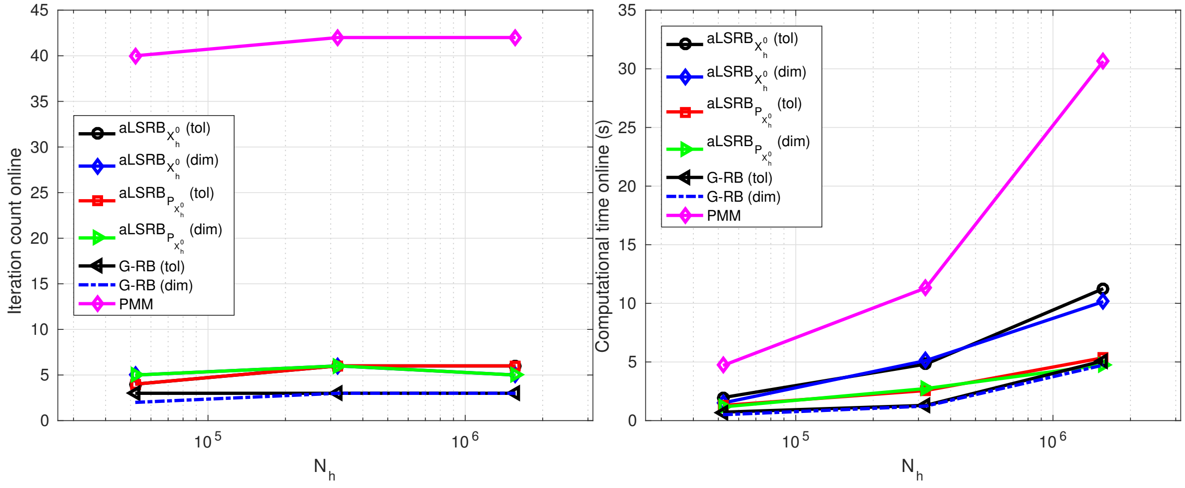
$$\begin{cases} -\Delta \vec{d}^{\boldsymbol{\mu}} = \vec{0} & \text{in } \Omega^0 \\ \vec{d}^{\boldsymbol{\mu}} = \vec{0} & \text{on } \Gamma_{in} \cup \Gamma_{out} \\ \dfrac{\partial \vec{d}^{\boldsymbol{\mu}}}{\partial \vec{n}} = \vec{h}^{\boldsymbol{\mu}} & \text{on } \Gamma_w. \end{cases} \tag{40}$$

The parametrized datum $\vec{h}^{\boldsymbol{\mu}}$ is a stress load entailing a deformation leading to the narrowing of one of the branches of the bifurcation. We consider as parameter $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D} = [4,\, 5] \times [0,\, 0.5]$ and introduce a $\boldsymbol{\mu}-$dependent region $A^{\boldsymbol{\mu}}$, such that

$$A^{\boldsymbol{\mu}} = \{\vec{x} \in \mathbb{R}^3 : \ (x_1 + 0.8)^2 + (x_2 - \mu_1)^2 + (x_3)^2 < R^2,\, R = 0.65\},$$

which identifies the portion of volume where $\vec{h}^{\boldsymbol{\mu}}$ is loaded as follows

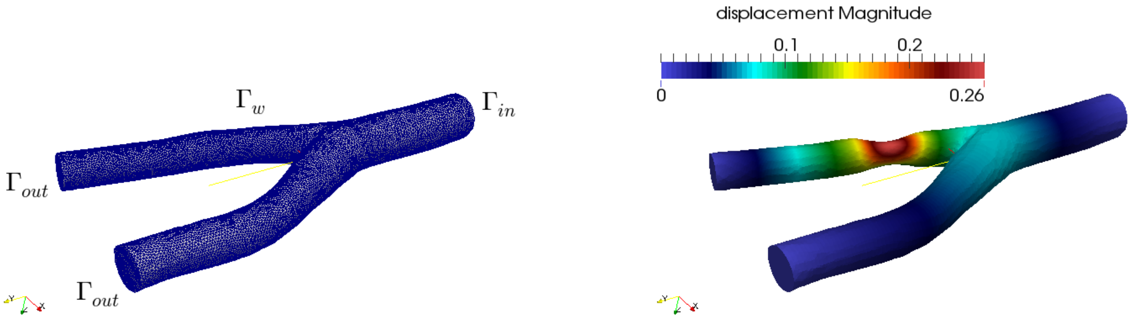
$$\vec{h}^{\boldsymbol{\mu}} = \vec{h}^{\boldsymbol{\mu}}(\vec{x}) = -\mu_2 \Big(1 - \frac{r^2(\vec{x})}{R^2}\Big) \vec{n}^{\boldsymbol{\mu}} \mathbb{I}_{A^{\boldsymbol{\mu}}}(\vec{x}), \qquad \vec{x} \in \mathbb{R}^3,$$



(a) Iterations vs N_h.

