ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MATHICSE

Mathematics Institute of Computational Science and Engineering

School of Basic Sciences - Section of Mathematics

MATHICSE Technical Report

Nr. 32 .2016

August 2016 (New 22.05.2017)

Multi space reduced basis preconditioners for large-scale parametrized PDEs

Niccolo Dal Santo, Simone Deparis, Andrea Manzoni, Alfio Quarteroni

http://mathicse.epfl.ch

Address: EPFL - SB - MATHICSE (Bâtiment MA)
Station 8 - CH-1015 - Lausanne - Switzerland

Phone: +41 21 69 37648

Multi space reduced basis preconditioners for large-scale parametrized PDEs

N. DAL SANTO[1], S. DEPARIS[1], A. MANZONI[1], A. QUARTERONI[1,2]

May 22, 2017

[1] CMCS, École Polytechnique Fédérale de Lausanne (EPFL),
Station 8, 1015 Lausanne, Switzerland.
[2]Mox-Laboratory for Modeling and Scientific Computing, Department of Mathematics, Politecnico di Milano, P.za Leonardo da Vinci 32, 20133 Milano, Italy (on leave).

Abstract

In this work we introduce a new two-level preconditioner for the efficient solution of large scale linear systems arising from the discretization of parametrized PDEs. The proposed preconditioner combines in a multiplicative way a reduced basis solver, which plays the role of coarse component, and a "traditional" fine grid preconditioner, such as one-level Additive Schwarz, block Gauss-Seidel or block Jacobi preconditioners. The coarse component is built upon a new Multi Space Reduced Basis (MSRB) method that we introduce for the first time in this paper, where a reduced basis space is built through the proper orthogonal decomposition (POD) algorithm at each step of the iterative method at hand, like the flexible GMRES method. MSRB strategy consists in building reduced basis (RB) spaces that are well-suited to perform a single iteration, by addressing the error components which have not been treated yet. The Krylov iterations employed to solve the resulting preconditioned system targets small tolerances with a very small iteration count and in a very short time, showing good optimality and scalability properties. Simulations are carried out to evaluate the performance of the proposed preconditioner in different large scale computational settings related to parametrized advection diffusion equations and compared with the current state of the art algebraic multigrid preconditioners.

1 Introduction

The repeated solution of parametric partial differential equations (PDEs), that is, PDEs depending on a vector of parameters, is computationally challenging. When using a high-fidelity numerical approximation method based on Galerkin or Petrov-Galerkin projection on a subspace V_h of dimension N_h (see e.g. [40]) we end up with a parametrized linear system of the form

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}), \tag{1}$$

where $\mathbf{u}_h(\boldsymbol{\mu})$, $\mathbf{f}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are N_h-dimensional vectors and $\mathbf{A}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix; $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p$ is a vector of p parameters describing physical and/or geometrical properties of the model. Solving such a problem for a huge number of parameter instances is essential when dealing with sensitivity analysis, uncertainty quantification for problems with random input data or PDE-constrained optimization. However, this may become a critical issue because of the extensive CPU time required by each query to the high-fidelity solver. The solution of the high-fidelity problem (1) indeed depends on the dimension N_h of the high-fidelity space, which can be of order 10^6 to 10^{10} in some extreme cases.

Problem (1) is usually solved by means of suitable preconditioned iterative methods, such as the preconditioned conjugate gradient (CG) or the preconditioned GMRES (see e.g. [42, 51, 57]) methods, whose cost per iteration is comparable to a matrix-vector multiplication; if suitably preconditioned, these methods provide scalable and optimal solvers. In general, a vast choice of preconditioners is currently available for many classes of problems: notable examples are domain decomposition (DD), see e.g. [43, 56, 54], or multilevel (ML) preconditioners, see e.g. [51, 54, 55]. However, these classical techniques do not generally take advantage of the parametric dependence of the PDE. Taking advantage of storing repeated solutions to similar systems can enhance efficiency in such a context. For instance, several Krylov-subspace recycling approaches have been introduced [50] to handle sequences of linear systems arising, e.g., from parametrized, time-dependent and/or nonlinear PDEs. The strategy consists in augmenting the usual Krylov subspace

with data retrieved from previous cycles (in the case of restarted algorithms) or solves (in the case of problems with both varying matrices and right hand sides). For instance, the first contributions in this field made use of the whole Krylov subspaces of previous solutions of linear systems, see e.g. [26, 45, 46, 48], yielding however a severe computational and memory effort, especially when the problem features a large dimension and a slow convergence. Consequently, research has focused on truncation methods that select a limited number of (significant) linear combinations of Krylov vectors. For the solution of a single linear system of equations, in [19, 20] the authors propose optimal truncation strategies of the GCR (generalized conjugate residual) method (GCRO), while in [34, 14, 24] deflation techniques to find an approximation of the eigenvectors associated to the extremal eigenvalues are employed. These techniques have been extended to the case of a sequence of linear systems with varying right hand sides in [52], where a deflated version of the CG algorithm is presented, and in [38] where the GCRO method is combined with deflated restarting for sequences of linear systems where both matrices and right hand sides vary.
Krylov subspace methods have been exploited in the context of reduced order modeling (ROM) to deal with sequences of single linear systems in [5] and in the iterative rational Krylov algorithm (IRKA) for sequences of dual linear systems in [1]. More recently, proper orthogonal decomposition (POD)-ROM has been successfully employed in [13] to truncate the augmented Krylov subspace and retain only the high-energy modes. This technique, suited for linear systems with symmetric matrices, allows to compute efficiently inexact (yet, very accurate) solutions. Although relying on reduced order modeling, the approach we propose in this paper exploits low-dimensional subspaces to build efficient preconditioners to speed up the solution of problems as (1), where both the matrix and the right hand side depend on the parameter $\boldsymbol{\mu}$. More specifically, we do not augment the Krylov subspace for the solution of any linear system; rather, we propose a new preconditioner which exploits ROM techniques to build an accurate coarse correction to speed up the solution of the iterative solver. Projection- and interpolatory-based ROM techniques have been extensively used in the past decade to construct efficient and accurate low-rank solvers for the solutions of large-scale parametrized systems, for an in-depth discussion see e.g. [41, 29, 2, 6] and references therein. In this work we employ the Reduced Basis (RB) method as particular case of ROM technique.

The RB method emerged as one of the most successful reduced order modeling paradigms for parametrized PDEs, and has been employed for multi-query problems such as input/output evaluations, sensitivity analysis, uncertainty quantification, PDE-constrained optimization, see e.g. [30, 53] and references therein. It has been successfully applied to elliptic problems, see e.g. [53, 39] and then extended to saddle-point [47], nonlinear [32, 33, 21, 22, 17], optimal control problems [37, 36], just to mention a few classes of problems in the context of time-independent PDEs. Given $\boldsymbol{\mu} \in \mathcal{D}$, the RB method seeks an approximation of the high-fidelity solution $\mathbf{u}_h(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})$ in a *reduced* space $V_N \subset V_h$ that is spanned by a set of N basis functions given by linear combinations of high-fidelity solutions corresponding to different instances of parameters $\mathbf{u}_h(\boldsymbol{\mu}_i)$, $i = 1, \dots, N$, where $N \ll N_h$. From an algebraic standpoint, after orthonormalizing of the RB functions, V_N can be represented by a matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, $\mathbf{V} = [\boldsymbol{\xi}_1 | \dots | \boldsymbol{\xi}_N]$, whose columns are orthonormal with respect to a prescribed scalar product. Finally, system (1) is replaced by a smaller one

$$\mathbf{V}^T \mathbf{A}_h(\boldsymbol{\mu}) \mathbf{V} \mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{V}^T \mathbf{f}_h(\boldsymbol{\mu}), \tag{2}$$

with $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ being the reduced solution, obtained by performing a projection onto the subspace V_N. We can introduce the reduced arrays, obtained from the corresponding high-fidelity arrays, as

$$\mathbf{A}_N(\boldsymbol{\mu}) = \mathbf{V}^T \mathbf{A}_h(\boldsymbol{\mu}) \mathbf{V} \in \mathbb{R}^{N \times N}, \qquad \mathbf{f}_N(\boldsymbol{\mu}) = \mathbf{V}^T \mathbf{f}_h(\boldsymbol{\mu}) \in \mathbb{R}^N. \tag{3}$$

Then, the reduced problem becomes

$$\mathbf{A}_N(\boldsymbol{\mu}) \mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu}). \tag{4}$$

The corresponding high-fidelity representation of the RB solution $\mathbf{u}_N(\boldsymbol{\mu})$ can be expressed as

$$\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{f}_N = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T\mathbf{f}_h(\boldsymbol{\mu}) \approx \mathbf{u}_h(\boldsymbol{\mu}). \tag{5}$$

We remark that the high-fidelity system (1) is large and sparse, whereas the reduced system (4) is small dense; usually the latter is solved using direct methods, since $N \ll N_h$. Indeed, it is well-known that in many situations the RB method provides an exponential decay of the approximation error with respect to the dimension N of the RB space; however, the decay ratio is considerably affected by the parametrization of the problem (both in terms of number and nature of parameters), the regularity of the parameter-to-solution map, the physical nature of the problem and, ultimately, the Kolmogorov n-width of the solution manifold, an intrisic property of the problem. For instance, advection-diffusion problems where the advection is highly variable because of the $\boldsymbol{\mu}$-dependence, and possibly dominant, may yield to a slower decay of the error with

respect to N.
Another key factor required for RB efficiency is the affine parameter dependence of both operators and data (see Appendix A, equation (57)). If these assumptions are not verified, an approximated affine decomposition, up to a certain tolerance, must be recovered through proper techniques which could heavily limit the accuracy or the efficiency of the RB method.

The aim of this work is to present a new class of two-level preconditioners for parameter dependent linear systems as (1) arising from the numerical approximation of second order elliptic PDEs, with focus on advection-diffusion (AD) problems. Our preconditioners are constructed upon the combination of the RB method, which plays the role of coarse component, and a fine preconditioner, e.g. Gauss-Seidel, Jacobi or one-level additive Schwarz preconditioners. Very few attempts to link RB and preconditioning techniques have been made so far: some works have proposed *ad hoc* preconditioning techniques for reduced systems arising from the RB method, see e.g. [16] in the case of the reduced collocation method when dealing with PDEs with random input data or [23] in the case of the Galerkin RB method. Concerning the preconditioning of parametrized linear system, remarkable efforts have been devoted to preconditioning strategies for shifted linear system. At first, these techniques compute a preconditioner for the unshifted high-fidelity matrix, and then they suitably modify it for the shifted matrix. This has proven to be particularly helpful when employing time-advancing schemes with adaptively chosen time steps, see [4, 7, 27]. More recently, techniques to deal with sequences of (not necessarily shifted) linear systems, which compute approximate inverse (AINV) preconditioners by interpolation, have been developed in [8]. Furthermore, in [58] a preconditioner for the parametrized high-fidelity problem (1) which relies on an interpolation of the matrix inverse based on a pre-computed basis of matrix inverses corresponding to selected values of the parameter has been introduced. This latter method stores the basis of inverted matrices as exact factorizations, thus yielding a huge amount of storage memory, and is computationally efficient only for relatively small problems. Finally, in [31], a low-rank tensor approximation of $\mathbf{u}_h(\boldsymbol{\mu})$ has been exploited to present low-rank tensor variants of short-recurrence Krylov subspace methods.

Alternatively to the techniques above, the preconditioners we propose in this paper combine in a multiplicative way existing preconditioners on the given (fine) finite element mesh with a coarse RB solver. The former guarantees the nonsingularity of the resulting preconditioner, whereas the latter can be regarded as a coarse correction built upon the RB method meant to boost the convergence of Krylov iterations. The RB problems must be small in order for their solution to be computationally cheap; to take the best advantage from the ROM, we rely on a sequence of RB spaces which are iteration-dependent, thus leading to a procedure that involves the construction of several RB spaces. In particular, the k-th space is trained on the error equation corresponding to the k-th iteration of the iterative method, up to a prescribed tolerance $\delta_{RB,k}$. We refer to this (new) approach as Multi Space Reduced Basis (MSRB) preconditioning method. We then show that an iterative method for the large-scale linear system preconditioned with the MSRB preconditioner requires very few steps to achieve any desired target tolerance, since at every step the error equation is solved approximately, yet with high accuracy.

We point out that, when dealing with parameter-dependent linear systems, classical preconditioners may have performances which differ according to the value of the parameter, e.g. in advection-diffusion problems when, for certain values of the parameter, the model is advection-dominated or includes a strong anisotropy effect. On the other hand, the use of RB coarse components built upon the parametrized problem at hand (and trained on the whole parameter range) allows to gain robustness across the whole parameter space, meaning that the preconditioner efficiency is almost constant for all parameter values .

The coarse component built upon the RB method depends essentially on the underlying physical problems, and it is shown to be independent of the size of the high-fidelity discretization. For this reason, employing a parametrized preconditioner allows to solve system (1) very rapidly for any new parameter instance, even for problems with a large number of degrees of freedom. As iterative solvers for problem (1), we first employ a Richardson iteration, and then adapt the method to the flexible GMRES (FGMRES). In particular, we employ the former to analyze the properties of the preconditioner and show how the RB method enters into play, whereas FGMRES is meant to provide a very efficient tool to tackle large scale problems arising from real applications.

In the numerical tests presented in this work, we compare the iteration counts and the computational times provided by our MSRB preconditioner with the ones obtained by relying on an algebraic multigrid preconditioned Krylov method and the ML-preconditioned GCRO-DR method (both built from the Belos and ML package of Trilinos, [28]) for advection-diffusion problems. We show that the MSRB preconditioner is a valuable option in some relevant and involved modeling and numerical settings, namely when the problem is advection-dominated and/or includes anisotropy, or when the high-fidelity dimension N_h is very large, up to several millions.

The structure of the paper is as follows. In Section 2 we present the class of problems we deal with,

and how to build a MSRB preconditioner, motivating the introduction of an approach involving several RB spaces for the Richardson method and detailing the properties of the resulting preconditioner; then we extend the MSRB preconditioner to the case of the FGMRES methods. In Section 3 we test the MSRB preconditioner on 3D problems governed by second-order advection-diffusion equations, reporting results for several modeling and numerical settings; finally, in Section 4 we draw some conclusions and possible extensions. In the Appendix A we report a review of the classic RB method, which is meant to provide a basic background to those readers less-acquainted with this topic.

2 Multi space RB preconditioners for parametrized PDEs

In this paper we focus on parametrized linear elliptic second-order PDEs. Let us denote by $\mathcal{D} \subset \mathbb{R}^p$, $p \geq 1$, the parameter space and by $\boldsymbol{\mu} \in \mathcal{D}$ a parameter vector encoding physical and/or geometrical properties of the problem. Our goal is to solve a parametrized PDE which under weak form reads as: given $\boldsymbol{\mu} \in \mathcal{D}$, find $u = u(\boldsymbol{\mu}) \in V = V(\Omega)$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \qquad \forall v \in V, \tag{6}$$

being $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, a regular domain and $V = V(\Omega)$ a Hilbert space. We further assume that for any $\boldsymbol{\mu} \in \mathcal{D}$, $a(\cdot, \cdot; \boldsymbol{\mu})$ is a bilinear, continuous and coercive form, and $f(\cdot; \boldsymbol{\mu})$ a linear and continuous form. Under these hypotheses, the Lax-Milgram lemma (see e.g. [40]) ensures the existence and uniqueness of a solution to problem (6), for any $\boldsymbol{\mu} \in \mathcal{D}$.

