DE GRUYTER
OPEN

Communications in Applied
and Industrial Mathematics
ISSN 2038-0909

Research Article

Commun. Appl. Ind. Math. 9 (1), 2017, 282–297

DOI: 10.1515/caim-2017-0015

A numerical investigation of multi space reduced basis preconditioners for parametrized elliptic advection-diffusion equations

Niccolò Dal Santo[1*], Simone Deparis[1] , Andrea Manzoni[1]

[1]CMCS, École Polytechnique Fédérale de Lausanne (EPFL),
Station 8, 1015 Lausanne, Switzerland.

[*]Email address for correspondence: niccolo.dalsanto@epfl.ch

Communicated by Giorgio Fotia
Received on 06 22, 2017. Accepted on 11 28, 2017.

Abstract

We analyze the numerical performance of a preconditioning technique recently proposed in [1] for the efficient solution of parametrized linear systems arising from the finite element (FE) discretization of parameter-dependent elliptic partial differential equations (PDEs). In order to exploit the parametric dependence of the PDE, the proposed preconditioner takes advantage of the reduced basis (RB) method within the preconditioned iterative solver employed to solve the linear system, and combines a RB solver, playing the role of coarse component, with a traditional fine grid (such as Additive Schwarz or block Jacobi) preconditioner. A sequence of RB spaces is required to handle the approximation of the error-residual equation at each step of the iterative method at hand, whence the name of Multi Space Reduced Basis (MSRB) method. In this paper, a numerical investigation of the proposed technique is carried on in the case of a Richardson iterative method, and then extended to the flexible GMRES method, in order to solve parameterized advection-diffusion problems. Particular attention is payed to the impact of anisotropic diffusion coefficients and (possibly dominant) transport terms on the proposed preconditioner, by carrying out detailed comparisons with the current state of the art algebraic multigrid preconditioners.

Keywords: Finite elements, preconditioners, reduced basis, high performance computing, parametrized advection-diffusion.

AMS subject classification: 65N30, 65F08, 65F10

1. Introduction.

The accurate numerical approximation of parametrized partial differential equations (PDEs), that is, PDEs depending on parameters which describe physical and geometrical properties of the model, is computationally challenging whenever interested to characterize their solutions for a wide range of parameter values. By relying on high-fidelity solvers such as the finite element (FE) method [2–4] and spectral methods [4,5], a discretized problem is obtained through a (Petrov-)Galerkin projection onto a subspace V_h of dimension N_h, $10^4 \leq N_h \leq 10^{10}$, thus yielding a parametrized linear system

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

to solve for each new parameter instance. Here $\mathbf{u}_h(\boldsymbol{\mu})$, $\mathbf{f}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and $\mathbf{A}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix; $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p$, $p \geq 1$, is a vector of p parameters. When dealing with 3D problems, the linear system (1) is usually solved by preconditioned iterative methods, such as the preconditioned conjugate gradient (PCG) or the preconditioned GMRES methods, which provide scalable and optimal solvers if suitable preconditioners are employed, for instance domain decomposition (DD) or multilevel (ML) methods [6–9]. Nevertheless, solving problem (1) for a large number of parameter instances may become a critical issue because of the huge CPU time required by each query to the high-fidelity solver; indeed, classical preconditioning methods do not generally exploit the parametric dependence of the PDE. Solving the parametrized linear system for several values of the parameter is often required in those *many-query* industrial contexts such as, e.g., sensitivity analysis, uncertainty quantification or PDE-constrained optimization. As a matter of fact, the solution of parametrized linear systems is a challenging research area, for which several contributions have been proposed in the last decade. Reduced order modeling (ROM) techniques, and in particular the reduced basis (RB) method, represent an accurate, reliable and efficient tool to deal with parametrized problems, see e.g. [10,11]. Given $\boldsymbol{\mu} \in \mathcal{D}$, the RB method seeks an approximation of the high-fidelity solution $\mathbf{u}_h(\boldsymbol{\mu})$ in a *reduced* space that is spanned by a set of basis functions given by linear combinations of high-fidelity solutions corresponding to different parameter instances. As a matter of fact, a small RB linear system is solved in place of the large FE linear system. The RB method has been applied to several classes of equations, including elliptic, (nonlinear) saddle-point, parabolic problems.

More recently, alternative options have been proposed to tackle the original, high-fidelity parameterized FE system. In [12], a preconditioner based on the interpolation of the matrix inverse which is computed as a linear combination of a pre-computed basis of matrix inverses is proposed, and its performances are analyzed for relatively small problems. On the other hand, in [13], the authors combine proper orthogonal decomposition (POD) ROM techniques to truncate the augmenting Krylov subspace, allowing to retain only the high-energy modes of the recycled Krylov subspace. This technique allows to compute efficiently very accurate approximations to the FE solution, however it is specifically designed for symmetric problems.

The multi space reduced basis (MSRB) preconditioner, recently proposed in [1], takes advantage of the parametric dependence of the PDE to devise a preconditioning technique tailored for the whole class of linear systems obtained when the parameter vector varies in the selected parameter domain. Such a technique relies on a two-level approach which combines multiplicatively a *fine*, nonsingular preconditioner (e.g. Gauss-Seidel, Jacobi or one-level additive Schwarz) with a *coarse correction* RB solver. This latter exploits the parametric dependence to accelerate the convergence of the iterative method, while the former is a general-purpose (easy-to-implement, even if not efficient) method ensuring that the resulting preconditioner is nonsingular. The resulting MSRB preconditioning method relies on a sequence of RB spaces which are iteration-dependent. In particular, the k-th space is tailored to solve the error equation corresponding to the k-th step of the iterative method, whence the name of multi space reduced basis method. This setting leads to an iteration-dependent preconditioner, which has been proposed for Richardson and flexible GMRES (FGMRES) iterations [14].

In this work, we investigate the computational performances of the MSRB preconditioner in the case of large-scale linear systems arising from the discretization of advection-diffusion problems. We will focus on those cases where anisotropic diffusion coefficients/fields and/or dominant advection are considered. Both the Richardson and the FGMRES method are considered as iterative solver for the resulting linear systems. We compare the results, namely the iteration count and the wall time for solving the parametrized systems, with those obtained by employing an algebraic multilevel preconditioner (AMG). 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, then showing extremely good optimality and scalability properties.

The proposed MSRB approach is strongly related to the classical reduced basis (RB) method for the sake of solving problem (1); this is indeed the best option in those cases where $\mathbf{A}_h(\boldsymbol{\mu})$ and $\mathbf{f}_h(\boldsymbol{\mu})$ fulfill the so-called affine parametric dependence property, i.e. that they can be expressed as

$$\mathbf{A}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\mathbf{A}_h^q, \qquad \mathbf{f}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu})\mathbf{f}_h^q, \tag{2}$$

where $\Theta_a^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_a$ and $\Theta_f^q : \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_f$ 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. Such an assumption allows to uncouple the dependence of the matrix $\mathbf{A}_h(\boldsymbol{\mu})$ and the right hand side $\mathbf{f}_h(\boldsymbol{\mu})$ from the parameters, and is crucial to achieve the full independence of the assembling the RB arrays from the size N_h of the high-fidelity problem. However, involved parameter dependences (such as those occurring when dealing with PDEs on domains with variable shape) do not provide these expressions automatically built-in when formulating the problem – that is, problems are naturally nonaffine – thus requiring for extensive hyper-reduction algorithms to restore, in an approximate way, such a decomposition. Such a preprocessing stage can easily become very expensive if the parametric dependence is involved and a very good accuracy has to be fulfilled; this latter requirement is crucial to ensure the overall accuracy of the RB approximation, and usually entails huge – often unaffordable – computational costs in the case of very large-scale systems. The proposed MSRB preconditioner has been shown to be a viable alternative to standard RB methods to face the complexity of nonaffine problems in a large-scale context.