(b) Computational times (sec) vs N_h.

Figure 3: Test case I, iteration number and computational times vs N_h. The name *dim* or *tol* specifies that a fixed accuracy or fixed dimension algorithm has been used for constructing the RB coarse components, respectively. We refer to Tab. 1 for how the RB method employed for the construction (**GRB**, **aLSRB-$\mathbf{X_h^0}$** or **aLSRB-$\mathbf{P_{X_h^0}}$**).



(a) Reference domain Ω^0.

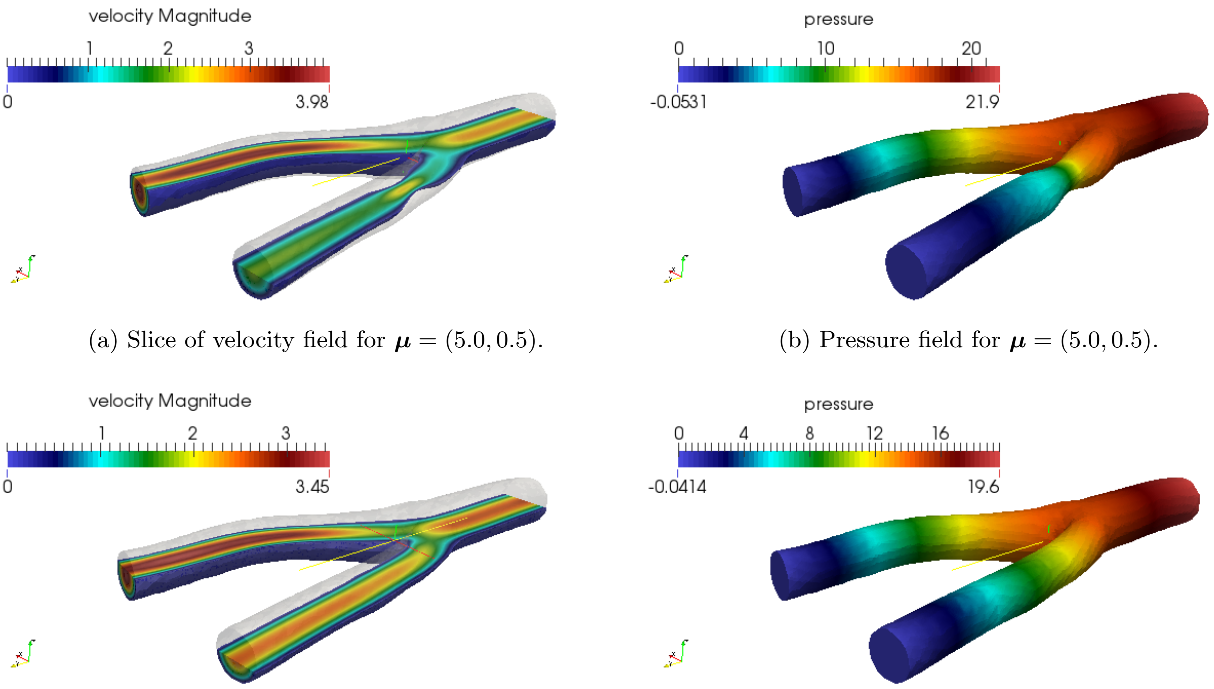
(b) Displacement field for $\boldsymbol{\mu} = (5.0, 0.5)$.

Figure 4: Test case II, reference domain Ω^0 and displacement $\mathbf{d}_h^{\boldsymbol{\mu}}$ for $\boldsymbol{\mu} = (5.0, 0.5)$.

where $r(\vec{x}) = r^{\boldsymbol{\mu}}(\vec{x}) = \sqrt{(x_1 + 0.8)^2 + (x_2 - \mu_1)^2 + (x_3)^2}$, $R = 0.65$ and $\mathbb{I}_{A^{\boldsymbol{\mu}}}(\vec{x})$ is the indicator function over the set $A^{\boldsymbol{\mu}}$. This parametrization entails a narrowing of the straight branch in different positions along the coordinate x_2 (according to the value of μ_2) and simulates an occlusion. An example of deformation computed for $\boldsymbol{\mu} = (5.0, 0.5)$ is shown in Fig. 4b. Examples of solutions for different values of the parameter $\boldsymbol{\mu}$ are shown in Fig. 5a-5b and 5c-5d.