Solving problem (6) requires the use of suitable numerical approximation techniques, here called *high-fidelity* (or *full order*) approximations, providing a discretized solution which is close to the exact solution up to a (controllable) discretization error. Examples are the finite element (FE) method [10, 25, 40] and spectral methods [11, 40]. All these approaches are built upon the use of a finite dimensional space $V_h \subset V$, with $dim(V_h) = N_h$, and require to find an approximate solution $u_h(\boldsymbol{\mu})$ to (6) by solving the following Galerkin problem: given $\boldsymbol{\mu} \in \mathcal{D}$, find $u_h(\boldsymbol{\mu}) \in V_h$ such that:

$$a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) \qquad \forall v_h \in V_h, \tag{7}$$

which can be equivalently expressed as (1) in algebraic form.

Our goal is to exploit the RB method to build efficient preconditioners for the iterative solution of (1) featuring uniform performances in the parameter space.

2.1 Multi space RB preconditioners

In this section, we first detail the construction of the preconditioner to be used for Richardson iterations; this is primarily done for methodological and theoretical purposes, since it allows to amenably derive the method and compute theoretical estimates. Consequently, we turn our attention to the FGMRES method in section 2.2.

2.1.1 Preconditioning the Richardson method

Given two matrices $\mathbf{Q}_1 = \mathbf{Q}_1(\boldsymbol{\mu})$, $\mathbf{Q}_2 = \mathbf{Q}_2(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$, a multiplicative Richardson iteration for the system (1) can be expressed as

$$\begin{cases} \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}) &= \mathbf{u}^{(k-1)}(\boldsymbol{\mu}) + \mathbf{Q}_1(\boldsymbol{\mu})\mathbf{r}^{(k-1)}(\boldsymbol{\mu}), \\ \mathbf{u}^{(k)}(\boldsymbol{\mu}) &= \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}) + \mathbf{Q}_2(\boldsymbol{\mu})\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots \end{cases} \tag{8}$$

where $\mathbf{u}^{(k)} = \mathbf{u}^{(k)}(\boldsymbol{\mu})$ is the $\boldsymbol{\mu}-$dependent iterate at the step k, and $\mathbf{r}^{(k)} = \mathbf{r}^{(k)}(\boldsymbol{\mu})$ is the corresponding high-fidelity residual of the Richardson method

$$\mathbf{r}^{(k)}(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(k)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots.$$

Equations (8) can be equivalently formulated as a single iteration

$$\mathbf{u}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}^{(k-1)}(\boldsymbol{\mu}) + \mathbf{Q}(\boldsymbol{\mu})\mathbf{r}^{(k-1)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots, \tag{9}$$

where $\mathbf{Q}(\boldsymbol{\mu})$ in (9) is defined as

$$\mathbf{Q}(\boldsymbol{\mu}) = \mathbf{Q}_1(\boldsymbol{\mu}) + \mathbf{Q}_2(\boldsymbol{\mu}) - \mathbf{Q}_2(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})\mathbf{Q}_1(\boldsymbol{\mu}). \tag{10}$$

If $\mathbf{Q}(\boldsymbol{\mu})$ is non singular, (9) can be regarded as a Richardson iteration, with acceleration constant equal to 1, for the preconditioned system

$$\mathbf{Q}(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{Q}(\boldsymbol{\mu})\mathbf{f}_h(\boldsymbol{\mu}), \tag{11}$$

where the preconditioner is $\mathbf{Q}^{-1}(\boldsymbol{\mu})$.

The main idea of our approach is to exploit a standard two level domain decomposition approach relying on a RB solver as coarse (low-rank) component. Therefore, an intuitive choice for the Richardson method (8) would be to take

$$\mathbf{Q}_1(\boldsymbol{\mu}) = \mathbf{P}_h^{-1}(\boldsymbol{\mu}), \qquad \mathbf{Q}_2(\boldsymbol{\mu}) = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T, \tag{12}$$

where $\mathbf{P}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is a nonsingular matrix which plays the role of fine preconditioner, which can be chosen among all existing preconditioners, and $\mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T$ is the RB coarse component.

However, we have experienced that the convergence rate of (8) is not faster than the one obtained by setting $\mathbf{Q}_2(\boldsymbol{\mu}) = 0$ (i.e. just using $\mathbf{P}_h(\boldsymbol{\mu})$ as preconditioner) and taking the RB solution $\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})$ as initial guess $\mathbf{u}^{(0)}(\boldsymbol{\mu})$. Indeed, determining

$$\mathbf{Q}_2(\boldsymbol{\mu})\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}) \tag{13}$$

can be reinterpreted as the approximate solution of the error equation

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}), \tag{14}$$

through the RB method, where $\mathbf{e}^{(k-1/2)} = \mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu})$. In other words, by computing the quantity in (13), we are implicitly seeking an approximation of $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu})$ in the RB space V_N, that is, expressed as a linear combination of basis functions obtained from snapshots of the high-fidelity problem (1). The main issue related with this approach is that the employed ROM (i.e. the RB space V_N) is tailored only for equation (1), while we are trying to use it to solve approximately equation (14), which features the same stiffness matrix $\mathbf{A}_h(\boldsymbol{\mu})$ but a different right hand side. Therefore, the space V_N is not well suited to approximate the solution of problem (14), yielding a very poor numerical approximation of the error, as confirmed by numerical experiments.

We thus introduce at each step k a new RB space that is trained on equation (14), and where a better approximation of $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu})$ can be found. Since the error highly depends on the iterate k, it makes sense to introduce a different RB space V_{N_k} at every iteration k, generated by high-fidelity solutions of problem (14), that is

$$V_{N_k} = span\Big\{\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_j)\Big\}_{j=1}^{N_k}, \tag{15}$$

where $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_j)$, $j = 1, \dots, N_k$ are the errors at the $(k-1/2)$-th iteration, computed for (properly chosen) instances of the parameters $\boldsymbol{\mu}_j$, $j = 1, \dots, N_k$. Following the standard RB method, we can construct the matrices

$$\mathbf{V}_k = [\boldsymbol{\xi}_1^k | \dots | \boldsymbol{\xi}_N^k], \qquad \mathbf{A}_{N_k}(\boldsymbol{\mu}) = \mathbf{V}_k^T\mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}_k, \tag{16}$$

where $\{\boldsymbol{\xi}_j^k\}_{j=1}^{N_k}$ denotes an orthonormalized basis for V_{N_k}, and write the MSRB-preconditioned Richardson iterations as

$$\begin{cases} \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}) &= \mathbf{u}^{(k-1)}(\boldsymbol{\mu}) + \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{r}^{(k-1)}(\boldsymbol{\mu}) \\ \mathbf{u}^{(k)}(\boldsymbol{\mu}) &= \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}) + \mathbf{Q}_{N_k}\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots, \end{cases} \tag{17}$$

where $\mathbf{Q}_{N_k}(\boldsymbol{\mu}) = \mathbf{V}_k\mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T$. The formulation (17) leads to

$$\mathbf{u}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}^{(k-1)}(\boldsymbol{\mu}) + \mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})\mathbf{r}^{(k-1)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots, \tag{18}$$

where the matrix $\mathbf{Q}_{\mathrm{MSRB},k} = \mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ (replacing $\mathbf{Q}(\boldsymbol{\mu})$ in (9)) is now

$$\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu}) = \mathbf{P}_h^{-1}(\boldsymbol{\mu}) + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big), \tag{19}$$

and can be regarded as a multiplicative combination of $\mathbf{P}_h^{-1}(\boldsymbol{\mu})$ and $\mathbf{Q}_{N_k}(\boldsymbol{\mu})$.

Given the error $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu})$, its RB approximation onto V_{N_k} is defined by $\mathbf{e}_{N_k}^{(k-1/2)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_k}$ such that

$$\mathbf{e}_{N_k}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}), \tag{20}$$

and we highlight that its high-fidelity representation $\mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is computed in (17) as

$$\mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{V}_k\mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}(\boldsymbol{\mu}). \tag{21}$$

In this setting, we take as initial guess the (standard) RB approximation $\mathbf{u}^{(0)} = \mathbf{u}^{(0)}(\boldsymbol{\mu}) = \mathbf{V}_0\mathbf{A}_{N_0}^{-1}(\boldsymbol{\mu})\mathbf{V}_0^T\mathbf{f}_h(\boldsymbol{\mu})$, and set $V_{N_0} = V_N$, i.e. the first RB space is the one provided by the standard RB method. The subsequent spaces V_{N_k}, $k \geq 1$, aim at damping those components of the error that have not been cured by the previous RB iterations and cannot be addressed by the application of $\mathbf{P}_h(\boldsymbol{\mu})$; they are therefore directly constructed on the error equation (14).

2.1.2 Nonsingularity of the resulting preconditioner

We show in this section that the matrix $\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ is invertible. Given a subspace $W \subset \mathbb{R}^{N_h}$ such that $dim(W) = M$ and a basis $\{\mathbf{w}_j\}_{j=1}^M$ such that $W = span\{\mathbf{w}_j, j = 1, \ldots, M\}$, we denote by $W^\perp$ the orthogonal complement of W and by $\mathbf{W} \in \mathbb{R}^{N_h \times M}$, $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$, the matrix of basis vectors. Moreover, given any 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 2.1. *Let W be a M-dimensional subspace of $\mathbb{R}^{N_h}$, $\{\mathbf{w}_j\}_{j=1}^M$ a basis thereof and $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M] \in \mathbb{R}^{N_h \times M}$. Moreover, let $\mathbf{B}$ be a nonsingular $N_h \times N_h$ matrix and assume that $\mathbf{W}^T\mathbf{B}\mathbf{W}$ is nonsingular. Then the following implication holds:*

$$\mathbf{x} \in \mathbf{B}W \quad \textit{and} \quad \mathbf{W}^T\mathbf{x} = \mathbf{0} \quad \Rightarrow \quad \mathbf{x} = \mathbf{0}. \tag{22}$$

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

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

As $\mathbf{W}^T\mathbf{B}\mathbf{W} \in \mathbb{R}^{M \times M}$ is invertible, $\mathbf{z}_M = \mathbf{0}$. Finally, we have

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

which implies $\mathbf{x} = \mathbf{0}$ thanks to the nonsingularity of $\mathbf{B}$. □

In the following we employ Lemma 2.1 by taking $W = V_{N_k}$, $\mathbf{B} = \mathbf{P}_h(\boldsymbol{\mu})$, $\mathbf{W} = \mathbf{V}_k$ in order to prove that $\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ is nonsingular. To this aim, we define

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

Theorem 2.1. *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 $\mathbf{V}_k^T\mathbf{P}_h(\boldsymbol{\mu})\mathbf{V}_k$ is nonsingular. Then the matrix $\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ is nonsingular.*

Proof. We want to prove that if $\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})\mathbf{x} = \mathbf{0}$, then it must be $\mathbf{x} = \mathbf{0}$. Since any $\mathbf{x} \in \mathbb{R}^{N_h}$ can be expressed as $\mathbf{x} = \mathbf{x}_{/\!/} + \mathbf{x}_\perp$, where $\mathbf{x}_{/\!/} \in V_{N_k}^{\mathbf{P}_h /\!\!/}$, $\mathbf{x}_\perp \in V_{N_k}^{\mathbf{P}_h \perp}$, we first compute the result of the application of $\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ on $\mathbf{x}_{/\!/}$:

$$\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})\mathbf{x}_{/\!/} = \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_{/\!/} + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{x}_{/\!/}$$

Being $\mathbf{x}_{/\!/} \in V_{N_k}^{\mathbf{P}_h /\!\!/}$, we can write $\mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_{/\!/} = \mathbf{V}_k\mathbf{z}_N(\boldsymbol{\mu})$ for some $\mathbf{z}_{N_k}(\boldsymbol{\mu}) \in \mathbb{R}^{N_k}$, yielding

$$\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})\mathbf{x}_{/\!/} = \mathbf{V}_k\mathbf{z}_N(\boldsymbol{\mu}) + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{x}_{/\!/} - \mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}_k\mathbf{z}_N(\boldsymbol{\mu}) = \mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{x}_{/\!/}, \tag{23}$$

since $\mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}_k\mathbf{z}_N = \mathbf{V}_k\mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T\mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}_k\mathbf{z}_N = \mathbf{V}_k\mathbf{z}_N$. As of the component $\mathbf{x}_\perp$, we have

$$\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})\mathbf{x}_\perp = \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_\perp + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{x}_\perp,$$

which leads to

$$\mathbf{0} = \mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})\mathbf{x} = \mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{x}_{/\!/} + \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_\perp + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{x}_\perp. \tag{24}$$

By rewriting equation (24) as follows

$$\mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{x}_{/\!/} + \mathbf{x}_\perp + \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_\perp\Big) = -\mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_\perp, \tag{25}$$

we can notice that the left hand side is an element of the space V_{N_k}, whereas the right hand side is an element of its orthogonal complement $V_{N_k}^\perp$, so that the only way these two elements are equal is when they are both zero. Being $\mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{x}_\perp = \mathbf{0}$, the nonsingularity of $\mathbf{P}_h(\boldsymbol{\mu})$ yields $\mathbf{x}_\perp = \mathbf{0}$, allowing us to rewrite equation (25) as

$$\mathbf{0} = \mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{x}_{/\!/} = \mathbf{V}_k\mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T\mathbf{x}_{/\!/}. \tag{26}$$

The columns of $\mathbf{V}_k$ being linearly independent, equation (26) yields

$$\mathbf{0} = \mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T\mathbf{x}_{/\!/}, \tag{27}$$

which, thanks to the non singularity of the RB matrix $\mathbf{A}_{N_k}(\boldsymbol{\mu})$, see Appendix A, implies

$$\mathbf{V}_k^T\mathbf{x}_{/\!/} = \mathbf{0}. \tag{28}$$

Finally, by applying Lemma 2.1 with $W = V_{N_k}$, $\mathbf{W} = \mathbf{V}_k$ and $\mathbf{B} = \mathbf{P}_h(\boldsymbol{\mu})$, we obtain that $\mathbf{x}_{/\!/} = \mathbf{0}$, and thus the thesis. □

Now, since the matrix $\mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ is invertible, we can define the MSRB preconditioner as

$$\mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu}) = \mathbf{Q}_{\mathrm{MSRB},k}^{-1}(\boldsymbol{\mu}). \tag{29}$$

The MSRB preconditioner $\mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ resulting from the combination of $\mathbf{P}_h(\boldsymbol{\mu})$ and $\mathbf{Q}_{N_k}(\boldsymbol{\mu})$, leads to the generation of a k-dependent RB space (hereon also called level) V_{N_k}. At each iteration k, we seek an approximation of the error $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu})$ in V_{N_k}. Each V_{N_k}, $k = 0, 1, \ldots$ is associated to a pair $(N_k, \delta_{RB,k})$; N_k identifies the number of basis functions in the space V_{N_k} (and therefore its dimension) and $\delta_{RB,k}$ is the tolerance prescribed to construct the space V_{N_k}, e.g. with a greedy algorithm or the POD method, see e.g. [41]. In particular, in this work we employ the POD for the purpose of space construction, see Appendix A for further details. In analogy with the standard RB method, the MSRB preconditioner can be split in an *offline* and an *online* stage. In the former we construct the reduced structures that are needed by the algorithm (17), which is then employed in the latter to solve problem (1) for any new parameter instance.