The structure of the paper is as follows. In Section 2 we briefly recall the POD-RB method for parametrized elliptic PDEs and in Section 3 we report the motivation and the essential ingredients for the construction of MSRB preconditioners in the Richardson and FGMRES cases. In Section 4 we present numerical results obtained for advection-diffusion parametrized PDEs and in Section 5 we draw some conclusions.

2. The reduced basis method for parametrized PDEs.

The reduced basis (RB) method for parametrized PDEs relies on the assumption 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 $\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu})$ of $N \ll N_h$ global basis functions $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ obtained from a set of snapshots, that is, high-fidelity solutions corresponding to (suitably chosen) parameter values. A possible strategy for the construction of the basis functions is detailed below; here we denote by

$$\mathbf{V} = [\boldsymbol{\xi}_1 | \ldots | \boldsymbol{\xi}_N] \in \mathbb{R}^{N_h \times N}$$

the algebraic representation of the RB functions. The degrees of freedom $\mathbf{u}_N(\boldsymbol{\mu}) = [u_N^1(\boldsymbol{\mu}), \ldots, u_N^N(\boldsymbol{\mu})]^T \in \mathbb{R}^N$ of the RB approximation are computed by solving the following RB problem,

$$\mathbf{V}^T\mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{V}^T\mathbf{f}_h(\boldsymbol{\mu}), \tag{3}$$

obtained by performing a Galerkin projection of the original problem onto the RB subspace $V_N \subset V_h$.

Equivalently, we can introduce the RB arrays, obtained from the corresponding high-fidelity arrays, as

$$\mathbf{A}_N(\boldsymbol{\mu}) = \mathbf{V}^T\mathbf{A}_h(\boldsymbol{\mu})\mathbf{V} \in \mathbb{R}^{N\times N}, \qquad \mathbf{f}_N(\boldsymbol{\mu}) = \mathbf{V}^T\mathbf{f}_h(\boldsymbol{\mu}) \in \mathbb{R}^N. \tag{4}$$

and express the RB problem under the more convenient form

$$\mathbf{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu}). \tag{5}$$

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

$$\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{f}_N = \mathbf{V}\mathbf{A}_N^{-1}(\boldsymbol{\mu})\mathbf{V}^T\mathbf{f}_h(\boldsymbol{\mu}) \approx \mathbf{u}_h(\boldsymbol{\mu}). \tag{6}$$

Once an extensive offline stage has been performed – including the construction of the basis and the construction of the arrays required to assemble the RB matrix and vector, thanks to the assumption of affine parametric dependence – problem (5) can be easily solved, usually by means of direct methods, since $N \ll N_h$. For an extensive look to the RB method see, e.g., [10]; here we limit ourselves to recall the most remarkable points of this technique, instrumental to setting the proposed preconditioning technique. dataCoarseLsrbSolver The construction of the RB space V_N can be performed by means of a (weak) greedy algorithm or proper orthogonal decomposition (POD). Because of its flexibility and the fact that it does not require the evaluation of error bounds or indicators for the selection of basis functions, we rely on this latter, see e.g. [10,11,15] for a further understanding of (weak) greedy algorithms and POD. We start by computing n_s high-fidelity solutions $\{\mathbf{u}_h(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ (called snapshots) corresponding to the parameter values $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$. POD then aims at finding the best N-dimensional subspace, with $N \leq n_s$, that approximates the space $V_{n_s} = span\{\mathbf{u}_h(\boldsymbol{\mu}_i),\ i = 1,\dots,n_s\}$. In particular, we denote by $\mathbf{Y}_h \in \mathbb{R}^{N_h\times N_h}$ a symmetric positive definite matrix and 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 $\|\mathbf{x}\|^2_{\mathbf{Y}_h} = (\mathbf{x},\mathbf{x})_{\mathbf{Y}_h}$. Moreover, given a matrix $\mathbf{X} \in \mathbb{R}^{N_h\times n}$, we refer by $Col(\mathbf{X})$ to the vector space generated by the columns of $\mathbf{X}$. Then, the construction of the RB space is pursued by performing a singular value decomposition (SVD) of the matrix $\mathbf{Y}_h^{1/2}\mathbf{S} = \mathbf{Y}_h^{1/2}[\mathbf{u}_h(\boldsymbol{\mu}_1), \mathbf{u}_h(\boldsymbol{\mu}_2), \dots, \mathbf{u}_h(\boldsymbol{\mu}_{n_s})]$, resulting in the factorization

$$\mathbf{Y}_h^{1/2}\mathbf{S} = \mathbf{U}\Sigma\mathbf{Z}^T,$$

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 $\Sigma_{i,i} = \sigma_i$, $i = 1,\dots n_s$, $\Sigma_{i,j} = 0$, $i \neq j$. Then, the first N columns of the matrix $\mathbf{U}$ form an $\mathbf{Y}_h-$orthonormal basis for the best N-dimensional approximation subspace V_N, whose matrix representation is $\mathbf{V} = \mathbf{U}(:,1:N)$, i.e. $Col(\mathbf{V}) = V_N$. Indeed, among all possible N-dimensional $\mathbf{Y}_h$-orthonormal bases $\mathcal{V}_N = \{\mathbf{W} \in \mathbb{R}^{N_h\times N} : \mathbf{W}^T\mathbf{Y}_h\mathbf{W} = \mathbf{I}_N\}$, the POD method seeks the subspace of dimension N such that

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

i.e. such that the sum of the projection errors obtained when the snapshots are projected onto $\mathbf{W}$, evaluated in the $\mathbf{Y}_h$-norm, is minimized. In the context of second-order elliptic PDEs, the matrix $\mathbf{Y}_h$ is chosen to recover the $H^1(\Omega)$ scalar product on the FE space V_h.