We remark that the solution $\vec{d}^{\boldsymbol{\mu}}$ of problem (40) is not known analytically; consequently, its numerical approximation $\vec{d}_h^{\boldsymbol{\mu}}$ is computed employing the FE method on its corresponding variational formulation. We denote by $\mathbf{d}_h^{\boldsymbol{\mu}} \in \mathbb{R}^{N_h^d}$ the solution of the corresponding FE linear system.

In the numerical results we show, Taylor-Hood FE ($\mathbb{P}^2 - \mathbb{P}^1$), with a mesh leading to $N_h = N_h^u + N_h^p = 3'198'820$ degrees of freedom, are employed for the FE discretization of the Stokes problem. The computation is carried out by using 360 computing cores.



(a) Slice of velocity field for $\boldsymbol{\mu} = (5.0, 0.5)$.

(b) Pressure field for $\boldsymbol{\mu} = (5.0, 0.5)$.

(c) Slice of velocity field for $\boldsymbol{\mu} = (4.0, 0.0)$.

(d) Pressure field for $\boldsymbol{\mu} = (4.0, 0.0)$.

Figure 5: Test case II, numerical solution for two values of $\boldsymbol{\mu}$ obtained with the MSRB preconditioning technique.

7.2.1. Simulation setup

When considering a new instance of the parameter $\boldsymbol{\mu}$, we compute $\mathbf{d}_h^{\boldsymbol{\mu}}$ by solving the corresponding FE linear system with the PCG method, preconditioned with the AMG preconditioner. The system is solved up to a tolerance 10^{-8} on the Euclidean norm of the residual rescaled with the Euclidean norm of the right hand side. The computation of the deformation $\mathbf{d}_h^{\boldsymbol{\mu}}$ requires on average 1.9 seconds and this time is not included in the results reported, since it does not vary in the different scenarios presented. Notice that we could accelerate the computation of $\mathbf{d}_h^{\boldsymbol{\mu}}$ by employing the MSRB preconditioning strategy or the standard RB method to deal with problem (40). Then, the solution of the Stokes problem (5) is computed employing the MSRB preconditioner, we report in particular the results obtained with the **aLSRB-P**$_{\mathbf{X}_\mathbf{h}^\mathbf{0}}$ case and the fixed dimension approach only, however an analysis similar to the one carried out to Test case I can be done. For the aim of RB spaces construction, we use $n_s = 350$ snapshots, which are computed incrementally as explained in Section 6.1.4, with $M = 3$ and $n_s^1 = 100$, $n_s^2 = 100$ and $n_s^3 = 150$. Then, we set $\varepsilon_r = 10^{-7}$, by choosing $N_k^u = N_k^p = 50$ for any $k = 0, \dots, L-1$, leading to L coarse corrections with dimension $N_k = 100$ for any $k = 0, \dots, L-1$. We test the resulting preconditioner on 100 online instances of the parameter randomly chosen, by solving the resulting FE problem up to a tolerance 10^{-5}. For the MSRB preconditioner, we employ MDEIM (with tolerance $\delta_{\mathrm{mdeim}} = 10^{-4}$) to compute an approximated affine decomposition of $\mathbf{A}_h^{\boldsymbol{\mu}}$, allowing us to cheaply assemble online the coarse corrections $\mathbf{A}_{N_k}^{\boldsymbol{\mu}}$, $k = 0, \dots, L-1$.

We compare the results obtained with the MSRB precondtioner with the ones obtained by relying on the standard RB method, where the **aLSRB-P**$_{\mathbf{X}_\mathbf{h}^\mathbf{0}}$ approach detailed in AppendixA.2 is used as solver. For this latter, we build the RB basis functions by using POD with a tolerance of 10^{-9} on $n_s = 350$ snapshots; then we construct the RB approximation by affinely approximating the FE right hand sides and matrices in (6) by using DEIM and MDEIM, respectively. Indeed, we remark that, as highlighted

Table 10: Test case 2, (M)DEIM number of affine basis functions.

$\delta_{\text{deim}} = \delta_{\text{mdeim}}$	MDEIM - $\mathbf{D}_h^{\boldsymbol{\mu}}$	MDEIM - $\mathbf{B}_h^{\boldsymbol{\mu}}$	DEIM - $\mathbf{f}_h^{\boldsymbol{\mu}}$	DEIM - $\mathbf{r}_h^{\boldsymbol{\mu}}$	t_{affine} (sec)
1e-02	1	3	3	4	75.41
1e-03	1	6	6	13	184.68
1e-04	3	17	15	25	1165.29
1e-05	8	36	29	48	5013.85
1e-06	19	79	63	117	49129.40

Table 11: Test case 2, **aLSRB-P**$_{\mathbf{X_h^0}}$ solver, $\delta_{RB} = 10^{-9}$, $N_u = 327$ and $N_p = 111$.