Remark 2.1. *The assumption that the matrix $\mathbf{V}_k^T\mathbf{P}_h(\boldsymbol{\mu})\mathbf{V}_k$ to be nonsingular is fairly mild. For example it is satisfied for any matrix $\mathbf{P}_h(\boldsymbol{\mu})$ such that $\mathbf{x}^T\mathbf{P}_h(\boldsymbol{\mu})\mathbf{x} \neq 0$ for any $\mathbf{x} \neq \mathbf{0}$. This is indeed the case for the classical preconditioners, like Jacobi, Gauss-Seidel or Additive Schwarz preconditioners.*

2.1.3 Convergence results

In this section we prove a priori estimates of the error and the residual decay for the Richardson method (17). For the ease of notation, hereon we omit the $\boldsymbol{\mu}$−dependence and denote by $\mathbf{I}_{N_h}$ the identity $N_h \times N_h$ matrix.

Proposition 2.1. *For any vector norm $\|\cdot\|$, let the spaces V_{N_k} $k = 1, \ldots, L$ satisfy the following relation*

$$\|\mathbf{e}^{(k-1/2)} - \mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}\| \leq \delta_k\|\mathbf{e}^{(k-1/2)}\| \qquad k = 1, \ldots, L, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{30}$$

for given tolerances δ_k for $k = 1, \ldots, L$. Moreover, let the assumption of Theorem 2.1 be satisfied. Then the following estimate holds:

$$\|\mathbf{e}^{(k)}\| \leq C^k\delta\|\mathbf{e}^{(0)}\|, \qquad k = 1, \ldots, L, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{31}$$

with $C = \left\|\mathbf{I}_{N_h} - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})\right\|$ and $\delta = \prod_{j=1}^k \delta_j$.

Proof. We consider equations (17). The error $\mathbf{e}^{(k)} = \mathbf{u}_h - \mathbf{u}^{(k)}$ at iteration k can be computed as

$$\mathbf{e}^{(k)} = \Big(\mathbf{I}_{N_h} - \mathbf{Q}_{N_k}\mathbf{A}_h\Big)\mathbf{e}^{(k-1/2)} = \mathbf{e}^{(k-1/2)} - \mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)},$$

where the equation (21) has been used. Then

$$\begin{aligned}\Big\|\mathbf{e}^{(k)}\Big\| &= \Big\|\Big(\mathbf{I}_{N_h} - \mathbf{Q}_{N_k}\mathbf{A}_h\Big)\mathbf{e}^{(k-1/2)}\Big\| \leq \delta_k\Big\|\mathbf{e}^{(k-1/2)}\Big\| \\ &= \delta_k\Big\|\Big(\mathbf{I}_{N_h} - \mathbf{P}_h^{-1}\mathbf{A}_h\Big)\mathbf{e}^{(k-1)}\Big\| \leq \delta_k\Big\|\mathbf{I}_{N_h} - \mathbf{P}_h^{-1}\mathbf{A}_h\Big\|\Big\|\mathbf{e}^{(k-1)}\Big\|.\end{aligned}$$

By proceeding recursively we obtain (31). □

A similar result holds for the residuals of the Richardson method.

Proposition 2.2. *For any vector norm $\|\cdot\|$, let the spaces V_{N_k} $k = 1, \ldots, L$ satisfy the following relation*

$$\|\mathbf{r}^{(k-1/2)} - \mathbf{A}_h\mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}\| \leq \delta_k\|\mathbf{r}^{(k-1/2)}\| \qquad k = 1, \ldots, L, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{32}$$

and given tolerances δ_k for $k = 1, \ldots, L$. Moreover, let the assumption of Theorem 2.1 be satisfied. Then the following estimate holds:

$$\|\mathbf{r}^{(k)}\| \leq C^k\delta\|\mathbf{r}^{(0)}\|, \qquad k = 1, \ldots, L, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{33}$$

with $C = \Big\|\mathbf{I}_{N_h} - \mathbf{P}_h^{-1}\mathbf{A}_h\Big\|$ and $\delta = \prod_{j=1}^{k}\delta_j$.

Proof. We consider equations (17). The residual at iteration k can be computed as

$$\begin{aligned}\mathbf{r}^{(k)} &= \Big(\mathbf{I}_{N_h} - \mathbf{A}_h\mathbf{Q}_{N_k}\Big)\mathbf{r}^{(k-1/2)} = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h\mathbf{V}_k\mathbf{A}_{N_k}^{-1}\mathbf{V}_k^T\Big)\mathbf{r}^{(k-1/2)} \\ &= \mathbf{r}^{(k-1/2)} - \mathbf{A}_h\mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}.\end{aligned} \tag{34}$$

Thanks to (32) we obtain

$$\begin{aligned}\Big\|\mathbf{r}^{(k)}\Big\| &= \Big\|\Big(\mathbf{I}_{N_h} - \mathbf{A}_h\mathbf{Q}_{N_k}\Big)\mathbf{r}^{(k-1/2)}\Big\| \leq \delta_k\Big\|\mathbf{r}^{(k-1/2)}\Big\| \\ &= \delta_k\Big\|\Big(\mathbf{I}_{N_h} - \mathbf{A}_h\mathbf{P}_h^{-1}\Big)\mathbf{r}^{(k-1)}\Big\| \leq \delta_k\Big\|\mathbf{I}_{N_h} - \mathbf{A}_h\mathbf{P}_h^{-1}\Big\|\Big\|\mathbf{r}^{(k-1)}\Big\|.\end{aligned}$$

By proceeding recursively we end up with (33). □

Remark 2.2. *We underline that the hypothesis* (30) *of Proposition 2.1 holds only for a training set $\Xi_{train} \subset \mathcal{D}$ when the space V_{N_k} are constructed, for instance, relying upon a greedy algorithm with a prescribed tolerance $\delta_k = \delta_{RB,k}$ on the error and $\|\cdot\| = \|\cdot\|_{\mathbf{Y}_h}$, where $\mathbf{Y}_h$ is a symmetric positive definite matrix used to orthonormalize the reduced basis functions. On the other hand, the hypothesis* (32) *holds for $\Xi_{train} \subset \mathcal{D}$ if we build the spaces V_{N_k} upon a weak greedy algorithm with a prescribed tolerance $\delta_k = \delta_{RB,k}$ on the residual and $\|\cdot\| = \|\cdot\|_{\mathbf{Y}_h^{-1}}$. If we employ POD with a prescribed tolerance $\delta_{RB,k}$ and the norm $\|\cdot\|_{\mathbf{Y}_h}$ for the sake of space construction (see Appendix A), neither hypothesis* (30) *or* (32) *hold, even if they are assessed from a numerical standpoint. In fact, by solving the reduced problem relying on these reduced space provides an approximate solution $\mathbf{e}_{N_k}^{(k-1/2)}$ whose corresponding relative error $\|\mathbf{e}^{(k-1/2)} - \mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}\|_{\mathbf{Y}_h}/\|\mathbf{e}^{(k-1/2)}\|_{\mathbf{Y}_h}$ and residual $\|\mathbf{r}^{(k-1/2)} - \mathbf{A}_h\mathbf{V}_k\mathbf{e}_{N_k}^{(k-1/2)}\|_{\mathbf{Y}_h^{-1}}\|/\mathbf{r}^{(k-1/2)}\|_{\mathbf{Y}_h^{-1}}$ are of the order of $\delta_{RB,k}$.*

Propositions 2.1 and 2.2 state that the error $\mathbf{e}^{(k)}(\boldsymbol{\mu})$ and the residual $\mathbf{r}^{(k)}(\boldsymbol{\mu})$ of the Richardson method decay as the product of the tolerances $\delta_{RB,j}$, $j = 0, 1, \ldots$ used to build the reduced spaces. If we employ a stopping criterion based on the relative residual for the Richardson method

- this means that, given a tolerance ϵ_r, it reaches convergence at iteration m such that

$$\frac{\|\mathbf{r}^m(\boldsymbol{\mu})\|_2}{\|\mathbf{f}_h(\boldsymbol{\mu})\|_2} \leq \epsilon_r, \tag{35}$$

- we must build the RB spaces $V_{N_0}, \ldots, V_{N_k}$, such that

$$\delta = \prod_{j=0}^{k}\delta_{RB,j} \leq \varepsilon_r. \tag{36}$$

In other words, we require that the combination of all RB spaces yields an error which is lower than or equal to the target tolerance ε_r of the Richardson method.

In the algorithm we propose, we employ POD to build the basis for each reduced space. The construction of the spaces is performed recursively: at first we choose n_s values of the parameter $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$ and compute the snapshots $\{\mathbf{u}_h(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ as the high-fidelity solutions of (1) for $\boldsymbol{\mu} = \boldsymbol{\mu}_i$, $i = 1, \dots, n_s$. Following the standard RB method, we build upon them the first space V_{N_0} by performing POD on this set of snapshots. With the aim of building the subsequent spaces, we express the solution of problem (14) as follows:

$$\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_i) = \mathbf{u}_h(\boldsymbol{\mu}_i) - \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}_i) = \mathbf{e}^{(k-1)}(\boldsymbol{\mu}_i) - \mathbf{P}_h^{-1}(\boldsymbol{\mu}_i)\mathbf{r}^{(k-1)}(\boldsymbol{\mu}_i) \quad \forall\, i = 1, \dots, n_s. \tag{37}$$

Then, given the spaces $V_{N_0}, \dots, V_{N_k}$, we compute the snapshots errors $\{\mathbf{e}^{(k+1/2)}(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ through the relation (37), and construct the space $V_{N_{k+1}}$ space by performing POD on those snapshots. We highlight that the construction of the k-th space, employing equation (37), does not require to solve any additional linear system. In order to design our algorithm, a POD approach has been preferred to a (weak) greedy approach because of the intrinsic nonaffinity of $\mathbf{P}_h^{-1}(\boldsymbol{\mu})$, that appears in relation (37). Indeed, a (weak) greedy algorithm would build the reduced space relying on a fast evaluation of the error (or a residual-based a posteriori error bound) for a large number of offline parameters in a training set Ξ_{train}, typically computed with N_h-independent routines. On the other hand, computing the error or the residual for the equation (37) requires N_h-dependent operations, which would yield extremely huge offline costs for each $\boldsymbol{\mu} \in \Xi_{train}$. Relying on a POD approach makes the proposed technique also feasible in view of more involved applications (e.g. nonlinear problems) where residual-based a posteriori error bounds are not available.

Regarding the choice of the tolerances $\delta_{RB,k}$, $k = 0, 1, \dots$, (and, consequently, of the number N_k, $k = 0, 1, \dots$, of basis functions) for each RB space, we can follow two approaches:

- *fixed space accuracy*: we build each RB space prescribing the same tolerance δ_{RB}, i.e. $\delta_{RB,k} = \delta_{RB}$, $k = 0, 1, \dots$;
- *fixed space dimension*: we build each RB space prescribing the same space dimension N, i.e. $N_k = N$, $k = 0, 1, \dots$.

Since we need to construct a sufficiently large number of spaces such that inequality (36) is satisfied, in the former approach we shall implicitly fix the number of spaces larger than $\lceil \log(\varepsilon_r)/\log(\delta_{RB}) \rceil$, which however may lead to a huge number of RB functions employed at each RB space. In the latter, instead, we are not limiting the number of spaces. The detailed algorithms corresponding to these two approaches are reported in Algorithms 1 and 2, respectively. In Section 3 we report results for both these approaches. Once the spaces V_{N_k}, $k = 0, 1, \dots, L-1$ have been generated, it is possible to solve the high-fidelity system (1) by Richardson iterations (17), which are expected to converge in less than L iterations. However, since POD does not explicitly provide any error bound depending on $\delta_{RB,k}$, the number of iterations may in practice exceed L, in which case one can choose either to reuse the space $V_{N_{L-1}}$, or to drop the second step of (17) for the remaining iterations.

Algorithm 1 MSRB - Fixed Accuracy

procedure MSRB-FIXEDACCURACY($\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}, \varepsilon_r, \delta_{RB}$)
  Set the number of RB spaces $L = \lceil \log(\varepsilon_r)/\log(\delta_{RB}) \rceil$
  Compute the high-fidelity solutions $\{\mathbf{u}_h(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ and set $\mathbf{S} = [\mathbf{u}_h(\boldsymbol{\mu}_1), \dots, \mathbf{u}_h(\boldsymbol{\mu}_{n_s})]$
  Build the basis $\mathbf{V}_0 = POD(\mathbf{S}, \delta_{RB})$
  for $k = 1, \dots, L-1$ **do**
    Compute new snapshots $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_i) = \mathbf{e}^{(k-1)}(\boldsymbol{\mu}_i) - \mathbf{P}_h^{-1}(\boldsymbol{\mu}_i)\mathbf{r}^{(k-1)}(\boldsymbol{\mu}_i)$
    Set $\mathbf{S} = [\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_1), \dots, \mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_{n_s})]$
    Build the new basis $\mathbf{V}_k = POD(\mathbf{S}, \delta_{RB})$
  end for
end procedure

2.2 Multispace RB preconditioners for flexible GMRES

In the previous Section, our MSRB preconditioner has been built over a Richardson iteration mainly for illustrative reasons. In order to use a more efficient Krylov iterative method, we opt instead for the flexible GMRES method, FGMRES [49], since the MSRB preconditioner changes at each iteration. Indeed, the (classical) preconditioned GMRES algorithm does not ensure convergence in the case the preconditioner

Algorithm 2 MSRB - Fixed Dimension

procedure MSRB-FIXEDDIMENSION($\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}, \varepsilon_r, N$)
  Compute the high-fidelity solutions $\{\mathbf{u}_h(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ and set $\mathbf{S} = [\mathbf{u}_h(\boldsymbol{\mu}_1), \dots, \mathbf{u}_h(\boldsymbol{\mu}_{n_s})]$, $k = 0$
  while $\prod_k \delta_{RB,k} > \varepsilon_r$ **do**
    Build the new basis $\mathbf{V}_k = POD(\mathbf{S}, N)$ and set $k = k + 1$
    Compute new snapshots $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_i) = \mathbf{e}^{(k-1)}(\boldsymbol{\mu}_i) - \mathbf{P}_h^{-1}(\boldsymbol{\mu}_i)\mathbf{r}^{(k-1)}(\boldsymbol{\mu}_i)$
    Set $\mathbf{S} = [\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_1), \dots, \mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_{n_s})]$
  end while
end procedure

changes at every iteration, while its flexible variant allows to precondition the system with an iteration-dependent operator. For ease of presentation, we report in Algorithm 3 the version of this method taken from [51].