In practice, the basis is built by solving the eigenvalue problems $\mathbf{C}\mathbf{w}_i = \sigma_i^2 \mathbf{w}_i$, $i = 1, \dots, n_s$ where $\mathbf{C} = \mathbf{S}^T \mathbf{Y}_h \mathbf{S} \in \mathbb{R}^{n_s \times n_s}$ is the (symmetric positive definite) correlation matrix. The eigenvalues σ_i^2, $i = 1, \dots, n_s$, are the squared singular values of $\mathbf{Y}_h^{1/2}\mathbf{S}$, whose eigenvectors

$$\boldsymbol{\xi}_i = \frac{1}{\sigma_i}\mathbf{S}\mathbf{w}_i, \qquad i = 1, \dots, n_s \tag{7}$$

form a $\mathbf{Y}_h$-orthonormal basis of the snapshots subspace $Col(\mathbf{S})$. The RB space V_N is built selecting the first N eigenvectors, also called high-modes, of $\mathbf{Y}_h^{1/2}\mathbf{S}$. In the RB space V_N, an approximation of the snapshots with an accuracy of order

$$\delta_{RB} \;=\; \sqrt{\sum_{i=N+1}^{n_s} \sigma_i^2 / \sum_{i=1}^{n_s} \sigma_i^2},$$

can be found, see [10] for further details.

3. Multi space RB preconditioners for parametrized PDEs.

The underlying idea of the proposed approach is to exploit the RB method to build efficient preconditioners for the iterative solution of (1) featuring uniform performance in the parameter space. Indeed, the goal is to implement a preconditioner which *(i)* can easily handle PDE problems within a prescribed class, although showing remarkable differences in the physical regime, and that *(ii)* can be efficiently built for several instances of the PDE problem, relying on common structures that can be pre-computed and stored. A more in-depth analysis, which goes beyond the goal of this paper, is reported in [1].

3.1. *MSRB preconditioners for the Richardson method.*

In this section we report the construction of the MSRB preconditioner for the Richardson method; the case of the FGMRES iterations will be addressed in Section 3.2. Given a matrix $\mathbf{Q}(\boldsymbol{\mu})$, consider the iterative method

$$\begin{cases} \mathbf{u}^{(k)}(\boldsymbol{\mu}) & = \mathbf{u}^{(k-1)}(\boldsymbol{\mu}) + \mathbf{Q}(\boldsymbol{\mu})\mathbf{r}^{(k-1)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots, \\ \mathbf{u}^{(0)}(\boldsymbol{\mu}) & = \mathbf{u}_0 \end{cases} \tag{8}$$

where $\mathbf{u}^{(k)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the $k-$th iterate and

$$\mathbf{r}^{(k)}(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(k)}(\boldsymbol{\mu}), \qquad k = 0, 1, \dots$$

denotes the residual of problem (1) at the $k-$th step. If $\mathbf{Q}(\boldsymbol{\mu})$ is non singular, (8) can be regarded as a Richardson iteration (with acceleration constant equal to 1) for the preconditioned system

$$\mathbf{Q}(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{Q}(\boldsymbol{\mu})\mathbf{f}_h(\boldsymbol{\mu}), \tag{9}$$

with preconditioner $\mathbf{Q}^{-1}(\boldsymbol{\mu})$. Note that

$$\mathbf{Q}(\boldsymbol{\mu})\mathbf{A}_h(\boldsymbol{\mu})(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}^{(k-1)}(\boldsymbol{\mu})) = \mathbf{Q}(\boldsymbol{\mu})(\mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(k-1)}(\boldsymbol{\mu})), \tag{10}$$

where $\mathbf{e}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}^{(k)}(\boldsymbol{\mu})$ denotes the error at step k. Hence, the left multiplication by $\mathbf{Q}(\boldsymbol{\mu})$ enables to compute an approximation of the solution $\mathbf{e}^{(k-1)}(\boldsymbol{\mu})$ of the error equation

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{e}^{(k-1)}(\boldsymbol{\mu}) = \mathbf{r}^{(k-1)}(\boldsymbol{\mu}). \tag{11}$$

Depending on the construction of the matrix $\mathbf{Q}(\boldsymbol{\mu})$, several different approximations of the solution of problem (11) can be obtained. Because of the parametric dependence, we aim at providing an accurate approximation of the error $\mathbf{e}^{(k-1)}(\boldsymbol{\mu})$ through the RB method. This is done by setting the preconditioner as follows

$$\mathbf{Q}(\boldsymbol{\mu}) \equiv \mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu}) = \mathbf{P}_h^{-1}(\boldsymbol{\mu}) + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big), \tag{12}$$

that is by a multiplicative combination of $\mathbf{P}_h^{-1}(\boldsymbol{\mu})$ and $\mathbf{Q}_{N_k}(\boldsymbol{\mu})$, where $\mathbf{P}_h^{-1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is a nonsingular fine grid preconditioner and $\mathbf{Q}_{N_k}(\boldsymbol{\mu})$ is an iteration-dependent coarse component which is a tailored for the error equation (11). When using a multiplicative combination of two preconditioners, the Richardson iterations can be rewritten by means of two half-steps, that is, if $\mathbf{Q}(\boldsymbol{\mu}) = \mathbf{Q}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$, (8) is equivalent to

$$\begin{cases} \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}) & = \mathbf{u}^{(k-1)}(\boldsymbol{\mu}) + \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{r}^{(k-1)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots, \\ \mathbf{u}^{(k)}(\boldsymbol{\mu}) & = \mathbf{u}^{(k-1/2)}(\boldsymbol{\mu}) + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}), \qquad k = 1, 2, \dots, \\ \mathbf{u}^{(0)}(\boldsymbol{\mu}) & = \mathbf{u}_0, \end{cases} \tag{13}$$

where $\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(k-1/2)}(\boldsymbol{\mu})$ is the residual after the first half-step. In particular, as it can be observed from the second equality of equation (13), the RB coarse component $\mathbf{Q}_{N_k}(\boldsymbol{\mu})$ must be constructed to approximate the solution of the following problem

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

or analogously

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}) = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{r}^{(k-1)}(\boldsymbol{\mu}), \tag{15}$$

which is obtained by plugging in (14) the definitions of $\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu})$ and $\mathbf{u}^{(k-1/2)}(\boldsymbol{\mu})$. We thus introduce at each step k a new RB space V_{N_k} that is trained on equation (15) and onto which an accurate approximation of $\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu})$ can be found, that is

$$V_{N_k} = span\Big\{\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_j)\Big\}_{j=1}^{N_k}, \tag{16}$$

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 construction of the RB method, we can obtain the arrays

$$\mathbf{V}_k = [\boldsymbol{\xi}_1^k | \dots | \boldsymbol{\xi}_N^k], \qquad \mathbf{A}_{N_k}(\boldsymbol{\mu}) = \mathbf{V}_k^T \mathbf{A}_h(\boldsymbol{\mu})\mathbf{V}_k, \tag{17}$$

where $\{\boldsymbol{\xi}_j^k\}_{j=1}^{N_k}$ denotes an orthonormalized basis for V_{N_k}. Finally, we define

$$\mathbf{Q}_{N_k}(\boldsymbol{\mu}) = \mathbf{V}_k \mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_k^T. \tag{18}$$