$\delta_{\text{deim}} = \delta_{\text{mdeim}}$	r_{RB}	$t_{\text{RB}}^{\text{onl}}$ (sec)	t_{off} (sec)
1e-02	1.9e-02	5.75	41931.61
1e-03	4.0e-03	5.39	42040.87
1e-04	1.1e-03	5.33	43021.49
1e-05	2.8e-04	5.81	46870.05
1e-06	6.3e-05	8.66	90985.60

in Section 6.1.2, the standard RB method also relies on the affine dependence of the FE right hand side $\mathbf{g}_h^{\boldsymbol{\mu}}$. Since in the considered test case this assumption is not satisfied, DEIM is performed on the right hand side to compute an affine approximation of the vectors $\mathbf{f}_h^{\boldsymbol{\mu}}$ and $\mathbf{r}_h^{\boldsymbol{\mu}}$. Furthermore, MDEIM is used to compute an approximated affine decomposition of the FE stiffness matrix $\mathbf{A}_h^{\boldsymbol{\mu}}$, which is used to cheaply assemble the RB matrix $\mathbf{A}_N^{\boldsymbol{\mu}}$ online.

7.2.2. Numerical results: comparison with the standard RB method

We show the results obtained by using the **aLSRB-P**$_{\mathbf{X_h^0}}$ method as solver on a set of 100 instances of the parameter and varying the tolerances δ_{mdeim} and δ_{deim} employed for the MDEIM and DEIM algorithms, respectively. In Tab. 10, the number of affine components for the different FE arrays is reported, together with the computational time (part of the offline phase of the standard RB method) t_{affine} to build and store the affine RB matrices $\mathbf{A}_N^{q_1,q_2}$, $q_1, q_2 = 1, \ldots, Q_a$ in (A.6) and the RB vectors $\mathbf{g}_N^{q_1,q_2}$, $q_1, \ldots, Q_a, q_1, \ldots, Q_g$ in (A.7). Notice that the number of affine basis functions largely affects the time t_{affine}, leading overall to a more demanding offline phase.

By setting $\delta_{RB} = 10^{-9}$ to construct the RB space, we obtain $N_u = 327$ and $N_p = 111$ basis functions for velocity and pressure, respectively. In order to evaluate the accuracy of the RB solution, we compute the relative residual of the FE problem evaluated on the RB solution

$$r_{RB}^{\boldsymbol{\mu}} = \frac{\left\|\mathbf{g}_h^{\boldsymbol{\mu}} - \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{z}_N^{\boldsymbol{\mu}}\right\|_2}{\left\|\mathbf{g}_h^{\boldsymbol{\mu}}\right\|_2},$$

which we report in Tab. 11. As a matter of fact, in order to obtain an accurate RB solution, it is mandatory to build an accurate approximate affine decomposition of the FE arrays, cf. Tab. 11, since the accuracy of the RB solution is strongly related to the accuracy of the affine approximations. The online time t_{onl} to assemble and solve the RB problem is largely affected by the values δ_{deim} and δ_{mdeim} and reaches up to 8.66 seconds in the most demanding case. In particular, the time for assembling the RB matrix $\mathbf{A}_N^{\boldsymbol{\mu}}$ and the time for assembling the RB right hand side $\mathbf{g}_N^{\boldsymbol{\mu}}$ are the most affected ones by the number of affine components. As regards the computational time t_{off} required by the offline phase, it largely increases according to the number of affine terms, since it takes into account the time t_{affine} reported in Tab. 10.

In Tab. 12, the results obtained with the FGMRES method preconditioned with MSRB preconditioner (with **aLSRB-P**$_{\mathbf{X_h^0}}$ coarse corrections) are presented. We employ MDEIM with $\delta_{\text{mdeim}} = 10^{-4}$ to build an approximated affine decomposition of the FE matrices $\mathbf{D}_h^{\boldsymbol{\mu}}$ and $\mathbf{B}_h^{\boldsymbol{\mu}}$, leading to $Q_d = 3$ and

Table 12: Test case 2, fixed dimension with **aLSRB-P$_{\mathbf{X}_h^0}$**, RES $= 10^{-5}$, $N_k^u = N_k^p = 50$, $\forall k$, $\sim$ 8890 dofs per CPU.

N_h	L	$t^{\text{onl}}_{\text{MSRB}}$ (sec)	$It^{\text{onl}}_{\text{MSRB}}$	t_{PMM} (sec)	It_{PMM}	t_{off} (sec)	BEP
3198820	4	6.45	5	80.69	87	46554.90	628

$Q_b = 17$ affine basis functions, respectively. As a matter of fact, the value $\delta_{\text{mdeim}} = 10^{-4}$ does not affect the accuracy of the RB approximation to the solution of the error equation constructed with the RB functions obtained by setting $N_k^u = N_k^p = 50$, $\forall k$. Furthermore, we notice that in this context there is no need to employ DEIM to approximate $\mathbf{f}_h^{\boldsymbol{\mu}}$ and $\mathbf{r}_h^{\boldsymbol{\mu}}$, as explained in Section 6.1.2.