Algorithm 3 Flexible GMRES (as formulated in [51])

procedure FGMRES($\mathbf{A}$, $\mathbf{b}$, $\mathbf{u}_0$, $\{\mathbf{M}_k\}_k, m$)
  Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{u}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$
  for $k = 1, \dots, m$ **do**
    Compute $\mathbf{z}_k = \mathbf{M}_k^{-1}\mathbf{v}_k$
    Compute $\mathbf{w} = \mathbf{A}\mathbf{z}_k$
    for $j = 1, \dots, k$ **do**
      $h_{j,k} = (\mathbf{w}, \mathbf{v}_j)$
      $\mathbf{w} = \mathbf{w} - h_{j,k}\mathbf{v}_j$
    end for
    Compute $h_{k+1,k} = \|\mathbf{w}\|$ and $\mathbf{v}_{k+1} = \mathbf{w}/h_{k+1,k}$
    Define $\mathbf{Z}_m = [\mathbf{z}_1, \dots, \mathbf{z}_m]$, $\tilde{\mathbf{H}}_m = \{h_{j,k}\}_{1 \le j \le k+1;\, 1 \le k \le m}$
  end for
  Compute $\mathbf{y}_m = \arg\min_{\mathbf{y} \in \mathbb{R}^m} \|\beta\mathbf{e}_1 - \tilde{\mathbf{H}}_m\mathbf{y}\|_2$ and $\mathbf{u}_m = \mathbf{u}_0 + \mathbf{Z}_m\mathbf{y}_m$
  If satisfied Stop, else set $\mathbf{u}_0 \leftarrow \mathbf{u}_m$ and GoTo 2.
end procedure

In Algorithm 3, the preconditioner employed at iteration k is denoted by $\mathbf{M}_k$. Since its inverse is applied to the k-th element of the Krylov basis $\mathbf{v}_k$, we infer that $\mathbf{M}_k$ is generally used to find an approximation of $\mathbf{c}_k$, which is defined as the solution of the following problem:

$$\mathbf{A}\mathbf{c}_k = \mathbf{v}_k. \tag{38}$$

Indeed, if by any chance $\mathbf{M}_k^{-1}\mathbf{v}_k = \mathbf{A}^{-1}\mathbf{v}_k$, we have that FGMRES converges to the exact solution. In the MSRB case, we have $\mathbf{M}_k = \mathbf{M}_k(\boldsymbol{\mu}) = \mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$, meaning that the action of its inverse on $\mathbf{v}_k$ can be computed as

$$\mathbf{M}_k^{-1}(\boldsymbol{\mu})\mathbf{v}_k = \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_k + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{v}_k, \qquad k = 1, 2, \dots. \tag{39}$$

To find the right problem for training the k-th RB space, we note that in equation (39) the reduced component of $\mathbf{P}_{\mathrm{MSRB},k}$ is applied to the vector $\left(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\right)\mathbf{v}_k$. In order to suitably precondition the FGMRES method, the k-th RB space must therefore be trained to solve the following problem

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{y}^{(k)}(\boldsymbol{\mu}) = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{v}_k, \qquad k = 1, 2, \dots, \tag{40}$$

yielding a RB space of the form

$$V_{N_k} = span\Big\{\mathbf{y}^{(k)}(\boldsymbol{\mu}_i)\Big\}_{i=1}^{N_k}, \qquad k = 1, 2, \dots, \tag{41}$$

where $\mathbf{y}^{(k)}(\boldsymbol{\mu}_i)$ is the solution of equation (40) with $\boldsymbol{\mu} = \boldsymbol{\mu}_i$.

Following a similar argument to the one used for the Richardson method in Section 2.1, and exploiting the expressions of the Krylov basis given in Algorithm 3, we can find an explicit formula for the basis of

the RB space k. The most suitable initial guess is the solution of the reduced basis system, we therefore set $\mathbf{u}^{(0)}(\boldsymbol{\mu}) = \mathbf{V}_0\mathbf{A}_{N_0}^{-1}(\boldsymbol{\mu})\mathbf{V}_0^T\mathbf{f}_h(\boldsymbol{\mu})$, which yields

$$\mathbf{r}_0(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(0)}(\boldsymbol{\mu}), \qquad \beta(\boldsymbol{\mu}) = \|\mathbf{r}_0(\boldsymbol{\mu})\|_2, \qquad \mathbf{v}_1 = \mathbf{r}_0(\boldsymbol{\mu})/\beta(\boldsymbol{\mu}). \tag{42}$$

Following (40), the first preconditioner $\mathbf{M}_1^{-1}(\boldsymbol{\mu})$ must effectively precondition the problem

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{y}^{(1)}(\boldsymbol{\mu}) = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{v}_1 = \frac{1}{\beta(\boldsymbol{\mu})}\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{r}_0(\boldsymbol{\mu}),$$

whose true high-fidelity solution $\mathbf{y}_1(\boldsymbol{\mu})$ has the following form:

$$\begin{aligned}
\mathbf{y}^{(1)}(\boldsymbol{\mu}) &= \mathbf{A}_h^{-1}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{v}_1 = \frac{1}{\beta(\boldsymbol{\mu})}\mathbf{A}_h^{-1}(\boldsymbol{\mu})\mathbf{r}_0(\boldsymbol{\mu}) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_1 \\
&= \frac{1}{\beta(\boldsymbol{\mu})}\mathbf{A}_h^{-1}(\boldsymbol{\mu})\Big(\mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(0)}(\boldsymbol{\mu})\Big) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_1 \\
&= \frac{1}{\beta(\boldsymbol{\mu})}\mathbf{A}_h^{-1}(\boldsymbol{\mu})\Big(\mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(0)}(\boldsymbol{\mu})\Big) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_1 \\
&= \frac{1}{\beta(\boldsymbol{\mu})}\Big(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}^{(0)}(\boldsymbol{\mu})\Big) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_1.
\end{aligned}$$

We now proceed by induction, supposing to have built our preconditioner up to step k, and show how to build the $(k+1)$-th step. Following (40), $\mathbf{y}^{(k+1)}(\boldsymbol{\mu})$ must have the form

$$\mathbf{y}^{(k+1)}(\boldsymbol{\mu}) = \mathbf{A}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_{k+1} - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_{k+1}, \tag{43}$$

where $\mathbf{v}_{k+1}$ is the $(k+1)$-th Krylov basis, that we can express through Algorithm 3 as

$$\mathbf{v}_{k+1} = \frac{1}{h_{k+1,k}}\Big(\mathbf{A}_h(\boldsymbol{\mu})\mathbf{M}_k^{-1}\mathbf{v}_k - \sum_{j=1}^{k} h_{j,k}\mathbf{v}_j\Big) \qquad k = 1, 2, \dots,$$

thus yielding

$$\mathbf{y}^{(k+1)}(\boldsymbol{\mu}) = \frac{1}{h_{k+1,k}}\Big(\mathbf{M}_k^{-1}(\boldsymbol{\mu})\mathbf{v}_k - \sum_{j=1}^{k} h_{j,k}\mathbf{A}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_j\Big) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_{k+1} \qquad k = 1, 2, \dots.$$

Finally, recalling that $\mathbf{z}_k = \mathbf{M}_k^{-1}\mathbf{v}_k$, and expressing $\mathbf{A}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_k = \mathbf{y}^{(k)}(\boldsymbol{\mu}) + \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_k$ from Equation (43) computed at step k, we generate the recursive formula

$$\begin{cases}
\beta(\boldsymbol{\mu}) = \|\mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(0)}(\boldsymbol{\mu})\|_2, \\
\mathbf{y}^{(1)}(\boldsymbol{\mu}) = \frac{1}{\beta(\boldsymbol{\mu})}\big(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{v}_1\big) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_1, \\
\mathbf{y}^{(k+1)}(\boldsymbol{\mu}) = \frac{1}{h_{k+1,k}}\Big[\mathbf{z}_k(\boldsymbol{\mu}) - \sum_{j=1}^{k} h_{j,k}\big(\mathbf{y}^{(j)}(\boldsymbol{\mu}) + \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_j\big)\Big] - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_{k+1}, \quad k \geq 1.
\end{cases} \tag{44}$$

The practical construction of the spaces (41) is handled similarly to the case of the Richardson method: we first compute the high-fidelity solutions for a set of parameters $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$, then we iteratively build the snapshots of the errors $\{\mathbf{y}^{(k)}(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ following relations (44) and then perform POD on this set of snapshots. Again, we highlight that the construction of the snapshots only involves the solution of the high-fidelity problem for the step $k = 0$. Compared to the Richardson case, the snapshots of the k-th step depend on the snapshots obtained at all the previous steps, hence requiring a (slightly) higher data storage during the offline stage.

3 Numerical experiments

In this section we present the results of numerical experiments related to several test cases governed by advection-diffusion (AD) equations to investigate the performance of the preconditioner developed so far. We first focus on a pure diffusion problem showing piecewise constant, parameter dependent diffusivities modeling a heat conduction problem across different materials. In the second test case we turn our attention to a parametrized advection diffusion equation describing the dynamics of a solute in a blood flow. We show

results for the FGMRES method, for which we take into account both the fixed accuracy and fixed dimension approaches for constructing the RB spaces. On the other hand, results with the Richardson method are not reported for the sake of information synthesis, since similar conclusions can be drawn. In all the tests POD is performed with respect to the scalar product induced by the symmetric positive definite matrix $\mathbf{Y}_h$, that represents the $H_0^1(\Omega)$ scalar product on V_h, see Appendix A. In each case, we use a stopping criterion for FGMRES based on the Euclidean norm of the FE vector of the residual, rescaled with respect to the Euclidean norm of the right hand side, with a tolerance that has been set to $\varepsilon_r = 10^{-7}$ for all test cases. On the other hand, we compute the RB spaces to fulfill (36) with $\delta = 10^{-9}$. This is necessary because the optimality of POD is recovered only on the sum of the snapshots, see Appendix A. Moreover, since each RB space is suited for a particular iteration up to iteration $L-1$, when the number of iterations required to reach the prescribed tolerance ε_r exceeds the number of RB spaces, the final iterations employ the last RB space as coarse correction, i.e. $\mathbf{P}_{\text{MSRB},k}(\boldsymbol{\mu}) = \mathbf{P}_{\text{MSRB},L-1}\ \forall k \geq L$.

As fine preconditioner, we employ $\mathbf{P}_h(\boldsymbol{\mu}) = \mathbf{P}_{\text{BJ}}(\boldsymbol{\mu})$, a Block Jacobi preconditioner, where each block represents the restrictions to the degrees of freedom of a subdomain. The subdomains are selected by Parmetis[1] at the mesh level. This allows to reduce communication costs for the preconditioner both in the construction and the application phases.

For all the simulations we report the number of RB spaces L and RB functions N_k, $k = 0, 1, \ldots$ produced by either Algorithm 1 or 2, the results obtained online with the MSRB preconditioner averaging on $N_{onl} = 250$ parameters chosen randomly and different from the ones used to construct the RB spaces. Finally, the number of snapshots n_s and the computational time t_{off} required by the offline phase are reported for each simulation.

All our experiments have been carried out using LifeV[2] [9] on the cluster Piz-Daint provided by the Swiss National Supercomputing Center (CSCS) on a Cray XC40 machine.

3.1 Test case 1: diffusion in a blockwise cubic domain

We consider a parametrized diffusion problem in a blockwise cubic domain, including anisotropy effects on the diffusion tensor. This class of problems represents a challenge for the standard RB method when the problem features a nonaffine dependence on the parameter $\boldsymbol{\mu}$; as a matter of fact, in this case it is necessary to recover an approximated affine dependence, which may however hamper the efficiency and/or the accuracy of the RB method.

3.1.1 Problem setting

Given $\Omega = (0,1)^3 \subset \mathbb{R}^3$, such that $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\mathring{\Gamma}_D \cap \mathring{\Gamma}_N = \emptyset$, we subdivide it into $\mathcal{J}$ subregions Ω_j, $j = 1, \ldots \mathcal{J}$ s.t. $\bar{\Omega} = \cup_{j=1}^{\mathcal{J}} \bar{\Omega}_j$ and $\mathring{\Omega}_i \cap \mathring{\Omega}_j$, $i \neq j$. More precisely we set $\mathcal{J} = 4$ and subdivide Ω such that $\mathring{\Omega}_1 = (0,1) \times (0,0.5) \times (0,0.5)$, $\mathring{\Omega}_2 = (0,1) \times (0,0.5) \times (0.5,1)$, $\mathring{\Omega}_3 = (0,1) \times (0.5,1) \times (0,0.5)$, $\mathring{\Omega}_4 = (0,1) \times (0.5,1) \times (0.5,1)$. Let us consider the following parametrized PDE:

$$\begin{cases} -\nabla \cdot (\mathcal{K}(\boldsymbol{\mu}) \nabla u(\boldsymbol{\mu})) = f(\boldsymbol{\mu}) & \text{in } \Omega \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_D, \\ \frac{\partial u(\boldsymbol{\mu})}{\partial n} = 0 & \text{on } \Gamma_N, \end{cases} \tag{45}$$

where the diffusion tensor is $\mathcal{K}(\boldsymbol{\mu}) = \mathcal{K}(\mathbf{x}; \boldsymbol{\mu}) = \nu(\mathbf{x}) diag(1, 1, 10^{-2})$, and $\nu(\mathbf{x}) > 0$ is the piecewise constant material property on each Ω_j:

$$\nu(\mathbf{x}) = \begin{cases} \nu_j & \text{if } \mathbf{x} \in \Omega_j, \quad j = 1, \ldots, \mathcal{J}-1 \\ 1 & \text{if } \mathbf{x} \in \Omega_{\mathcal{J}}. \end{cases}.$$

We consider as source term the following parameter dependent function

$$f(\boldsymbol{\mu}) = f(\mathbf{x}; \boldsymbol{\mu}) = \sigma + \frac{1}{\sigma} \exp\Big(\frac{\|\mathbf{x} - \mathbf{y}_0\|^2}{\sigma}\Big), \tag{46}$$

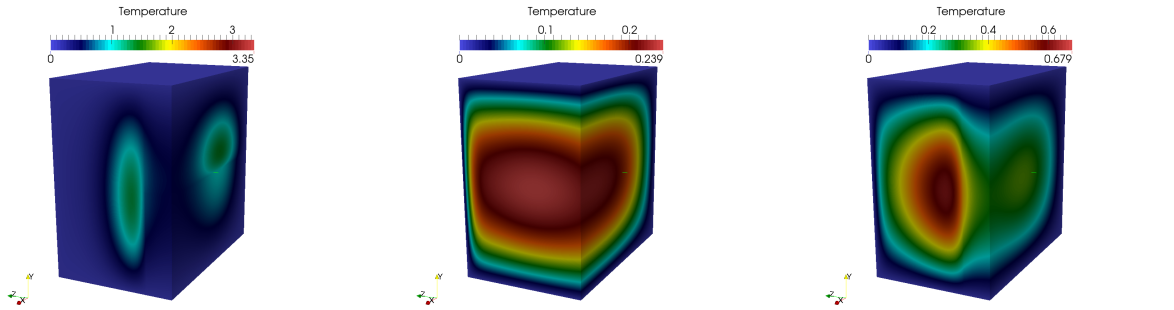
a Gaussian rescaled function centered in $\mathbf{y}_0 \in \Omega$, shifted and squeezed of a factor $\sigma > 0$.
Problem (45) is parametrized with respect to the diffusion coefficients ν_j, $j = 1, \ldots, \mathcal{J}-1$, the coordinates

[1] http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview
[2] www.lifev.org

$\mathbf{y}_0$ and the scaling factor σ, leading to the 7-dimensional parameter vector:

$$\boldsymbol{\mu} = (\nu_1, \ldots, \nu_{\mathcal{J}-1}, \mathbf{y}_0, \sigma) \in \mathcal{D} = [0.1, 1]^{\mathcal{J}-1} \times [0.4, 0.6]^3 \times [\sigma_{min}, 0.5] \subset \mathbb{R}^7, \tag{47}$$

where $\sigma_{min} > 0$. The localized (in space) parametrized nature of $f(\boldsymbol{\mu})$, together with the varying diffusion coefficients yield a problem which is challenging from the parameter viewpoint, as it is hardly solvable accurately by the standard RB method.
For the sake of simplicity, we consider homogeneous Dirichlet and Neumann boundary conditions, although the whole framework can be easily adapted to the case of nonhomogeneous (parametrized) boundary conditions in a straightforward way. Moreover, in all simulations, we employ linear piecewise continuous FE tetrahedrals on structured meshes as high-fidelity discretization. Examples of solutions obtained for different values of parameters, are reported in Fig. 1.