Remark 3.1. Under suitable assumptions on the fine preconditioner $\mathbf{P}_h(\boldsymbol{\mu})$ and the construction of the RB spaces V_{N_k}, $k = 1, \dots, L-1$, $\mathbf{Q}_{\text{MSRB},k}$ is nonsingular and we refer to its inverse as $\mathbf{Q}^{-1}_{\text{MSRB},k}(\boldsymbol{\mu}) = \mathbf{P}_{\text{MSRB},k}(\boldsymbol{\mu})$. Moreover, the error $\mathbf{e}^{(k)}(\boldsymbol{\mu})$ can be bounded as (see [1] for the proofs)

$$\|\mathbf{e}^{(k)}(\boldsymbol{\mu})\| \le C^k \delta \|\mathbf{e}^{(0)}\|, \qquad k = 1, \dots, L, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{19}$$

with $C = \left\| \mathbf{I}_{N_h} - \mathbf{P}_h^{-1}(\boldsymbol{\mu}) \mathbf{A}_h(\boldsymbol{\mu}) \right\|$ and $\delta = \prod_{j=1}^{k} \delta_j$ is the product of the tolerances $\{\delta_j\}_{j=0}^{L-1}$ such that

$$\|\mathbf{e}^{(k-1/2)} - \mathbf{V}_k \mathbf{e}^{(k-1/2)}_{N_k}\| \le \delta_k \|\mathbf{e}^{(k-1/2)}\| \qquad k = 1, \dots, L, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{20}$$

Hence, the relative error after L iterations will be under a certain tolerance ε_r provided the RB spaces are built fulfilling

$$\delta = \prod_{j=1}^{L} \delta_j < \varepsilon_r. \tag{21}$$

3.2. *Extension to flexible GMRES.*

In order to tackle large-scale applications, we rely on the flexible GMRES (FGMRES) method [14], reported in Algorithm 3.1, which is a variant of GMRES method which allows to employ an iteration-dependent operator as preconditioner, as it is the case for $\mathbf{P}_{\text{MSRB},k}$.

Algorithm 3.1 Flexible GMRES [6]

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

In Algorithm 3.1 $\mathbf{M}_k$ is the preconditioner operator used at iteration k, and aims at approximating the solution of the problem $\mathbf{A}\mathbf{c}_k = \mathbf{v}_k$, where $\mathbf{v}_k$ is the k-th Krylov Basis. In the MSRB case we have

$$\mathbf{M}_k^{-1}(\boldsymbol{\mu})\mathbf{v}_k = \mathbf{P}^{-1}_{\text{MSRB},k}(\boldsymbol{\mu})\mathbf{v}_k = \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_k + \mathbf{Q}_{N_k}(\boldsymbol{\mu})\Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{v}_k, \qquad k = 0, 1, \dots.$$

Consequently, the k-th RB space must be tailored to solve the following problem

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{y}^{(k)}(\boldsymbol{\mu}) = \Big(\mathbf{I}_{N_h} - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{P}_h^{-1}(\boldsymbol{\mu})\Big)\mathbf{v}_k, \qquad k = 0, 1, \dots, \tag{22}$$

yielding a RB space of the form

$$V_{N_k} = span\Big\{\mathbf{y}^{(k)}(\boldsymbol{\mu}_i)\Big\}_{i=1}^{N_k}, \qquad k = 0, 1, \dots,$$

where $\mathbf{y}^{(k)}(\boldsymbol{\mu}_i)$ is the solution of equation (22) with $\boldsymbol{\mu} = \boldsymbol{\mu}_i$. If we set as initial guess for the FGMRES method the solution of the standard RB problem (5), $\mathbf{u}^{(0)}(\boldsymbol{\mu}) = \mathbf{V}_0\mathbf{A}_{N,0}^{-1}(\boldsymbol{\mu})\mathbf{V}_0^T\mathbf{f}_h(\boldsymbol{\mu})$, which yields

$$\mathbf{r}_0(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(0)}(\boldsymbol{\mu}), \qquad \beta(\boldsymbol{\mu}) = \|\mathbf{r}_0(\boldsymbol{\mu})\|_2, \qquad \mathbf{v}_1 = \mathbf{r}_0(\boldsymbol{\mu})/\beta(\boldsymbol{\mu}), \tag{23}$$

the solution of problem (22) can be expressed as

$$\begin{cases} \beta(\boldsymbol{\mu}) = \|\mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}^{(0)}(\boldsymbol{\mu})\|_2, \\ \mathbf{y}^{(1)}(\boldsymbol{\mu}) = \dfrac{1}{\beta(\boldsymbol{\mu})}\big(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}^{(0)}(\boldsymbol{\mu})\big) - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_1, \\ \mathbf{y}^{(k+1)}(\boldsymbol{\mu}) = \dfrac{1}{h_{k+1,k}}\Big[\mathbf{z}_k(\boldsymbol{\mu}) - \displaystyle\sum_{j=1}^{k} h_{j,k}\big(\mathbf{y}^{(j)}(\boldsymbol{\mu}) + \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_j\big)\Big] - \mathbf{P}_h^{-1}(\boldsymbol{\mu})\mathbf{v}_{k+1}, \quad k \geq 1. \end{cases} \tag{24}$$

3.3. *Algorithmic procedures.*

The sequence of RB spaces for each step of the MSRB preconditioner is built recursively. By taking the (standard) RB approximation $\mathbf{u}^{(0)} = \mathbf{u}^{(0)}(\boldsymbol{\mu}) = \mathbf{V}_0\mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})\mathbf{V}_0^T\mathbf{f}_h(\boldsymbol{\mu})$ as initial guess, we set $V_{N_0} = V_N$. This space is obtained by choosing n_s values of the parameter $\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}$, computing the corresponding snapshots $\{\mathbf{u}_h(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ as the high-fidelity solutions of (1) and performing POD. Then, given the RB spaces $V_{N_0}, \dots, V_{N_k}$, we rely on POD to build the $(k+1)$-th space, where the errors $\{\mathbf{e}^{(k+1/2)}(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$ play the role of snapshots if the Richardson method is employed. These are computed according to the following relation

$$\mathbf{e}^{(k+1/2)}(\boldsymbol{\mu}_i) = \mathbf{u}_h(\boldsymbol{\mu}_i) - \mathbf{u}^{(k+1/2)}(\boldsymbol{\mu}_i) = \mathbf{e}^{(k)}(\boldsymbol{\mu}_i) - \mathbf{P}_h^{-1}(\boldsymbol{\mu}_i)\mathbf{r}^{(k)}(\boldsymbol{\mu}_i) \qquad \forall i = 1, \dots, n_s, \tag{25}$$

which is easily verified from (13). If FGMRES is instead employed, the POD is performed on the set of snapshots $\{\mathbf{y}^{(k+1)}(\boldsymbol{\mu}_i)\}_{i=1}^{n_s}$, which can be computed thanks to (24). We highlight that building the snapshots for the space k, employing equation (25) or (24), does not require to solve any additional linear system. As we are using POD to build each RB space, we choose 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 level.

The computation is divided in two phases: *offline* and *online*. In the former, the construction of the successive RB spaces keeps on until condition (21) is met, therefore, if we use a fixed tolerance δ_{RB} for all the RB spaces, we implicitly fix the number of spaces larger than $\lceil\log(\varepsilon_r)/\log(\delta_{RB})\rceil$. The detailed algorithm is reported in Algorithm 3.2.