$L = 4$ RB spaces are computed with a dimension $N_k^u = N_k^p = 50$ for $k = 0, 1, 2, 3$ for both velocity and pressure; as a matter of fact, the convergence up to a tolerance of 10^{-5} on $r_{RB}^{\boldsymbol{\mu}}$ is reached on average in 5 iterations and about 6.45 seconds.

The smaller number of MDEIM affine terms used by the MSRB preconditioner yields a significantly more accurate solution (with a residual $r_{RB}^{\boldsymbol{\mu}}$ lower than 10^{-5}) in a shorter computational time, compared to the one computed with the standard RB method. In addition, the obtained results show that a cheaper offline phase is also achieved, thanks to a smaller number of RB affine arrays which need to be constructed.

Finally, we compare the computational time employed by the FGMRES preconditioned with the MSRB preconditioner, with the one needed to solve the same problem with the FGMRES method preconditioned with the PMM preconditioner, reported in Tab. 12 as well. When this latter technique is employed, the problem is solved in about 80.69 seconds and 87 iterations, on average. Therefore the proposed MSRB technique allows to obtain the solution by reducing of more than 92% the time needed by employing the PMM preconditioner.

8. Conclusions

In this work we have extended the MSRB preconditioner to the case of parametrized linear saddle-point problems. This can be achieved by using a RB coarse correction which takes advantage of either an augmented RB space G-RB approach or a PG-RB formulation. If the former approach is employed, the well-posedness of the corresponding preconditioner is ensured by the results in [1]. In this work, we have extended such results to the case where the latter option is used. Furthermore, we have introduced a new sequential construction of the snapshots which mitigates the offline costs by using the MSRB preconditioning technique to compute part of the snapshots.

We have tested the MSRB preconditioning method when dealing with the 3-D parametrized Stokes equations of large dimension in parameter-dependent domains of variable shape. We compared the obtained results with the ones obtained by using a PMM preconditioner. The proposed technique enables to compute the solution for each new instance of the parameter much more rapidly than by employing only the PMM preconditioner in the online phase, reducing dramatically the computational time (up to about 92%) and the iteration count when a new instance of the parameter is considered. A comparison with the standard RB method has been carried out, showing that the MSRB preconditioning approach has a milder dependence on the affine approximation of the FE arrays than the RB method, and allows to compute a more accurate solution in a shorter time during the online phase, and not requiring a too expensive offline phase.

AppendixA. Building a well-posed RB Stokes problem

In the following we briefly recall how to build a stable Stokes RB problem either with an enriched-velocity G-RB or an aLSRB formulation. These two techniques are employed in Section 4 to build the RB coarse components of the MSRB preconditioner.

AppendixA.1. G-RB method with velocity enrichment

A stable G-RB approximation (13) is built by considering the matrix

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

instead of $\mathbf{V}$ in (13), and choosing $\mathbf{W}^{\boldsymbol{\mu}} = \tilde{\mathbf{V}}$. The columns of the matrix $\mathbf{V}_{N_s}$ span the enriching velocity space, and are computed by POD as

$$\mathbf{V}_{N_s} = \mathrm{POD}\Big(\mathbf{S}_{\vec{t}}, \mathbf{X}_u, \varepsilon_{\mathrm{POD}}\Big),$$

where the columns of the matrix $\mathbf{S}_{\vec{t}} \in \mathbb{R}^{N_h^u \times n_s}$ are the snapshots $\{\mathbf{t}_p^{\boldsymbol{\mu}_i}(\mathbf{p}_h^{\boldsymbol{\mu}_i})\}_{i=1}^{n_s}$, obtained by solving n_s problems

$$\mathbf{X}_u^{\boldsymbol{\mu}} \mathbf{t}_p^{\boldsymbol{\mu}_i} = (\mathbf{B}_h^{\boldsymbol{\mu}_i})^T \mathbf{p}_h^{\boldsymbol{\mu}_i} \qquad i = 1, \dots, n_s, \tag{A.1}$$

which involve the pressure snapshots $\{\mathbf{p}_h^{\boldsymbol{\mu}_i}\}_{i=1}^{n_s}$. Solving the FE system (A.1) corresponds to compute the element of $\mathbb{R}^{N_h^u}$ which reaches the supremum in (9) for a fixed pressure $\mathbf{p}_h^{\boldsymbol{\mu}_i}$ and a fixed parameter value $\boldsymbol{\mu}_i$. Although it is not possible to state a stability result for the G-RB approximation obtained in this way, numerically it provides very satisfying results; for a more detailed analysis see e.g. [41, 33].