(a) $\boldsymbol{\mu} = (0.1, 0.1, 0.1, 0.4, 0.4, 0.4, 0.05)$ (b) $\boldsymbol{\mu} = (1, 1, 1, 0.6, 0.6, 0.6, 0.5)$ (c) $\boldsymbol{\mu} = (1, 0.5, 1, 0.5, 0.5, 0.5, 0.2)$

Figure 1: Example of solutions with a null Neumann condition on $x = 1$.

3.1.2 Analysis with respect to the mesh size

We carry out a first analysis with respect to three different grids whose characteristic dimensions are $h = 0.05, 0.025, 0.0125$, leading to dimensions $N_h = 365'254$, $2'887'193$ and $22'767'295$, respectively, for the high-fidelity FE approximation. We choose $\sigma \in [0.25, 0.5]$ and construct the RB spaces by POD with $n_s = 750$ snapshots. These simulations have been carried out with 96, 768, 6144 processors, respectively, in order to maintain a constant number of degrees of freedom (about 3800) per processor. The meshes have been partitioned in subdomains (independently from the subregions identifying the material properties) by using Parmetis[3]. We compare the results with those obtained using an algebraic multigrid preconditioner $\mathbf{P}_{\text{ML}}(\boldsymbol{\mu})$ from the Trilinos ML package [28], which exploits an exact coarse component and 2-sweeps Gauss-Seidel smoother and with the GCRO-DR Krylov subspace recycling method proposed in [38], where $\mathbf{P}_{\text{ML}}(\boldsymbol{\mu})$ is again used as preconditioner, for sequences of linear systems with varying matrices and right hand sides. The latter method combines the optimal truncation strategy of GCRO [20] with deflation employed in GMRES with deflated restarting, GMRES-DR [34]. Both techniques are obtained with default settings from the Trilinos library.
The results are reported in Tab. 1, 2. The computational time employed online to solve the linear system (1) using $\mathbf{P}_{\text{MSRB},k}$ as preconditioner for the new instances of the parameter is not highly impacted by the FE dimension, since the number of RB coarse components and their dimensions are not significantly affected by changing the FE dimension. Indeed, the online computational time t_{MSRB} and the number of iterations are always lower than the ones obtained either by $\mathbf{P}_{\text{ML}}(\boldsymbol{\mu})$ (t_{GML}) or GCRO-DR ($t_{\text{GCRO-DR}}$), for both the fixed dimension and the fixed accuracy approaches. Moreover, we notice that the MSRB preconditioner built with the fixed accuracy approach features a faster online solution and a less expensive offline phase than the one built with the fixed dimension approach. The larger the FE dimension, the more expensive the offline phase, regarding in particular the computational time for snapshots computation t_{n_s}, while the time devoted to POD, t_{POD}, is less affected. Also in terms of memory requirements, the fixed accuracy approach (entailing the storage of about 1050 FE vectors to build the RB spaces for the problem at hand) is less demanding than the fixed dimension approach (1400 FE vectors). These requirements make data storage related to our preconditioners heavier than the one required by the ML preconditioner, although this latter is used only for a single-instance of the parameter, if no updating or recycling techniques are

[3] http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview

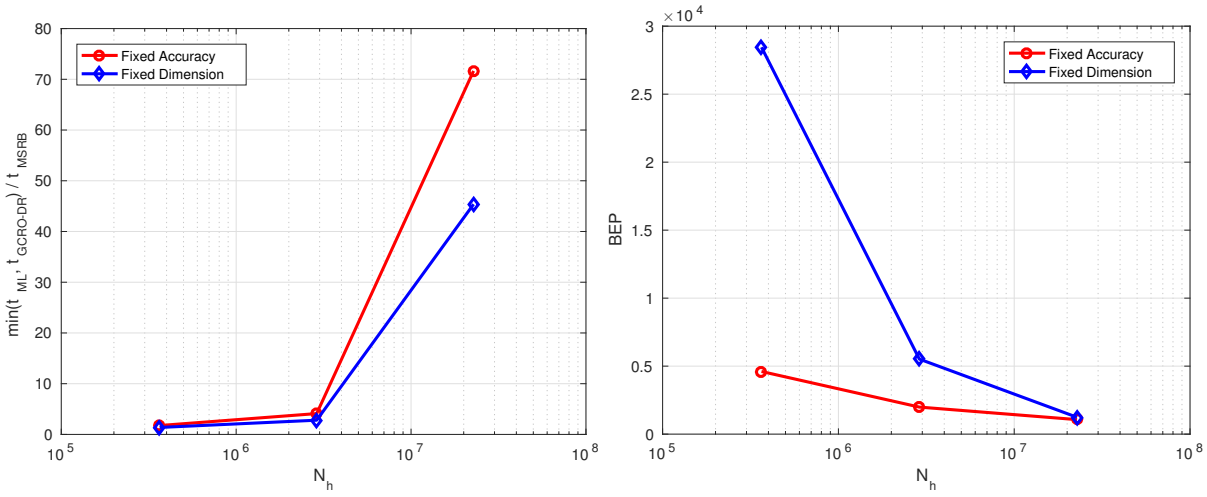
employed. Nevertheless, compared to the standard RB method, the number of FE vectors stored by our preconditioners is of the same order.
In Tab. 1, 2 we report the break-even point (BEP), i.e. the number of online evaluation needed to repay the offline phase, which is a decreasing function of the FE dimension. In Fig. 2a and 2b the speedup obtained with respect to using the most convenient between $\mathbf{P}_{\mathrm{ML}}(\boldsymbol{\mu})$ and GCRO-DR technique and the break-even point (BEP), i.e. the number of online evaluation needed to repay the offline phase, are reported as function of the FE dimension. The greater the FE dimension, the higher the speedup and the lower the break-even point. In the case with $N_h = 22'767'295$ both $\mathbf{P}_{\mathrm{ML}}(\boldsymbol{\mu})$ and GCRO-DR perform very poorly due to the very large FE dimension and the corresponding huge communication costs; in particular the latter succeeds in recycling the Krylov subspace in reducing the time of about 10%. On the other hand, the MSRB preconditioner relies on embarrassingly parallel fine and coarse components, and the linear system (1) is solved by the MSRB preconditioned FGMRES up to 70 (resp. 50) faster than either $\mathbf{P}_{\mathrm{ML}}(\boldsymbol{\mu})$ or GCRO-DR and 1067 (resp. 1240) online evaluations are required to reach the break-even point for the fixed accuracy (resp. fixed dimension) approach. This is the case for applications involving, e.g., sensitivity analysis or uncertainty quantification.

Table 1: Grid analysis FGMRES fixed accuracy, $L = 3$, $\delta_{RB,k} = 0.001$, $\forall k$, ~ 3800 dofs per CPU.

N_{cpu}	N_k	$t_{\mathrm{MSRB}}(It)$	$t_{\mathrm{GML}}(It)$	$t_{\mathrm{GCRO\text{-}DR}}(It)$	t_{off}	t_{n_s}	t_{POD}	BEP
96	49 296 725	0.34 (5)	0.59 (28)	0.48 (28)	1161.21	1071.66	89.55	4606
768	48 279 721	0.46 (9)	1.91 (41)	2.29 (38)	2872.27	2746.07	126.20	1989
6144	49 269 713	0.75 (12)	55.73 (54)	49.98 (53)	56768.20	56486.86	281.34	1067

Table 2: Grid analysis FGMRES fixed dimension, $N_k = 100$ $\forall k$, ~ 3800 dofs per CPU.

N_{cpu}	L	$t_{\mathrm{MSRB}}(It)$	$t_{\mathrm{GML}}(It)$	$t_{\mathrm{GCRO\text{-}DR}}(It)$	t_{off}	t_{n_s}	t_{POD}	BEP
96	15	0.43 (13)	0.59 (28)	0.48 (28)	4546.21	4390.47	155.74	28448
768	14	0.68 (25)	1.91 (41)	2.29 (38)	6775.94	6597.13	178.81	5517
6144	13	1.19 (40)	55.73 (54)	49.98 (53)	65437.90	64951.30	486.60	1240



(a) Speedup.

(b) Break-even point.

Figure 2: Test case 1: Speed up and break-even point (BEP) as function of N_h.

3.1.3 Comparison with the standard RB method

A natural question arising in this context is about the comparison, in terms of both accuracy and efficiency, between the proposed approach (MSRB preconditioning) and the classical RB method. In this latter case, the solution of system (1) is approximated by the one of the RB problem (4). In this section we compare the results obtained with the standard RB method and the FGMRES Krylov method preconditioned with the proposed MSRB preconditioner, showing results for the FE grid with $N_h = 2'887'193$.
At first, we notice that the function (46) nonaffinely depends on the parameter $\boldsymbol{\mu}$, leading to a nonaffine right hand side $\mathbf{f}_h(\boldsymbol{\mu})$ in (1). The nonaffine dependence of the operators is one of the most limiting bottleneck of

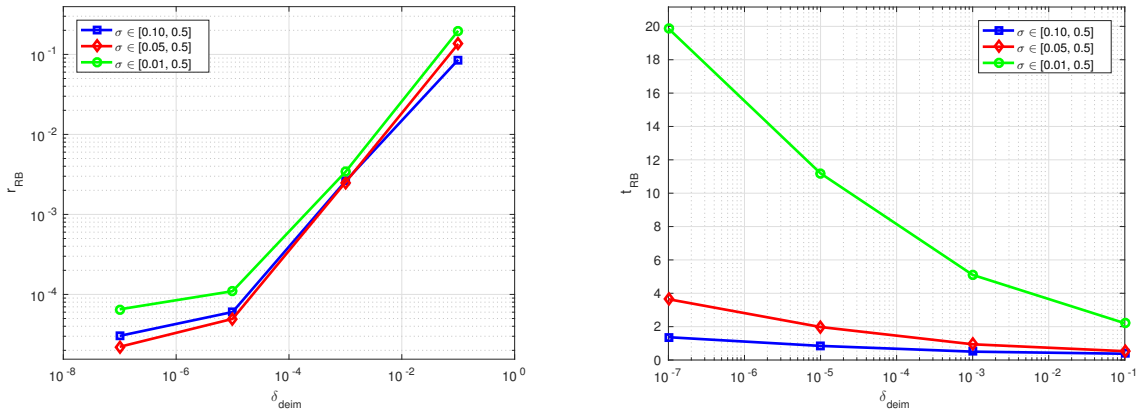
the standard RB method, as it does not allow to uncouple the RB arrays from the FE dimension and gain the maximum speed up with respect to the high-fidelity simulation. Consequently, the Empirical Interpolation Method (EIM) [3], or its discrete variants DEIM and MDEIM [15, 35], should be used to construct an approximated affine decomposition. In our case, we employ the DEIM algorithm [15], see Appendix A, to deal with the nonaffine right hand side. This is approximated as linear combination of properly chosen DEIM basis functions up to a certain tolerance δ_{deim}, which is plugged in the DEIM algorithm. It is well known that on one hand the tolerance δ_{deim} limits the accuracy of the RB approximation and, on the other hand, it may yield a huge overhead in the online phase due to a (possibly) large number of DEIM basis functions. This is indeed the case of (46) due to the localized (in space) nature of the source term.
We employ the POD-DEIM-RB method with different DEIM tolerances $\delta_{\text{deim}} = 10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}$ and values of $\sigma_{min} = 0.1, 0.05, 0.01$, while we build the RB space through POD algorithm by setting a tolerance of $\epsilon_{POD} = 10^{-9}$ for all the tests. We choose a number of snapshots equal to $n_s = 1000$ for $\sigma_{min} = 0.1$, $n_s = 2000$ for $\sigma_{min} = 0.05$ and $n_s = 3500$ for $\sigma_{min} = 0.01$. In Fig. 3a, we report the average relative residual, which is defined as

$$r_{\text{RB}} = \frac{\|\mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})\|_2}{\|\mathbf{f}_h(\boldsymbol{\mu})\|_2}, \tag{48}$$