Once the spaces V_{N_k}, $k = 0, 1, \dots, L-1$ have been generated, in the online phase it is possible to solve the high-fidelity system (1) by the chosen iterative method for any new instance of the parameter $\boldsymbol{\mu}$, which is expected to converge in less than L iterations. Should the number of iterations exceed the number of precomputed spaces L, we can either decide to use the last space as coarse correction in the subsequent iterations or use only the fine preconditioner component.

Algorithm 3.2 MSRB Preconditioner - Offline RB spaces construction

```
procedure MSRB-PRECONDITIONER($\{\boldsymbol{\mu}_i\}_{i=1}^{n_s}, \varepsilon_r, \delta_{RB}$)
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 $\mathbf{V}_k = POD(\mathbf{S}, \delta_{RB})$ and $\delta_{RB,k} = \delta_{RB}$
        Compute new snapshots $\mathbf{e}^{(k+1/2)}(\boldsymbol{\mu}_i) = \mathbf{e}^{(k)}(\boldsymbol{\mu}_i) - \mathbf{P}_h^{-1}(\boldsymbol{\mu}_i)\mathbf{r}^{(k)}(\boldsymbol{\mu}_i)$
        Set $\mathbf{S} = [\mathbf{e}^{(k+1/2)}(\boldsymbol{\mu}_1), \dots, \mathbf{e}^{(k+1/2)}(\boldsymbol{\mu}_{n_s})]$ and $k = k + 1$
    end while
end procedure
```

4. Numerical experiments.

Numerical experiments are presented to illustrate the capability of the proposed MSRB-preconditioner. We take into account several test cases governed by advection-diffusion (AD): we first focus on heat convection in a domain showing a piecewise constant (parametrized) thermal conductivity, to simulate the effect of different material properties in the domain. Then, we turn our attention to a parametrized advection diffusion problem within the same domain where also an advection term is present. These two examples, although simple from the point of view of the mathematical model and the geometrical configuration of the domain, feature solutions showing a wide variability because of the parameter dependence. Indeed, both anisotropic effects in the diffusion coefficient and advection dominant cases are studied, as typical situations challenging well-studied preconditioning techniques such as those based on multilevel (or multigrid) preconditioners because of the physical behavior of the solution, see e.g. [8,16].

Results for both the Richardson and the FGMRES method are presented. POD is always performed with $n_s = 1000$ snapshots and with respect to the scalar product induced by the symmetric positive definite matrix $\mathbf{Y}_h$, which represents the $H_0^1(\Omega)$ scalar product on V_h. A stopping criterion based on the Euclidean norm of the (finite element vector of the) residual, rescaled with respect to the Euclidean norm of the right hand side of the system, with a tolerance $\epsilon_r = 10^{-7}$ has been used for all the tests. Furthermore, we build the RB spaces such that inequality (21) is satisfied with $\delta = 10^{-9}$, since POD is optimal in the sense of minimizing the sum of the squared projection errors onto the reduced space evaluated on the selected snapshots, and when the reduced solution for a different parameter is computed, the corresponding error can be slightly larger. As fine preconditioner, we employ $\mathbf{P}_h(\boldsymbol{\mu}) = \mathbf{P}_{\mathrm{BJ}}(\boldsymbol{\mu})$, a Block Jacobi preconditioner, where each block represents the restrictions to the degrees of freedom of a subdomain selected by Parmetis[a] at the mesh level. If the number of iterations required by the iterative solver to reach the prescribed tolerance ϵ_r exceeds the number of spaces (which is fixed once the offline phase has been completed) the final iterations employ just the fine preconditioner, i.e. $\mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu}) = \mathbf{P}_h(\boldsymbol{\mu})\ \forall k \geq L$.

The results we present are obtained with the FE library LifeV[b], see [17]. The parallel implementation relies on the Trilinos software, and in particular on the Epetra package for the parallel management of the FE matrices and vectors and on the Ifpack package for the construction of the Block Jacobi preconditioner $\mathbf{P}_{\mathrm{BJ}}(\boldsymbol{\mu})$, where each block represents the in-

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

[b] www.lifev.org

verse of the local stiffness matrix corresponding to one subdomain generated by Parmetis. Concerning the solutions of the RB systems, i.e. the computation of $\mathbf{A}_{N_k}^{-1}(\boldsymbol{\mu})$ in (18), the very small number of RB functions allows to have RB problems of small size; consequently the sequential LU factorization is employed. For all the simulations[c], we report the number of spaces L and RB functions N_k, $k = 0, 1, \dots$ produced by Algorithm 3.2, the results obtained online with the MSRB preconditioner averaging on $N_{onl} = 250$ parameters and the computational time t_{off} required by the offline phase, which is the one needed to build the the RB spaces of the MSRB preconditioner by Algorithm 3.2. For each new value of $\boldsymbol{\mu}$ these RB spaces are employed to build the RB coarse corrections. We compare the results with those obtained using an algebraic multigrid (AMG) preconditioner $\mathbf{P}_{\text{ML}}(\boldsymbol{\mu})$, that exploits an exact coarse component and 2-sweeps Gauss-Seidel smoother obtained with default settings from the Trilinos ML package [18], and used as preconditioner for the results obtained with both the Richardson method (noted as RML) and the GMRES method (noted as GML).

4.1. *Problem setting: advection-diffusion in a blockwise cubic domain.*

Given a regular domain $\Omega \subset \mathbb{R}^3$, such that $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$, we subdivide it into $\mathcal{J}$ subregions Ω_j, $j = 1, \dots \mathcal{J}$ s.t. $\bar{\Omega} = \cup_{j=1}^{\mathcal{J}} \bar{\Omega}_j$ and $\overset{\circ}{\Omega}_i \cap \overset{\circ}{\Omega}_j$, $i \neq j$. Let us consider the following parametrized PDE:

$$\begin{cases} -\nabla \cdot (\mathcal{K}(\mathbf{x})\nabla u) + \mathbf{b} \cdot \nabla u = f & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_D, \\ \mathcal{K}\nabla u \cdot \mathbf{n} = 0 & \text{on } \Gamma_N, \end{cases} \tag{26}$$

where we define the diffusion tensor $\mathcal{K}(\mathbf{x}) = \nu(\mathbf{x})\mathcal{K}_\epsilon = \nu(\mathbf{x})diag(1, 1, \epsilon_{an})$, with $\epsilon_{an} = \{1; 10^{-2}\}$ encoding the anisotropy magnitude, and $\nu(\mathbf{x}) > 0$ denoting the piecewise constant material properties on each Ω_j:

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

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 boundary conditions in a straightforward way. Problem (26) is parametrized with respect to the diffusion coefficients ν_j, $j = 1, \dots, \mathcal{J} - 1$, leading to the following definition of the parameter vector:

$$\boldsymbol{\mu} = (\nu_1, \dots, \nu_{\mathcal{J}-1}) \in \mathcal{D} = [10^{-2}, 1]^{\mathcal{J}-1}. \tag{27}$$

As spatial domain, we consider $\Omega = (0, 1)^3$ and investigate the performance in several modeling scenarios: note that the diffusion coefficients on the subregions are always parametrized, while the transport term and the anisotropy structure are either not present or taken equal to a prescribed value, and never considered to be parameter dependent. Nevertheless, by taking diffusion coefficients in the range (27) we take into account scenarios featuring a dominant advection, in particular when $\epsilon_{an} = 10^{-2}$.