AppendixA.2. Algebraic Least Squares RB methods

By considering the matrix $\mathbf{P}_X$ introduced in Section 5.2, a well posed RB Stokes problem is obtained when choosing a projection matrix of the following form

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

is chosen, leading to the RB system

$$\mathbf{A}_N^{\boldsymbol{\mu}} \mathbf{z}_N^{\boldsymbol{\mu}} = \mathbf{g}_N^{\boldsymbol{\mu}}. \tag{A.2}$$

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

$$\mathbf{A}_N^{\boldsymbol{\mu}} = \mathbf{V}^T (\mathbf{A}_h^{\boldsymbol{\mu}})^T \mathbf{P}_X^{-1} \mathbf{A}_h^{\boldsymbol{\mu}} \mathbf{V} \qquad \mathbf{g}_N^{\boldsymbol{\mu}} = \mathbf{V}^T (\mathbf{A}_h^{\boldsymbol{\mu}})^T \mathbf{P}_X^{-1} \mathbf{g}_h^{\boldsymbol{\mu}}; \tag{A.3}$$

the resulting aLS-RB problem automatically fulfills (17). This technique has provided satisfying results especially when the domain $\Omega^{\boldsymbol{\mu}}$ depends on the parameter through a map which is not known analytically. See [33] for further details.

AppendixA.3. Assembling the RB problem

The standard RB method, together with the affine decomposition (32) of the matrix $\mathbf{A}_h^{\boldsymbol{\mu}}$, strongly relies also on the affine decomposition of the right hand side $\mathbf{g}_h^{\boldsymbol{\mu}}$, that is, it must hold

$$\mathbf{g}_h^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu}) \mathbf{g}_h^q, \tag{A.4}$$

where $\Theta_g^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_g$ are $\boldsymbol{\mu}$-dependent functions, while the vectors $\mathbf{g}_h^q \in \mathbb{R}^{N_h}$ are $\boldsymbol{\mu}$-independent. If assumptions (32) and (A.4) are verified, then the RB matrix $\mathbf{A}_N^{\boldsymbol{\mu}}$ and the RB vector $\mathbf{g}_N^{\boldsymbol{\mu}}$

can be constructed in the G-RB case as

$$\mathbf{A}_N^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \tilde{\mathbf{V}}^T \mathbf{A}_h^q \tilde{\mathbf{V}} = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_N^q, \qquad \mathbf{g}_N^{\boldsymbol{\mu}} = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu}) \tilde{\mathbf{V}}^T \mathbf{g}_h^q = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu}) \mathbf{g}_N^q. \tag{A.5}$$

and in the aLS-RB case as

$$\mathbf{A}_N^{\boldsymbol{\mu}} = \sum_{q_1,q_2=1}^{Q_a} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_a^{q_2}(\boldsymbol{\mu}) \mathbf{V}^T (\mathbf{A}_h^{q_1})^T \mathbf{P}_X^{-1} \mathbf{A}_h^{q_2} \mathbf{V} = \sum_{q_1,q_2=1}^{Q_a} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_a^{q_2}(\boldsymbol{\mu}) \mathbf{A}_N^{q_1,q_2}, \tag{A.6}$$

$$\mathbf{g}_N^{\boldsymbol{\mu}} = \sum_{q_1=1}^{Q_a} \sum_{q_2=1}^{Q_g} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_g^{q}(\boldsymbol{\mu}) \mathbf{V}^T (\mathbf{A}_h^{q_1})^T \mathbf{P}_X^{-1} \mathbf{g}_h^{q_2} = \sum_{q_1=1}^{Q_a} \sum_{q_2=1}^{Q_g} \Theta_a^{q_1}(\boldsymbol{\mu}) \Theta_g^{q_2}(\boldsymbol{\mu}) \mathbf{g}_N^{q_1,q_2}. \tag{A.7}$$

The matrices $\mathbf{A}_N^q \in \mathbb{R}^{N \times N}$, $q = 1, \ldots, Q_a$, $\mathbf{A}_N^{q_1,q_2} \in \mathbb{R}^{N \times N}$, $q_1, q_2 = 1, \ldots, Q_a$ and the vectors $\mathbf{g}_N^q \in \mathbb{R}^N$, $q = 1, \ldots, Q_a$, $\mathbf{g}_N^{q_1,q_2} \in \mathbb{R}^N$, $q_1 = 1, \ldots, Q_a, q_2 = 1, \ldots, Q_g$, depending on the chosen RB approximation, can be precomputed and stored once the RB space $\mathbf{V}$ is constructed. Then, given a new value $\boldsymbol{\mu}$ of parameter, only the sums in (A.5) or (A.6)-(A.7) must be carried out, boosting the efficiency of the RB appoximation computation.