evaluated over $N_{onl} = 250$ online parameters. The results show that the accuracy of the RB method is strongly hampered by the tolerance δ_{deim} provided to the DEIM algorithm. In order to obtain a satisfactory accuracy it is compulsory to use a small δ_{deim}. Moreover, we observe from Fig. 3a that from a certain point the residual stagnates to a the value 10^{-5} even if a smaller δ_{deim} has been provided. In Fig. 3b the wall time employed to assemble and solve the RB problem is reported for the different values of σ_{min} employed. The total time is largely affected by the value of σ_{min} and by the number of DEIM basis functions: the smaller the tolerance of the DEIM algorithm, the bigger the wall time required to compute the RB solution (even up to 18.66 s for $\sigma_{min} = 0.01$ and $\delta_{\text{deim}} = 10^{-7}$). In Fig. 4a, 4b and 4c we measure the computational wall time distinguishing the different phases of building and solving the corresponding RB problem online for $\sigma_{min} = 0.1, 0.05, 0.01$, respectively. The large wall time required by small DEIM tolerances is caused by assembling the RB right hand side. The huge number of DEIM functions and the communication needed to compute the coefficients $\tilde{\Theta}_f^q$, $q = 1, \ldots, Q_f$ in (58) yield an extremely expensive assembling phase which largely affects the computation. The detail of the computations are reported in Tab. 3, 4 and 5 for $\sigma_{min} = 0.1, 0.05, 0.01$, respectively. $t_{f_N(\boldsymbol{\mu})}$, $t_{A_N(\boldsymbol{\mu})}$ and t_{solve} correspond to the average time needed to assemble the RB right hand side, the RB matrix and to solve the RB linear system, respectively, while t_{onl} is the sum of these three stages. t_{DEIM} corresponds to the time needed to run the DEIM algorithm offline, i.e. build the DEIM basis composed of M_f functions, t_{RB} the time to compute the snapshot and run the POD to build N RB basis functions and t_{off} is the sum of the two.
On the contrary, in the FGMRES preconditioned with the MSRB preconditioner, an approximated affine decomposition of the right hand side is not needed, since we solve the full FE problem. In Tab. 6 the results obtained setting a final relative tolerance for the FGMRES equal to 10^{-7} are reported for the fixed dimension approach. For $\sigma_{min} = 0.1, 0.05, 0.01$, we have set $N_k = 180, 300, 600$, respectively. In all cases the algorithm has built $L = 13$ RB spaces. The total time needed to solve the FE system (reaching an accuracy on the relative residual of 2 orders of magnitude lower than in the standard RB case) ranges from 1.0 to 1.6 seconds. The case $\sigma \in [0.01, 0.5]$ is more costly than the other two since the RB coarse corrections are more expensive due to the bigger dimension of the RB spaces ($N_k = 600$). Hence, the total offline time t_{off} is larger than the one for the standard RB method, due to the larger number of PODs and the necessity to build the snapshots errors with (44). This is, however, highly repaid during the online phase, when the FGMRES with the MSRB preconditioner reaches a much more accurate (100 times) result with a relevant speedup, up to almost 12 times faster of the standard RB method for the case with $\sigma \in [0.01, 0.5]$. The MSRB-preconditioned FGMRES is therefore a promising technique to deal with challenging nonaffine problems, since it allows to exploit the parameter dependence overcoming the need to have an accurate affine decomposition of the right hand side.

Table 3: RB results $\sigma \in [0.1, 0.5]$, $n_s = 1000$.

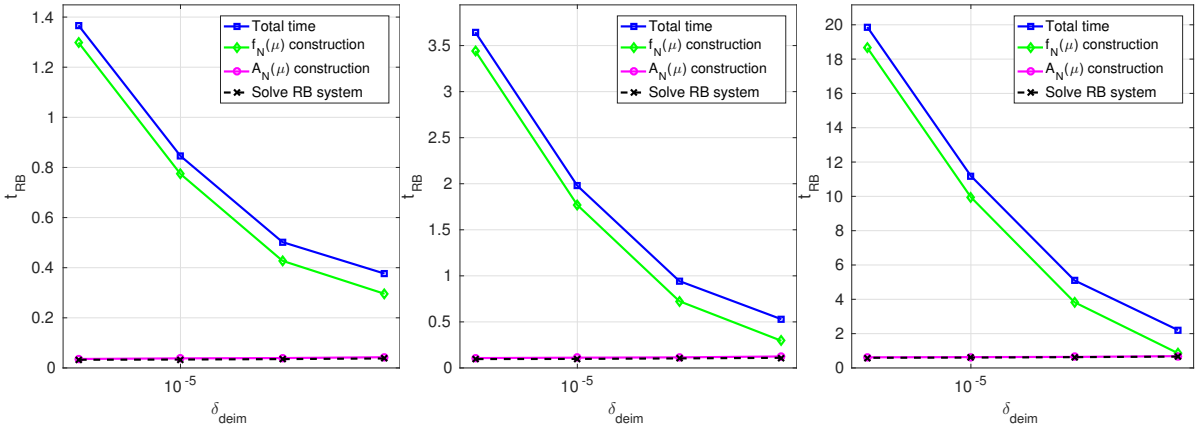
δ_{deim}	r_{RB}	N	M_f	$t_{f_N(\boldsymbol{\mu})}$	$t_{A_N(\boldsymbol{\mu})}$	t_{solve}	t_{onl}	t_{RB}	t_{DEIM}	t_{off}
1.00e-01	8.58e-02	670	5	0.30	0.04	0.04	0.38	7506.60	383.92	7890.52
1.00e-03	2.59e-03	670	33	0.43	0.04	0.04	0.50	7506.60	284.43	7791.03
1.00e-05	6.03e-05	670	94	0.77	0.04	0.03	0.85	7506.60	306.47	7813.07
1.00e-07	3.03e-05	670	196	1.30	0.04	0.03	1.37	7506.60	407.15	7913.75



(a) Relative residual.

(b) Wall times for solving and assembling the RB problem.

Figure 3: Test case 1: relative residual and average wall time as function of δ_{deim}.



(a) $\sigma \in [0.1, 0.5]$.

(b) $\sigma \in [0.05, 0.5]$.

(c) $\sigma \in [0.01, 0.5]$.

Figure 4: Test case 1: average wall time for online parameters as function of δ_{deim}.

Table 4: RB results $\sigma \in [0.05, 0.5]$, $n_s = 2000$.

δ_{deim}	r_{RB}	N	M_f	$t_{f_N(\boldsymbol{\mu})}$	$t_{A_N(\boldsymbol{\mu})}$	t_{solve}	t_{onl}	t_{RB}	t_{DEIM}	t_{off}
1.00e-01	1.35e-01	1055	8	0.29	0.12	0.11	0.53	16289.37	811.15	17100.52
1.00e-03	2.51e-03	1055	59	0.72	0.11	0.10	0.94	16289.37	765.92	17055.29
1.00e-05	4.95e-05	1055	172	1.77	0.11	0.10	1.98	16289.37	620.45	16909.82
1.00e-07	2.22e-05	1055	341	3.44	0.11	0.10	3.65	16289.37	1272.94	17562.31

Table 5: RB results $\sigma \in [0.01, 0.5]$, $n_s = 3500$.

δ_{deim}	r_{RB}	N	M_f	$t_{f_N(\boldsymbol{\mu})}$	$t_{A_N(\boldsymbol{\mu})}$	t_{solve}	t_{onl}	t_{RB}	t_{DEIM}	t_{off}
1.00e-01	1.98e-01	2143	24	0.86	0.69	0.66	2.21	21983.40	2110.75	24094.15
1.00e-03	3.39e-03	2143	203	3.83	0.64	0.63	5.10	21983.40	1494.40	23477.80
1.00e-05	1.10e-04	2143	562	9.95	0.63	0.60	11.19	21983.40	2369.54	24352.94
1.00e-07	6.52e-05	2143	1060	18.66	0.62	0.59	19.87	21983.40	8820.92	30804.32

Table 6: MSRB results Fixed dimension.

	L	N_k	$t_{\text{MSRB}}(It)$	n_s	t_{off}	t_{n_s}	t_{POD}
$\sigma \in [0.1, 0.5]$	13	180	1.0458 (22)	1000	13820	12791.51	1028.49
$\sigma \in [0.05, 0.5]$	13	300	1.0049 (15)	2000	26406.4	23213.01	3193.39
$\sigma \in [0.01, 0.5]$	13	600	1.5889 (17)	3500	50973.7	40561	10412.7

3.2 Test case 2: solute dynamics in blood flow and arterial walls

We investigate here the dynamics of a solute by focusing on the solution of a fluid-wall mass-transport model which describes the exchange of substances between blood in the lumen and arterial wall. In this context, the solute is regarded as a passive scalar transported along the artery by the blood, which is modeled as a Newtonian fluid and governs the exchange of the solute through the stress produced on the arterial wall. We take into account the so-called steady wall-free model for the absorption of the solute, [44], which couples the steady Navier-Stokes equations, which describe the velocity and pressure fields, with an advection diffusion equation governing the concentration of the solute. This model is parametrized with respect to the permeability of the arterial wall and the diffusion coefficient of the solute in the blood, whereas the concentration of the solute in the wall is considered to be constant. This problem has been largely addressed and studied in the literature, see e.g. [12, 44] and the references therein for an extensive description.

3.2.1 The physical model and its FE discretization

We consider an open bounded domain $\Omega_f \in \mathbb{R}^3$, such that $\partial\Omega_f = \Gamma_w \cup \Gamma_{out} \cup \Gamma_{in}$. Here, Γ_w, Γ_{out} and Γ_{in} denote the artery wall, the outlet and the inlet, respectively, see Fig. 5a. The physical domain Ω_f describes the carotid bifurcation with an average section radius $r = 0.3\,cm$. We define $C_f \in [0,1]$ as the normalized concentration of the solute, whose dynamics is governed by the following advection diffusion equation:

$$\begin{cases} -\nabla \cdot (\nu_f \nabla C_f) + \tilde{\mathbf{u}} \cdot \nabla C_f = 0, & \mathbf{x} \in \Omega_f \\ \mathbf{n} \cdot (\nu_f \nabla C_f) + \xi C_f = \xi k_w & \text{on } \Gamma_w \\ C_f = 1 & \text{on } \Gamma_{in} \\ \mathbf{n} \cdot (\nu_f \nabla C_f) = 0 & \text{on } \Gamma_{out}, \end{cases} \tag{49}$$

where ν_f is the diffusivity coefficient of the solute, ξ and k_w are the permeability and the concentration in the arterial wall, respectively. We model the permeability of the wall as $\xi = \xi(\tilde{\mathbf{u}}) = \beta(1 + \tau_w(\tilde{\mathbf{u}}))$, being $\tau_w(\tilde{\mathbf{u}})$ the wall shear stress (WSS) distribution on Γ_w, and we choose as vector of parameters $\boldsymbol{\mu} = (\nu_f, \beta) \in [5 \cdot 10^{-5}, 5 \cdot 10^{-2}] \times [10^{-4}, 10^{-3}]$. On the other hand, we fix the value of $k_w = 0.5$ for all the simulations. The advection field $\tilde{\mathbf{u}} = \tilde{\mathbf{u}}(\mathbf{x})$ describes the velocity of the blood flow, and it is governed by the steady Navier-Stokes (NS) equations corresponding to the systolic peak. As boundary conditions for the NS equations we set a no-slip condition on Γ_w, homogeneous Neumann conditions on Γ_{out} and a parabolic inlet velocity, with a peak 22.5 cm s^{-1}, on Γ_{in}. Finally we consider a constant kinematic viscosity of the blood $\nu = 0.035$ $cm^2 s^{-1}$. We remark that in our model the NS equations are not parametrized, their solution only representing a datum for problem (49).

Here we consider the solution of problem (49) for very small values of ν_f which yield huge Péclet numbers $Pe = \frac{|\tilde{\mathbf{u}}|r}{2\nu_f}$, and since the standard FE method may lead to oscillations for such convective dominant problems, we employ a stabilized FE formulation. Hence, contrarily to the Test case 1 described in Section 3.1, where the weak formulation yielding the high-fidelity approximation (1) is straightforward, we report the weak formulation of problem (49), which reads: find $C_f \in V = V(\Omega_f) = \left\{v \in H^1(\Omega_f) : v|_{\Gamma_{in}} = 1\right\}$ such that

$$\int_{\Omega_f} (\nu_f \nabla C_f \nabla w + \tilde{\mathbf{u}} \cdot \nabla C_f w) + \int_{\Gamma_w} \xi C_f w = \int_{\Gamma_w} \xi k_w w, \qquad \forall w \in H^1_{\Gamma_{in}}(\Omega_f), \tag{50}$$

where $H^1_{\Gamma_{in}}(\Omega_f) = \left\{v \in H^1(\Omega_f) : v|_{\Gamma_{in}} = 0\right\}$. As high-fidelity discretization, we employ a streamline-upwind/Petrov-Galerkin (SUPG) stabilized FE formulation. To this aim, we introduce a conforming partition $\mathcal{T}_h$ of Ω_f and the FE space

$$X_h^r = \left\{w_h \in C^0(\bar{\Omega}_f) : w_h|_K \in \mathcal{P}_r(K)\, \forall K \in \mathcal{T}_h\right\}, \tag{51}$$

where $\mathcal{P}_r(K)$ denotes the space of polynomials with degree lower than or equal to r on the element K. Then, the SUPG-FE formulation reads: find $C_{f,h} \in V_h = X_h^r \cap V$ such that

$$\begin{aligned} \int_{\Omega_f} (\nu_f \nabla C_{f,h} \nabla w_h + \tilde{\mathbf{u}} \cdot \nabla C_{f,h} w_h) + \int_{\Gamma_w} \xi C_{f,h} w_h + \sum_{K \in \mathcal{T}_h} \left(\tilde{\mathbf{u}} \cdot \nabla C_{f,h} - \nabla \cdot (\nu_f \nabla C_f) \tau_K \tilde{\mathbf{u}} \cdot \nabla w_h\right)_K \\ = \int_{\Gamma_w} \xi k_w w_h, \qquad \forall w_h \in W_h = X_h^r \cap H^1_{\Gamma_{in}}(\Omega_f); \end{aligned} \tag{52}$$

here $(\cdot,\cdot)_K$ denotes the $L^2(K)$ scalar product on $K \in \mathcal{T}_h$, whereas

$$\tau_K = \delta_S \frac{h_K}{|\tilde{\mathbf{u}}|}, \tag{53}$$

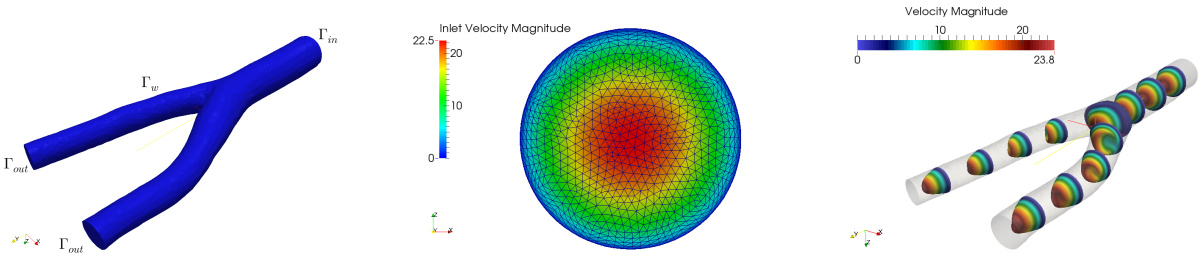
being δ_S a positive constant and h_K the diameter of the element $K \in \mathcal{T}_h$.