[c] All our experiments have been carried out using the clusters Piz-Dora/Piz-Daint provided by the Swiss National Supercomputing Center (CSCS) on Cray XC40 machines.

A summary of the test cases considered in this section is reported in Table 1. In the first case, we focus on isotropic diffusion, in the second one on isotropic diffusion with a constant advection field $\mathbf{b} = [10yz(1-y)(1-z), 0, 0]^T$), whereas in the third one we model an anisotropic diffusion ($\epsilon_{an} = 10^{-2}$, which leads to dominant advection) with a constant advection field $\mathbf{b}$. We set the number of blocks $\mathcal{J} = 4$, and analyze the modeling scenarios **Tk**, **k** = 1,2,3, cf. Tab 1.

Concerning the high-fidelity discretization, we employ both linear and quadratic piecewise continuous finite elements (FE) on structured tetrahedral meshes; examples of solutions obtained in the case **T3** for different values of parameters are reported in Figure 1.

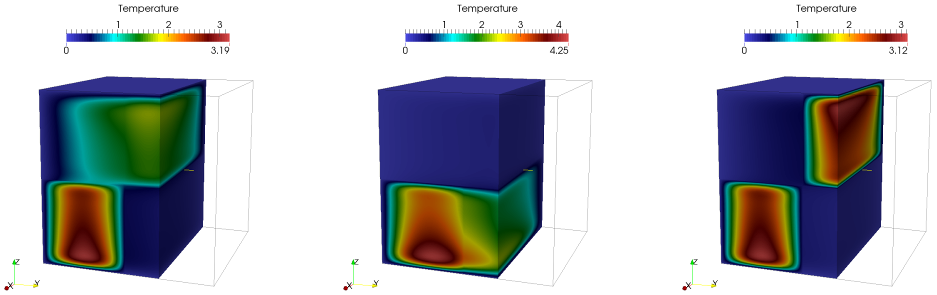

Figure 1: Example of solutions for test case **T3**, with a Neumann condition on $x = 1$.

Table 1: Considered models for numerical experiments.

# Test	$\mathbf{b}$	ϵ_{an}	**Model**	**Parameters**
T1	no	1	isotropic diffusion	$\boldsymbol{\mu} = (\nu_1, \ldots, \nu_{\mathcal{J}-1})$
T2	yes	1	isotropic diffusion with constant transport	$\boldsymbol{\mu} = (\nu_1, \ldots, \nu_{\mathcal{J}-1})$
T3	yes	10^{-2}	anisotropic diffusion with constant transport	$\boldsymbol{\mu} = (\nu_1, \ldots, \nu_{\mathcal{J}-1})$

We assess the performance of the proposed preconditioner in terms of number of iterations and computational wall time (seconds) for both the Richardson and the FGMRES method. In particular, we carry out the following analyses:

- in Section 4.1.1 we report the results obtained with the Richardson method for the isotropic diffusion test case **T1**, to show relevant features of the algorithm, namely the number of RB functions of the spaces V_{N_k}, $k = 0, 1, \ldots$ and the decay of the residual;
- in Section 4.1.2 we show the results obtained with the FGMRES method and identify the modeling scenarios where the preconditioner $\mathbf{P}_{\text{MSRB},k}(\boldsymbol{\mu})$ provides the best performances compared to $\mathbf{P}_{\text{ML}}(\boldsymbol{\mu})$;
- in Section 4.1.3 we consider different high-fidelity discretizations, with computational grids whose characteristic size is $h = 0.1 \cdot 2^{-l}$, $l = 2, 3, 4, 5$, yielding $N_h = 46'456$, $365'254$, $2'887'193$, $22'767'295$, respectively;
- in Section 4.1.4 we perform a strong scalability test with $\mathcal{P}_2$ finite elements, for $N_{cpu} = 2^l$, $l = 6, 7, \ldots 11$.

4.1.1. *Numerical results for the Richardson method.*

We start considering the Richardson method for the test case **T1**. In Table 2 we report the results for the former technique and a number of levels $L = 3$, 9, respectively built with a fixed tolerance equal to 0.001 and 0.1 respectively. All simulations have been run with an underlying mesh with $N_h = 365'254$ degrees of freedom, on $N_{cpu} = 96$ processors.

The results show that generating different levels with the same tolerance yields RB spaces whose dimensions grow with the iteration count k, see Table 2. This fact is also confirmed by the decay of the eigenvalues of the correlation matrix $\mathbf{C}_k = \mathbf{S}_k^T \mathbf{Y}_h \mathbf{S}_k$, reported in Figure 2 for $L = 3, 9$. Here $\mathbf{S}_k = [\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_1)| \dots |\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}_{n_s})]$ denotes the snapshots matrix employed for the construction of level k. As k grows, the decay of the eigenvalues is slower, so that larger RB spaces are needed to reach the same tolerance. This behavior can be ascribed to the fact that at step k the manifold $\mathcal{M}_k = \{\mathbf{e}^{(k-1/2)}(\boldsymbol{\mu}),\ \boldsymbol{\mu} \in \mathcal{D}\}$ is less regular compared to $\mathcal{M}_0, \dots, \mathcal{M}_{k-1}$: the higher k, then the more noisy the pattern of the error, the smaller its magnitude and the more difficult its approximation.

Once a new instance of the parameter $\boldsymbol{\mu}$ is considered, the computational time to solve the linear system with the MSRB-preconditioned Richardson method is reported as t_{MSRB}, which also takes into account the construction of $\mathbf{P}_h(\boldsymbol{\mu})$ and the coarse RB corrections $\mathbf{Q}_{N_k}(\boldsymbol{\mu})$, $k = 0, 1, \dots$ defined in (18). Notice that the latter mainly involves the assembly of $\mathbf{A}_{N_k}(\boldsymbol{\mu})$, since the RB spaces $\mathbf{V}_k$ are pre-constructed in the offline phase, whose computational time is reported as t_{off}. The computational time employed to solve the same problem with the AMG-preconditioned Richardson method and the AMG-preconditioned GMRES are reported as t_{RML} and t_{GML}, respectively, and also account for the construction of the AMG preconditioner. The results show that both the iteration count and the computational wall time are lower than the ones obtained employing the ML preconditioner in the Richardson method (RML) and the GMRES (GML) for all the numerical tests considered. We notice that in some test cases the number of iterations needed to reach the prescribed convergence tolerance exceeds the number of spaces L. In this circumstance, we employ only $\mathbf{P}_h(\boldsymbol{\mu})$ for all the remaining iterations. However, we highlight that building RB spaces with a lower local tolerance $\delta_{RB,k}$ reduced the number of additional iterations needed by the method.