If assumptions (32)-(A.4) can not be verified, one can rely on the EIM or its discrete variants DEIM and MDEIM to compute an approximated affine decomposition, see [42, 44, 43]. These techniques allow to build an *approximate* affine decomposition, such that relations (32) and (A.4) are satisfied up to a certain tolerance

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

where Q_a and Q_g are, in our case, the number of selected basis computed by MDEIM and DEIM, respectively. As stated in Section 6.1.2, in our experiments MDEIM is run separately on $\mathbf{D}_h^{\boldsymbol{\mu}}$ and $\mathbf{B}_h^{\boldsymbol{\mu}}$, while as the right hand side $\mathbf{g}_h^{\boldsymbol{\mu}}$ concerns, DEIM is run separately on $\mathbf{f}_h^{\boldsymbol{\mu}}$ and $\mathbf{r}_h^{\boldsymbol{\mu}}$, in order to obtain an approximated affine decomposition

$$\mathbf{f}_h^{\boldsymbol{\mu}} \approx \sum_{q=1}^{Q_f} \tilde{\Theta}_f^q(\boldsymbol{\mu}) \mathbf{f}_h^q, \qquad \mathbf{r}_h^{\boldsymbol{\mu}} \approx \sum_{q=1}^{Q_r} \tilde{\Theta}_r^q(\boldsymbol{\mu}) \mathbf{r}_h^q,$$

where the functions $\tilde{\Theta}_f^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_f$ and $\tilde{\Theta}_b^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_r$ are $\boldsymbol{\mu}-$dependent and the matrices $\mathbf{D}_h^q \in \mathbb{R}^{N_h^u \times N_h^u}$, $q = 1, \ldots, Q_f$ and $\mathbf{B}_h^q \in \mathbb{R}^{N_h^p \times N_h^u}$, $q = 1, \ldots, Q_r$ are $\boldsymbol{\mu}$-independent.

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. We acknowledge the CSCS for providing us the CPU resources under project ID s796.

References

[1] N. Dal Santo, S. Deparis, A. Manzoni, A. Quarteroni, Multi space reduced basis preconditioners for large-scale parametrized pdes, SIAM Journal on Scientific Computing 40 (2) (2018) A954–A983.

[2] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers, ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique 8 (R2) (1974) 129–151.

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

[4] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, Spectral methods in fluid dynamics, Springer Science & Business Media, 2012.

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

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

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

[8] B. Gmeiner, U. Rüde, H. Stengel, C. Waluga, B. Wohlmuth, Performance and scalability of hierarchical hybrid multigrid solvers for stokes systems, SIAM Journal on Scientific Computing 37 (2) (2015) C143–C168.

[9] B. Gmeiner, M. Huber, L. John, U. Rüde, B. Wohlmuth, A quantitative performance study for stokes solvers at the extreme scale, Journal of Computational Science 17 (2016) 509–521.

[10] J. Schmidt, M. Berzins, J. Thornock, T. Saad, J. Sutherland, Large scale parallel solution of incompressible flow problems using uintah and hypre, in: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, IEEE, 2013, pp. 458–465.

[11] S. Turek, Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approache, Vol. 6, Springer Science & Business Media, 1999.

[12] P. Wesseling, C. W. Oosterlee, Geometric multigrid with applications to computational fluid dynamics, Journal of Computational and Applied Mathematics 128 (1) (2001) 311–334.

[13] G. Wittum, Multi-grid methods for Stokes and Navier-Stokes equations, Numerische Mathematik 54 (5) (1989) 543–563.

[14] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, R. Tuminaro, Block preconditioners based on approximate commutators, SIAM Journal on Scientific Computing 27 (5) (2006) 1651–1668.

[15] D. A. May, L. Moresi, Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, Physics of the Earth and Planetary Interiors 171 (1) (2008) 33–47.

[16] D. Kay, D. Loghin, A. Wathen, A preconditioner for the steady-state Navier–Stokes equations, SIAM Journal on Scientific Computing 24 (1) (2002) 237–256.

[17] D. Silvester, H. C. Elman, D. Kay, A. Wathen, Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow, Journal of Computational and Applied Mathematics 128 (1) (2001) 261–279.

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

[19] J. Pestana, A. Wathen, Natural preconditioning and iterative methods for saddle point systems, SIAM Review 57 (1) (2015) 71–91. doi:10.1137/130934921.