A quantity of interest we are interested to evaluate for different values of the parameters is the Sherwood number, which measures the non-dimensional mass flux through the vessel wall, see e.g. [18], and is defined as

$$Sh = \frac{-2r(\nabla C_f \cdot \mathbf{n})}{C_{f,in} - k_w},$$

where $r = 0.3\,cm$ is the reference radius of the artery and $C_{f,in} = 1$ is the inlet concentration.

Concerning the numerical setting, we employ a mesh with boundary layer, and a $\mathcal{P}_2-\mathcal{P}_1$ FE discretization for the NS equations, whose resulting velocity field is reported in Figure 5c. Concerning the discretization of equation (52), we analyze the performance of $\mathbf{P}_{\text{MSRB},k}(\boldsymbol{\mu})$ with respect to the employment of $\mathcal{P}_1$ and $\mathcal{P}_2$ FE, resulting in 429'892 and 3'467'673 degrees of freedom, respectively. We are particularly interested in the case of quadratic ($\mathcal{P}_2$) FE because the evaluation of quantities involving the gradient of the concentration, as the Sherwood number, need a very accurate computation of the derivatives of the unknown. In Fig.6 we report the Sherwood number obtained for different instances of the parameter: we notice that employing quadratic FE polynomials can yield significantly more precise values. As in the previous test case, we use a stopping criterion based on the euclidean norm of the FE residual rescaled with respect to the right hand side for the iterative method, with a tolerance equal to $\varepsilon_r = 10^{-7}$.

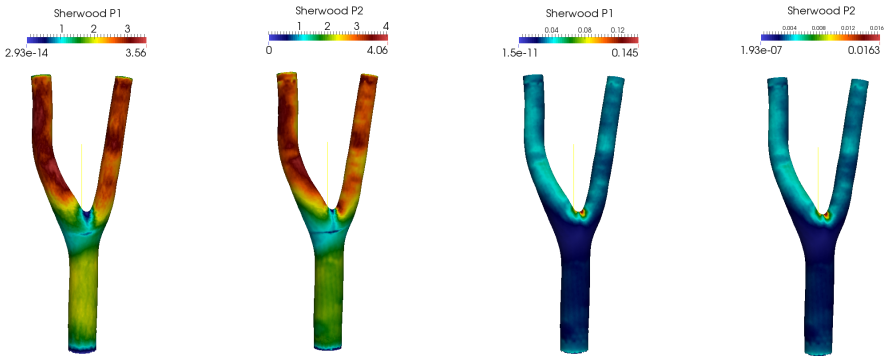


(a) Physical domain Ω_f. (b) Velocity inlet and grid. (c) Velocity field.

Figure 5: Physical domain and inlet velocity profile with mesh and velocity field.

3.2.2 Numerical results with the MSRB preconditioner

We now assess the computational performance of the MSRB preconditioner for the test case 2. We show results for both fixed accuracy (with $\delta_{RB,k} = 0.001$, $k = 0, 1, \dots$) and fixed dimension (with $N_k = 20$, $k =$



(a) $\boldsymbol{\mu} = (5 \cdot 10^{-5}, 10^{-4})$ (b) $\boldsymbol{\mu} = (5 \cdot 10^{-2}, 10^{-3})$

Figure 6: Sherwood number distribution for values of the parameter vector.

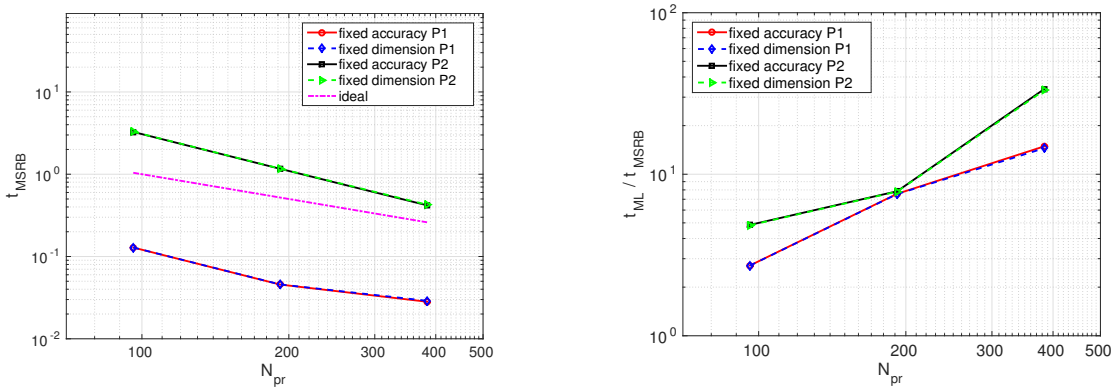
$0, 1, \dots$) approaches, see Tables 7 and 8, respectively, employing a number of processor $N_{cpu} = 96, 192, 384$. We first remark that very similar results are obtained either with the fixed accuracy or the fixed dimension approach. In both cases, the FGMRES method with the MSRB preconditioner converges in 3 iterations (or even less), both for $\mathcal{P}_1$ and $\mathcal{P}_2$ finite element: employing different FE degrees does not impact on the dimension of the reduced spaces, and consequently on the time needed for the solution online of the reduced problems. On the other hand, employing $\mathcal{P}_2$ FE has a huge impact on the performances of the $\mathbf{P}_{\mathrm{ML}}(\boldsymbol{\mu})$ preconditioner: the iteration count is three times higher and the overall computational times largely increase. Moreover, thanks to the small sizes of the RB spaces, the computational times obtained with $\mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ in the online phase are mainly governed by the construction of the fine preconditioner $\mathbf{P}_{\mathrm{BJ}}(\boldsymbol{\mu})$, which is embarrassingly parallel, thus yielding a very good overall scalability, see Fig. 7a. Indeed, the computational time is mainly governed by the LU factorizations of the local matrices in $\mathbf{P}_{\mathrm{BJ}}(\boldsymbol{\mu})$. On the other hand, solving the linear system with $\mathbf{P}_{\mathrm{ML}}(\boldsymbol{\mu})$ (and consequently the offline phase as it mainly involves snapshots computation) results in a larger time when using 384 CPUs due to the communication costs of the ML preconditioner. In Fig. 7b we report the speedup in computational time obtained by employing $\mathbf{P}_{\mathrm{ML}}(\boldsymbol{\mu})$ and $\mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$: increasing the number of processors we solve the problem online up to 14 times faster than ML in the case of $\mathcal{P}_1$ elements and 35 in the case of $\mathcal{P}_2$ elements. In this case the break-even point (BEP) of online evaluations decreases with the number of processors up to about 450 (resp. about 500) for $\mathcal{P}_2$ (resp. $\mathcal{P}_1$) elements.

Table 7: Test case 2: results for FGMRES, fixed accuracy approach, $n_s = 300$.

	N_{cpu}	L	N_k	t_{MSRB} (It)	t_{GML} (It)	t_{off}	t_{n_s}	t_{POD}	BEP
P1	96	3	8 22 46	0.12852 (2)	0.35 (57)	260.85	258.37	2.48	1178
P1	192	3	8 22 45	0.0456 (2)	0.34 (62)	185.31	183.16	2.15	619
P1	384	3	8 22 44	0.0282 (2)	0.42 (67)	188.47	186.34	2.13	482
P2	96	3	9 24 50	3.2651 (2)	15.86 (177)	9172.65	9158.77	13.88	729
P2	192	3	9 23 49	1.1635 (2)	9.13 (195)	4935.78	4927.46	8.32	620
P2	384	3	9 23 49	0.4188 (2)	14.12 (401)	5877.46	5872.17	5.29	429

Table 8: Test case 2: results for FGMRES, fixed dimension approach, $n_s = 300$.

	N_{cpu}	L	N_k	t_{MSRB} (It)	t_{GML} (It)	t_{off}	t_{n_s}	t_{POD}	BEP
P1	96	4	20	0.12868 (2)	0.35 (57)	314.01	307.74	6.27	1419
P1	192	4	20	0.04576 (2)	0.34 (62)	208.85	205.67	3.18	698
P1	384	4	20	0.02896 (2)	0.42 (67)	201.1	198.05	3.05	515
P2	96	4	20	3.2818 (3)	15.86 (177)	10391.4	10370.12	21.28	827
P2	192	4	20	1.1689 (3)	9.13 (195)	5363.1	5350.8	12.3	674
P2	384	4	20	0.4264 (3)	14.12 (401)	6095.57	6087.78	7.79	446



(a) Scalability. (b) Speedup.

Figure 7: Test case 2: Scalability and speedup as function of the number of processors N_{cpu}.

4 Conclusions

In this paper we have proposed and analyzed a new two-level preconditioner based on the combination of a RB coarse component and a fine preconditioner for large-scale FE problems. In order to gain the maximum

efficiency from the reduced order model, instead of employing the standard RB method we have introduced an iteration-dependent coarse component, which at the k-th step of the iterative method is tailored to solve the k-th error equation. By employing such a strategy, we are able to tune the decay of the error at each step of the iterative method. We have first proposed the preconditioner and analyzed its properties in the amenable case of Richardson method; then, we have suitably modified it to accelerate the convergence rate of FGMRES iterations. We have proposed two approaches for constructing the RB spaces: *i*) a fixed accuracy approach, which ensures a constant decay of the error, and *ii*) a fixed dimension approach, which instead guarantees a limited number of basis functions for each RB space. We carried out several numerical tests to verify the performance of the MSRB preconditioner in the case of parametrized advection diffusion equations, showing that the proposed preconditioner, which is based on the parametrized physical model, enhances significantly the convergence of the preconditioned iterative method.
We have extensively investigated the performance of the MSRB preconditioner with respect to the grid size and the FE degree, highlighting a numerical independence of the dimension of the high-fidelity space, due to the use of RB coarse components that are indeed independent of this latter. We have carried out a comparison with the standard RB method for a problem featuring a nonaffine parameter dependence, for which the MSRB preconditioner has given better results in less computational time than the standard RB method. Finally, results show that the MSRB preconditioner is a promising technique, overcoming a severe computational bottleneck of the RB method, which requires the use of hyper-reduction techniques, and competitive with AMG and Krylov subspace recycling methods for challenging modeling and numerical scenarios.

A The reduced basis method for parametrized PDEs

The reduced basis (RB) method relies on the idea that the $\boldsymbol{\mu}$-dependent solution of the $N_h \times N_h$ high-fidelity problem (1) can be well approximated by a linear combination of $N \ll N_h$ high-fidelity solutions corresponding to (suitably chosen) parameter values. We report a a brief overview of the method to make the paper self-contained and fully understandable to those readers less acquainted with reduced order modeling.

The RB method is based on an *offline/ online* splitting: in the former phase a *reduced space* $V_N \subset V_h$, whose dimension is $N \ll N_h$, is built, algebraically represented by the matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, $\mathbf{V} = [\boldsymbol{\xi}_1 | \dots | \boldsymbol{\xi}_N]$; in the latter, the high-fidelity problem (1) is replaced by the reduced problem (4) for any new instance of the parameter $\boldsymbol{\mu}$. For an extensive introduction to the RB method see, e.g., [41, 29]; here we limit ourselves to recall the most remarkable points of this technique.

The construction of the RB space V_N can be performed by means of a (weak) greedy algorithm or proper orthogonal decomposition (POD). In particular, we recall the definitions and basic principles of the latter, since it is employed in the algorithms we propose. We start by computing n_s high-fidelity solutions $\{\mathbf{u}_h(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ (called snapshots) corresponding to selected parameter values $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$. POD then aims at compressing the snapshots data by finding the best N-dimensional subspace, with $N \leq n_s$, that approximates the space $V_{n_s} = span\{\mathbf{u}_h(\boldsymbol{\mu}_i),\ i = 1, \dots, n_s\}$. This is pursued by performing a singular value decomposition of the snapshot matrix $\mathbf{S} = [\mathbf{u}_h(\boldsymbol{\mu}_1), \mathbf{u}_h(\boldsymbol{\mu}_2), \dots, \mathbf{u}_h(\boldsymbol{\mu}_{n_s})]$, such that $V_{n_s} = Col(\mathbf{S})$, and resulting in a factorization

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

where $\mathbf{U} \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{Z} \in \mathbb{R}^{n_s \times n_s}$ and $\Sigma \in \mathbb{R}^{N_h \times n_s}$, such that $\Sigma_{ii} = \sigma_i$, $i = 1, \dots n_s$, $\Sigma_{ij} = 0$, $i \neq j$. Then, the first N columns of the matrix $\mathbf{U}$, $\mathbf{V} = \mathbf{U}(:, 1:N)$, form an orthonormal basis of a $N-$th dimensional subspace of V_h, that is, $Col(\mathbf{V}) = V_N$. This space results as the best N-dimensional approximation subspace, according to the following Proposition (see e.g. [41])

Proposition A.1. *Let* $\mathcal{V}_N = \{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T\mathbf{W} = \mathbf{I}_N\}$ *be the set of all N-dimensional orthonormal bases. Then*

$$\sum_{i=1}^{n_s} \|\mathbf{u}_h(\boldsymbol{\mu}_i) - \mathbf{V}\mathbf{V}^T\mathbf{u}_h(\boldsymbol{\mu}_i)\|_2^2 = \min_{\mathbf{W} \in \mathcal{V}_N} \sum_{i=1}^{n_s} \|\mathbf{u}_h(\boldsymbol{\mu}_i) - \mathbf{W}\mathbf{W}^T\mathbf{u}_h(\boldsymbol{\mu}_i)\|_2^2 = \sum_{i=N+1}^{n_s} \sigma_i^2. \tag{54}$$

Hence, the RB space V_N minimizes the projection error of the snapshots onto the reduced subspace of dimension N, among all possible N-dimensional subspaces of V_h. The POD method can be generalized to any matrix-induced norm. Specifically, given a symmetric positive definite matrix $\mathbf{Y}_h \in \mathbb{R}^{N_h \times N_h}$, we can define the scalar product $(\mathbf{x}, \mathbf{y})_{\mathbf{Y}_h} = (\mathbf{Y}_h\mathbf{x}, \mathbf{y})_2$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{N_h}$, inducing the norm $\|\mathbf{x}\|_{\mathbf{Y}_h}^2 = (\mathbf{x}, \mathbf{x})_{\mathbf{Y}_h}$, and consider the SVD decomposition of $\mathbf{Y}_h^{1/2}\mathbf{S}$. Then, the following proposition holds.