In order to assess the performance of the $\mathbf{P}_{\mathrm{MSRB},k}(\boldsymbol{\mu})$ preconditioner at each iteration, we define the coefficient $\eta_{RB}^{(k)} = \eta_{RB}^{(k)}(\boldsymbol{\mu}), k = 0, 1, \dots,$ as

$$(28) \qquad \eta_{RB}^{(k)} = \frac{\|\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{Q}_{N_k}(\boldsymbol{\mu})\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu})\|_{\mathbf{Y}_h^{-1}}}{\|\mathbf{r}^{(k-1/2)}(\boldsymbol{\mu})\|_{\mathbf{Y}_h^{-1}}} \qquad k = 0, \dots, L-1,$$

which measures the decay of the residual at step k; its evaluation is reported in Figure 2 for $L = 9, 3$. All the RB spaces provide an online accuracy proportional to the prescribed tolerance $\delta_{RB,k}$, thus yielding a constant decay of the residual.

Table 2: Richardson method with $\mathcal{P}_1$ FE: $\delta_{RB,k} = 0.1,\ 0.001$, $k = 0, \dots L-1$.

	L	$\delta_{RB,k}$	N_k, $k = 0, \dots, L-1$	t_{MSRB} $[s]$ (It)	t_{RML} $[s]$ (It)	t_{GML} $[s]$ (It)	t_{off} $[s]$
T1	9	0.1	3 7 10 18 33 48 75 100 137	0.14 (18)	0.27 (31)	0.16 (12)	1093.51
T1	3	0.001	11 50 151	0.08 (6)	0.27 (31)	0.16 (12)	429.06

4.1.2. *Numerical results for the FGMRES method.*

We now turn to the case of the FGMRES method. Table 3 reports the results obtained with all the modeling cases **T1**, **T2** and **T3** with an underlying mesh with $N_h = 365'254$ degrees of freedom and $N_{cpu} = 96$ processors. Let us focus on the case of isotropic diffusion **T1**: we notice

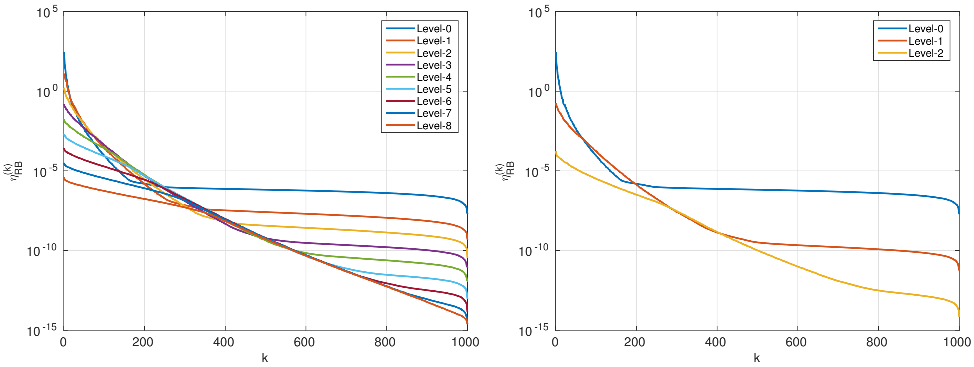

Figure 2: Eigenvalues of the correlation matrix $\mathbf{C}_k$, $k = 0, \ldots, L-1$ for modeling case **T1**, $N_h = 365'254$.

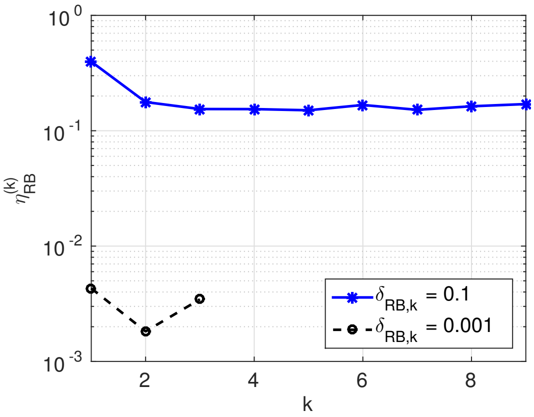

Figure 3: Evaluation of $\eta_{RB}^{(k)}$ for modeling case **T1**, $N_h = 365'254$.

that the FGMRES method gives, as expected, better results than the Richardson method at the online stage, even though it induces spaces of larger dimension: the computational times are comparable but the iteration counts are smaller. Concerning a comparison with the results obtained with the preconditioner ML, the total iteration count and the computational times are lower than the ones obtained with GML for all the simulations. For the model **T1**, isotropic diffusion, ML is known to perform extremely well, see e.g. [6,8], and the MSRB-preconditioners only slightly improves the outcome. However, for the models **T2** and especially **T3** the performances of ML worsen, while the ones of the MSRB preconditioner are impacted only in the number of basis functions of the RB coarse components, which does not affect much the performances of the iterative solver, both in terms of iteration count and computational time.

Table 3: FGMRES with $\mathcal{P}_1$ FE: $\delta_{RB,k} = 0.1$, 0.001.

	L	$\delta_{RB,k}$	N_k, $k = 0, \ldots, L-1$	t_{MSRB} $[s]$ (It)	t_{GML} $[s]$ (It)	t_{off} $[s]$
T1	9	0.1	3 7 10 16 32 55 97 155 231	0.15 (13)	0.16 (12)	1403.98
T2	9	0.1	4 11 23 48 87 143 217 277 233	0.16 (13)	0.20 (21)	1460.79
T3	9	0.1	4 11 21 38 65 105 164 235 217	0.19 (21)	0.32 (40)	1552.77
T1	3	0.001	11 53 177	0.09 (2)	0.16 (12)	516.25
T2	3	0.001	22 112 294	0.10 (4)	0.20 (21)	524.55
T3	3	0.001	22 111 290	0.11 (6)	0.32 (40)	646.57

4.1.3. *Dependence on the grid size.*

In this section we carry out an analysis with respect to four different grids whose characteristic dimensions are $h = 0.1$, 0.05, 0.025, 0.0125, leading to dimensions $N_h = 46'456$, $365'254$, $2'887'193$ and $22'767'295$, respectively, for the high-fidelity $\mathcal{P}_1$ FE approximation. These simulations have been carried out with $N_{cpu} = 12$, 96, 768, 6144 processors, respectively, in order to maintain a constant number of degrees of freedom (about 3800) per processor.

The results reported in Table 4 confirm that the iterations and the computational times do not rapidly grow by increasing the FE dimension, contrarily to what happens when other preconditioners are employed. Indeed, neither the size nor the accuracy of the RB problems are affected by the FE dimension, yielding comparable times and iterations for the different grid size. The problem with the finest grid is solved on average in $0.26\,s$, making possible a real time evaluation of the solution for new instances of the parameter. Indeed, the speed up gained by the use of the MSRB technique compared to the ML preconditioner, shown in Figure 4, is a monotonically increasing function of the FE dimension, leading to a speed up of 200 for the largest FE dimension. The break-even point (BEP) of online evaluations, i.e. the number of online parameters needed to repay the time invested in the offline phase, is reached for the finest grid with roughly 1200 online evaluations, which is the case for applications involving sensitivity analysis or uncertainty quantification.