[20] J. W. Pearson, J. Pestana, D. J. Silvester, Refined saddle-point preconditioners for discretized stokes problems, Numerische mathematik 138 (2) (2018) 331–363.

[21] K. Vuik, A. Saghir, G. Boerstoel, The Krylov accelerated SIMPLE (R) method for flow problems in industrial furnaces, International Journal for Numerical Methods in Fluids 33 (7) (2000) 1027–1040.

[22] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, R. Tuminaro, A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations, Journal of Computational Physics 227 (3) (2008) 1790–1808.

[23] L. Little, Y. Saad, Block preconditioners for saddle point problems, Numerical Algorithms 33 (1-4) (2003) 367–379.

[24] D. Forti, Parallel algorithms for the solution of large-scale fluid-structure interaction problems in hemodynamics, Ph.D. thesis, EPFL (2016).

[25] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta numerica 14 (2005) 1–137.

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

[27] A. J. Wathen, Preconditioning, Acta Numerica 24 (2015) 329–376. doi:10.1017/S0962492915000021.

[28] 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) (2008) 2039–2067.

[29] 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) (2014) 1199–1226.

[30] F. Negri, G. Rozza, A. Manzoni, A. Quarteroni, Reduced basis method for parametrized elliptic optimal control problems, SIAM Journal on Scientific Computing 35 (5) (2013) A2316–A2340.

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

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

[33] N. Dal Santo, S. Deparis, A. Manzoni, A. Quarteroni, An algebraic least squares reduced basis method for the solution of parametrized Stokes equations, Tech. Rep. 21.2017, MATHICSE–EPFL (2017).

[34] A. Abdulle, O. Budáč, A Petrov–Galerkin reduced basis approximation of the Stokes equation in parameterized geometries, C. R. Math. Acad. Sci. Paris 353 (7) (2015) 641–645.

[35] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Springer, 2016.

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

[37] N. Dal Santo, S. Deparis, A. Manzoni, A numerical investigation of multi space reduced basis preconditioners for parametrized elliptic advection-diffusion equations, Communications in Applied and Industrial Mathematics 8 (1) (2017) 282–297.

[38] Y. Saad, Iterative methods for sparse linear systems, SIAM, 2003.

[39] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM Journal on Scientific Computing 14 (2) (1993) 461–469.

[40] G. Rozza, D. Huynh, 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) (2013) 115–152.

[41] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1136–1161.

[42] M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique Académie des Sciences Paris 339 (9) (2004) 667–672.

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

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

[45] A. Manzoni, F. Negri, Efficient reduction of PDEs defined on domains with variable shape, in: P. Benner, M. Ohlberger, A. Patera, G. Rozza, K. Urban (Eds.), Model Reduction of Parametrized Systems, Vol. 17, Springer, Cham, 2017, pp. 183–199.

[46] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, M. G. Sala, Ml 5.0 smoothed aggregation user's guide, Tech. rep., SAND2006-2649, Sandia National Laboratories (2006).

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

MOX Technical Reports, last issues

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

52/2018 Possenti, L.; di Gregorio, S.; Gerosa, F.M.; Raimondi, G.; Casagrande, G.; Costantino, M.L.; Z
A computational model for microcirculation including Fahraeus-Lindqvist effect, plasma skimming and fluid exchange with the tissue interstitium

53/2018 Giantesio, G.; Musesti, A.; Riccobelli, D.
A comparison between active strain and active stress in transversely isotropic hyperelastic materials

51/2018 Stella, S.; Vergara, C.; Giovannacci, L.; Quarteroni, A.; Prouse, G.
Assessing the disturbed flow and the transition to turbulence in the arteriovenous fistula

50/2018 Gervasio, P.; Quarteroni, A.
The INTERNODES method for non-conforming discretizations of PDEs

47/2018 Stefanucci, M.; Sangalli, L.M.; Brutti, P.
PCA-based discrimination of partially observed functional data, with an application to Aneurisk65 dataset

49/2018 Massi, M.C.; Ieva, F.; Lettieri, E.
Data Mining Application to Healthcare Fraud Detection: A Two-Step Unsupervised Clustering Model for Outlier Detection with Administrative Databases

48/2018 Arnone, E.; Azzimonti, L.; Nobile, F.; Sangalli, L.M.
Modeling spatially dependent functional data via regression with differential regularization

46/2018 Riccobelli, D.; Ciarletta, P.
Morpho-elastic model of the tortuous tumour vessels

44/2018 Bernardi, M.S.; Sangalli, L.M.
Modelling spatially dependent functional data by spatial regression with differential regularization

45/2018 Bernardi, M.S.; Carey, M.; Ramsay, J.O.; Sangalli, L.M.
Modeling spatial anisotropy via regression with partial differential regularization