Proposition A.2. *Let* $\mathcal{V}_N = \{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T\mathbf{Y}_h\mathbf{W} = \mathbf{I}_N\}$ *be the set of all N-dimensional* $\mathbf{Y}_h$*-orthonormal bases. Then*

$$\sum_{i=1}^{n_s} \|\mathbf{u}_h(\boldsymbol{\mu}_i) - \mathbf{V}\mathbf{V}^T\mathbf{Y}_h\mathbf{u}_h(\boldsymbol{\mu}_i)\|^2_{\mathbf{Y}_h} = \min_{\mathbf{W}\in\mathcal{V}_N} \sum_{i=1}^{n_s} \|\mathbf{u}_h(\boldsymbol{\mu}_i) - \mathbf{W}\mathbf{W}^T\mathbf{Y}_h\mathbf{u}_h(\boldsymbol{\mu}_i)\|^2_{\mathbf{Y}_h} = \sum_{i=N+1}^{n_s} \sigma_i^2. \tag{55}$$

In other words, the POD method allows to compute the space of dimension N, that minimizes the $\mathbf{Y}_h$-projection error of the snapshots in the $\mathbf{Y}_h$-norm. Typically, in the Galerkin RB (G-RB) method for second-order elliptic PDEs, $\mathbf{Y}_h$ encodes the $H^1(\Omega)$ scalar product on the space V_h, that is, $(\mathbf{Y}_h)_{ij} = (\phi_j, \phi_i)_{H^1(\Omega)}$, $i, j = 1, \dots, N_h$, where $\{\phi_i\}_{i=1}^{N_h}$ denote the FE basis.

From a practical standpoint, POD is performed by solving an eigenvalue problem associated to the correlation matrix $\mathbf{C} = \mathbf{S}^T\mathbf{Y}_h\mathbf{S}$, whose eigenvalues directly provide the singular values squared σ_i^2, $i = 1, \dots, n_s$. Through the eigenvectors $\mathbf{w}_i$, $i = 1, \dots, n_s$ of $\mathbf{C}$ one can build a $\mathbf{Y}_h$-orthonormal basis $\{\boldsymbol{\xi}_i\}_{i=1}^{n_s}$ of the snapshots subspace V_{n_s}

$$\boldsymbol{\xi}_i = \frac{1}{\sigma_i}\mathbf{S}\mathbf{w}_i, \qquad i = 1, \dots, n_s. \tag{56}$$

Finally, the reduced space $V_N \subset V_{n_s}$ is built selecting the first N eigenvectors, given by the SVD, see [41] for the details. According to Proposition A.2, constructing the RB space with the first N eigenvectors yields a relative approximation accuracy on the snapshots equal to $\delta_{RB}^2 = \sum_{i=N+1}^{n_s} \sigma_i^2 / \sum_{i=1}^{n_s} \sigma_i^2$.

Therefore, if we aim at building a RB space relying on POD we can follow two approaches:

- $POD(\mathbf{S}, \mathbf{Y}_h, \delta_{RB})$: given a target accuracy δ_{RB}, we choose the first $N = N(\delta_{RB})$ columns of $\mathbf{U}$ as basis for the RB space V_N, where N is such that $\sum_{i=1}^{N} \sigma_i^2 / \sum_{i=1}^{n_s} \sigma_i^2 \geq 1 - \delta_{RB}^2$;
- $POD(\mathbf{S}, \mathbf{Y}_h, N)$: given a fixed dimension $N > 0$, we select the first N vectors.

Depending on the reducibility of the problem at hand, the relation between N and δ_{RB} can significantly vary. Once the RB space has been built, for any new instance of the parameter $\boldsymbol{\mu}$, the high-fidelity problem (1) is replaced by the reduced problem (4) which can be easily assembled and solved inexpensively, usually with direct methods. We underline that the matrix $\mathbf{A}_N(\boldsymbol{\mu})$ inherits the properties of $\mathbf{A}_h(\boldsymbol{\mu})$, being positive-definite for coercive problems, and therefore nonsingular. We point out that the matrix $\mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T$ is the pseudoinverse matrix of $\mathbf{A}_h(\boldsymbol{\mu})$, and approximates exactly $\mathbf{A}_h^{-1}(\boldsymbol{\mu})$ on the subspace V_N. Indeed, should the high-fidelity solution $\mathbf{u}_h(\boldsymbol{\mu})$ belong to the reduced space, that is $\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})$, it can be recovered as $\mathbf{A}_h^{-1}(\boldsymbol{\mu})\mathbf{f}_h(\boldsymbol{\mu}) = \mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T\mathbf{f}_h(\boldsymbol{\mu})$. However, we generally have $\mathbf{u}_h(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})$, that is, $\mathbf{u}_h(\boldsymbol{\mu})$ does not belong to V_N.

A vital assumption that allows to speed up the RB method is made by requiring that $\mathbf{A}_h(\boldsymbol{\mu})$ and $\mathbf{f}_h(\boldsymbol{\mu})$ depend affinely on the parameter $\boldsymbol{\mu}$, i.e. that they can be expressed as

$$\mathbf{A}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\mathbf{A}_h^q, \qquad \mathbf{f}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu})\mathbf{f}_h^q, \tag{57}$$

where $\Theta_a^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_a$ and $\Theta_f^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_f$ are $\boldsymbol{\mu}$-dependent functions, while the matrices $\mathbf{A}_h^q \in \mathbb{R}^{N_h \times N_h}$ and the vectors $\mathbf{f}_h^q \in \mathbb{R}^{N_h}$ are $\boldsymbol{\mu}$-independent. This property is crucial to achieve the full independence of the assembling of the RB arrays from the size N_h of the high-fidelity problem. In the case (57) is not automatically satisfied, such affine parametric dependence can be recovered through the use of algorithms based on the empirical interpolation method (EIM) and its discrete variants DEIM and M-DEIM, see [3, 15, 35], meaning that instead of equations (57) we have

$$\mathbf{A}_h(\boldsymbol{\mu}) \simeq \sum_{q=1}^{M_a} \tilde{\Theta}_a^q(\boldsymbol{\mu})\mathbf{A}_h^q, \qquad \mathbf{f}_h(\boldsymbol{\mu}) \simeq \sum_{q=1}^{M_f} \tilde{\Theta}_f^q(\boldsymbol{\mu})\mathbf{f}_h^q, \tag{58}$$

up to a certain tolerance, with M_a and M_f the number of selected basis computed by the corresponding algorithms. In the case of DEIM (resp. M-DEIM), the basis are again built through POD on a set of vector (resp. matrix) snapshots, and the coefficients $\tilde{\Theta}_f^q$ (resp. $\tilde{\Theta}_a^q$) are computed through an interpolation problem.

Acknowledgements

The authors are grateful to the anonymous referees and the editor for their constructive comments which have greatly contributed to the improvement of the paper. 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 Swiss National Supercomputing Centre (CSCS) for providing us the CPU resources under project ID s635.

References

[1] K. Ahuja, E. de Sturler, S. Gugercin, and E. R. Chang. Recycling BiCG with an application to model reduction. *SIAM Journal on Scientific Computing*, 34(4):1925–1949, 2012.

[2] A. C. Antoulas, C. A. Beattie, and S. Gugercin. Interpolatory model reduction of large-scale dynamical systems. In *Efficient Modeling and Control of Large-Scale Systems*, pages 3–58. Springer US, 2010.

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

[4] S. Bellavia, V. De Simone, D. Di Serafino, and B. Morini. Efficient preconditioner updates for shifted linear systems. *SIAM Journal on Scientific Computing*, 33(4):1785–1809, 2011.

[5] P. Benner and L. Feng. On recycling Krylov subspaces for solving linear systems with successive right-hand sides with applications in model reduction. In *Model Reduction for Circuit Simulation*, volume 74, pages 125–140. Springer, 2011.

[6] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

[7] M. Benzi and D. Bertaccini. Approximate inverse preconditioning for shifted linear systems. *BIT Numerical Mathematics*, 43(2):231–244, 2003.

[8] D. Bertaccini and F. Durastante. Interpolating preconditioners for the solution of sequence of linear systems. *Computers & Mathematics with Applications*, 72(4):1118–1130, 2016.

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

[10] S. Brenner and R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2007.

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

[12] M. Caputo, C. Chiastra, C. Cianciolo, E. Cutrì, G. Dubini, J. Gunn, B. Keller, F. Migliavacca, and P. Zunino. Simulation of oxygen transfer in stented arteries and correlation with in-stent restenosis. *International journal for numerical methods in biomedical engineering*, 29(12):1373–1387, 2013.

[13] K. Carlberg, V. Forstall, and R. Tuminaro. Krylov-subspace recycling via the pod-augmented conjugate-gradient method. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1304–1336, 2016.

[14] A. Chapman and Y. Saad. Deflated and augmented Krylov subspace techniques. *Numerical linear algebra with applications*, 4(1):43–66, 1997.

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

[16] Y. Chen, S. Gottlieb, and Y. Maday. Parametric analytical preconditioning and its applications to the reduced collocation methods. *Comptes Rendus Mathematique*, 352(7):661–666, 2014.

[17] C. Colciago, S. Deparis, and D. Forti. Fluid-structure interaction for vascular flows: from supercomputers to laptops. In *Fluid-Structure Interaction: Modeling, Adaptive Discretization and Solvers*. Walter de Gruyter, Berlin, Radon Series on Computational and Applied Mathematics, 2017.

[18] G. Coppola and C. Caro. Oxygen mass transfer in a model three-dimensional artery. *Journal of The Royal Society Interface*, 5(26):1067–1075, 2008.

[19] E. de Sturler. Nested Krylov methods based on GCR. *Journal of Computational and Applied Mathematics*, 67(1):15–41, 1996.

[20] E. De Sturler. Truncation strategies for optimal Krylov subspace methods. *SIAM Journal on Numerical Analysis*, 36(3):864–889, 1999.

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

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

[23] H. C. Elman and V. Forstall. Preconditioning techniques for reduced basis methods for parameterized elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 37(5):S177–S194, 2015.

[24] J. Erhel, K. Burrage, and B. Pohl. Restarted gmres preconditioned by deflation. *Journal of computational and applied mathematics*, 69(2):303–318, 1996.

[25] A. Ern and J.-L. Guermond. *Theory and practice of finite elements. Number 159 in Applied Mathematical Sciences*. Springer, New York, 2004.

[26] C. Farhat, L. Crivelli, and F. X. Roux. Extending substructure based iterative solvers to multiple load and repeated analyses. *Computer methods in applied mechanics and engineering*, 117(1-2):195–209, 1994.

[27] M. Ferronato, C. Janna, and G. Pini. Shifted FSAI preconditioners for the efficient parallel solution of non-linear groundwater flow models. *International Journal for Numerical Methods in Engineering*, 89(13):1707–1719, 2012.

[28] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. Ml 5.0 smoothed aggregation user's guide. Technical report, SAND2006-2649, Sandia National Laboratories, 2006.

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

[30] K. Ito and S. Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal Computational Physics*, 143(2):403–425, July 1998.

[31] D. Kressner and C. Tobler. Low-rank tensor krylov subspace methods for parametrized linear systems. *Research Reports of Seminar for Applied Mathematics, ETHZ*, 2010.

[32] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications (MS&A)*, pages 235–274. Springer International Publishing, Switzerland, 2014.

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

[34] R. B. Morgan. GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1):20–37, 2002.

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

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

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

[38] M. L. Parks, E. De Sturler, G. Mackey, D. D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674, 2006.

[39] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

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

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

[42] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37. Springer Verlag, 2007.

[43] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*, volume 10. Oxford University Press, 1999.

[44] A. Quarteroni, A. Veneziani, and P. Zunino. Mathematical and numerical modeling of solute dynamics in blood flow and arterial walls. *SIAM Journal on Numerical Analysis*, 39(5):1488–1511, 2002.

[45] F. Risler and C. Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numerical algorithms*, 23(1):1, 2000.

[46] F. X. Roux. 10. parallel implementation of a domain decomposition method for non-linear elasticity problems. In *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, pages 161–175. Society for Industrial and Applied Mathematics, 1995.

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

[48] Y. Saad. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics of computation*, 48(178):651–662, 1987.

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

[50] Y. Saad. Analysis of augmented Krylov subspace methods. *SIAM Journal on Matrix Analysis and Applications*, 18(2):435–449, 1997.

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

[52] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h. A deflated version of the conjugate gradient algorithm. *SIAM Journal on Scientific Computing*, 21(5):1909–1926, 2000.

[53] S. Sen, K. Veroy, D. Huynh, S. Deparis, N. Nguyen, and A. Patera. A "natural norm" a posteriori error estimators for reduced basis approximations. *Journal of Computational Physics*, 217(1):37–62, 2006.

[54] B. F. Smith, P. E. Bjørstad, and W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York, NY, USA, 1996.

[55] K. Stüben and U. Trottenberg. *Multigrid methods: Fundamental algorithms, model problem analysis and applications*. In *Multigrid Methods,* W. Hackbusch and U. Trottenberg, eds., volume 960, pages 1–176. Springer Verlag Berlin, 1982.

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

[57] A. J. Wathen. Preconditioning. *Acta Numerica*, 24:329–376, 5 2015.

[58] O. Zahm and A. Nouy. Interpolation of inverse operators for preconditioning parameter-dependent equations. *SIAM Journal on Scientific Computing*, 38(2):A1044–A1074, 2016.

Recent publications:

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics

Ecole Polytechnique Fédérale (EPFL)

CH-1015 Lausanne

19.2016 ANDREA MANZONI, FEDERICO NEGRI :
Automatic reduction of PDEs defined on domains with variable shape

20.2016 MARCO FEDELE, ELENA FAGGIANO, LUCA DEDÈ, ALFIO QUARTERONI:
A patient-specific aortic valve model based on moving resistive immersed implicit surfaces

21.2016 DIANA BONOMI, ANDREA MANZONI, ALFIO QUARTERONI:
A matrix discrete empirical interpolation method for the efficient model reduction of parametrized nonlinear PDEs: application to nonlinear elasticity problems

22.2016 JONAS BALLANI, DANIEL KRESSNER, MICHAEL PETERS:
Multilevel tensor approximation of PDEs with random data

23.2016 DANIEL KRESSNER, ROBERT LUCE:
Fast computation of the matrix exponential for a Toeplitz matrix

24.2016 FRANCESCA BONIZZONI, FABIO NOBILE, ILARIA PERUGIA:
Convergence analysis of Padé approximations for Helmholtz frequency reponse problems

25.2016 MICHELE PISARONI, FABIO NOBILE, PÉNÉLOPE LEYLAND:
A continuation multi level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible aerodynamics

26.2016 ASSYR ABDULLE, ONDREJ BUDÁČ:
Multiscale model reduction methods for flow in heterogeneous porous media

27.2016 ASSYR ABDULLE, PATRICK HENNING:
Multiscale methods for wave problems in heterogeneous media

28.2016 ASSYR ABDULLE, ORANE JECKER:
On heterogeneous coupling of multiscale methods for problems with and without scale separation

29.2016 WOLFGANG HACKBUSCH, DANIEL KRESSNER, ANDRÉ USCHMAJEW:
Perturbation of higher-order singular values

30.2016 ASSYR ABDULLE, ONDREJ BUDÁČ, ANTOINE IMBODEN:
Multiscale methods and model order reduction for flow problems in three-scale porous media

31.2016 DANIEL KRESSNER, ANA ŠUŠNJARA:
Fast computation of spectral projectors of banded matrices

32.2016 NEW NICCOLO DAL SANTO, SIMONE DEPARIS, ANDREA MANZONI, ALFIO QUARTERONI:
Multi space reduced basis preconditioners for large-scale parametrized PDEs