4.1.4. *Scalability test.*

We report in this section a study on the strong scalability for $\mathcal{P}_2$ finite elements. We employ the grid with 365'524 nodes, leading to $N_h = 2'848'000$, and a number of processors $N_{cpu} = 64$, 128, 256, 512, 1024. In Table 5 the computational time and the number of iterations employed online by the MSRB-preconditioned FGMRES is reported. The number of iterations slightly increases by increasing the number of processors, but it never exceeds 5 iterations. As for the computational times, the results confirm ideal scalability, as it is possible to infer from Figure 4. This is due to the fact that the dimensions of the RB coarse corrections are small and do not vary by changing the number N_{cpu} of processors, which does not impact the computational time required by solving the RB problems. Consequently, the computational time t_{MSRB} is mainly devoted to the construction and the application of the fine component $\mathbf{P}_{\text{BJ}}(\boldsymbol{\mu})$, which is embarrassingly parallel.

Table 4: Grid analysis: FGMRES with $\mathcal{P}_1$ FE.

	N_h	N_{cpu}	L	N_k, $k = 0, 1, 2$	t_{MSRB} $[s]$ (It)	t_{GML} $[s]$ (It)	t_{off} $[s]$	BEP
T3	46456	12	3	22 118 303	0.07 (3)	0.13 (41)	352.32	5488
T3	365254	96	3	22 111 290	0.11 (6)	0.32 (40)	646.57	3052
T3	2887193	768	3	24 110 276	0.47 (12)	1.71 (62)	3799.56	3062
T3	22767295	6144	3	22 100 258	0.26 (13)	49.56 (81)	57493.6	1167

Table 5: Scalability analysis: FGMRES with $\mathcal{P}_2$ FE, $N_h = 2'848'000$.

	N_{cpu}	Dof/cpu	N_k	t_{MSRB} $[s]$ (It)	t_{off} $[s]$
T3	64	44500	26 139 372	4.56 (3)	32194.30
T3	128	22250	26 136 365	2.03 (3)	18208.60
T3	256	11125	26 134 361	0.83 (4)	13572.30
T3	512	5563	26 131 354	0.47 (4)	12657.70
T3	1024	2782	26 129 345	0.26 (5)	12043.90

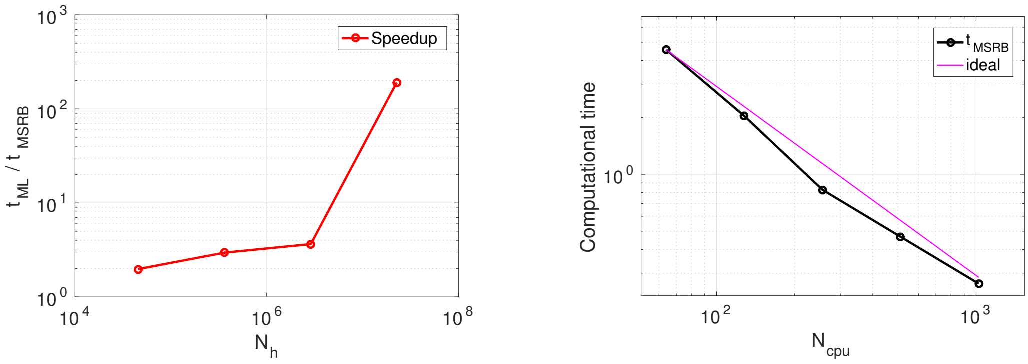

Figure 4: Speed up (left) and strong scalability (right) for online evaluations.

5. Conclusions.

In this paper we have assessed the computational performances of the MSRB preconditioner for large-scale FE advection-diffusion problems. While the performances of standard AMG preconditioners are usually affected for advection-dominated problems featuring an anisotropic behavior and *ad hoc* techniques must be employed, see e.g. [8,16]; such an impact is not observed in the case of the proposed MSRB preconditioner. Indeed, the results confirm that relying on a RB coarse correction allows to fasten the convergence of the preconditioned iterative method for all the cases investigated. This is made possible by the construction of the RB spaces which surfing the manifold of the PDE solutions in the parameter range as well as the ones generated by the errors at the successive steps. On the contrary, standard AMG preconditioners in a parametrized context would build the preconditioning matrix from scratch at any new parameter instance queried during the online stage, hence without taking into account the parametric dependence of the PDE solution, and possible correlations among PDE solutions obtained for parameter values not too far from each other. Finally, since the iteration count can be reduced to an extremely small amount, this methodology can be exploited to boost the convergence of iterative methods for other classes of parametrized problems, such as saddle-point problems, where the fine grid preconditioner cost per iteration may be particularly expensive. In this respect, suitable fine preconditioners trained for saddle-point problems can be combined with RB coarse components tailored for solving parametrized saddle-point PDEs, as shown in [19,20].

Acknowledgements.

The research of N. Dal Santo has been supported by the Swiss State Secretariat for Education, Research and Innovation (SERI), project No. C14.0068, in the framework of the COST action number TD1307. We acknowledge the Swiss National Supercomputing Centre (CSCS) for providing us the CPU resources under project ID s635.

REFERENCES

1. N. Dal Santo, S. Deparis, A. Manzoni, and A. Quarteroni, Multi space reduced basis preconditioners for large-scale parametrized PDEs, tech. rep., Mathicse 32, 2016.
2. S. Brenner and R. Scott, *The mathematical theory of finite element methods*, vol. 15. Springer Science & Business Media, 2007.

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

4. A. Quarteroni, *Numerical Models for Differential Problems*, vol. 9 of *Modeling, Simulation and Applications (MS&A).* Springer-Verlag Italia, Milano, 2nd ed., 2014.

5. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods in fluid dynamics.* Springer-Verlag, 1988.

6. Y. Saad, *Iterative methods for sparse linear systems.* SIAM, 2003.

7. A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, vol. 10. Oxford University Press, 1999.

8. A. J. Wathen, Preconditioning, *Acta Numerica*, vol. 24, pp. 329–376, 5 2015.

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

10. A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92. Springer, 2016.

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

12. O. Zahm and A. Nouy, Interpolation of inverse operators for preconditioning parameter-dependent equations, *SIAM Journal on Scientific Computing*, vol. 38, no. 2, pp. A1044–A1074, 2016.

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*, vol. 37, no. 3, pp. 1304–1336, 2016.

14. Y. Saad, A flexible inner-outer preconditioned gmres algorithm, *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 461–469, 1993.

15. M. Bebendorf, Y. Maday, and B. Stamm, Comparison of some reduced representation approximations, in *Reduced Order Methods for Modeling and Computational Reduction* (A. Quarteroni and G. Rozza, eds.), vol. 9 of *Modeling, Simulation and Applications*, pp. 67–100, Cham, Switzerland: Springer, 2014.

16. A. Ramage, A multigrid preconditioner for stabilised discretisations of advection-diffusion problems, *Journal of Computational and Applied Mathematics*, vol. 110, no. 1, pp. 187 – 203, 1999.

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

18. M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, Ml 5.0 smoothed aggregation user's guide, tech. rep., SAND2006-2649, Sandia National Laboratories, 2006.

19. N. Dal Santo, S. Deparis, A. Manzoni, and A. Quarteroni, An algebraic least squares reduced basis method for the solution of parametrized Stokes equations, tech. rep., Mathicse 21, 2017.

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