Noname manuscript No.
(will be inserted by the editor)

Reduced Basis' Acquisition by a Learning Process for Rapid On-line Approximation of Solution to PDE's : laminar flow past a backstep

Patrick Gallinari, Yvon Maday*, Maxime Sangnier, Olivier Schwander, Tommaso Taddei

Received: xxx / Accepted: date

Abstract Reduced Basis Methods for the approximation to parameter-dependent Partial Differential Equations are now well-developed and start to be used for industrial applications. The classical implementation of the Reduced Basis Method goes through two stages: in the first one, offline and time consuming, from standard approximation methods a reduced basis is constructed; then in a second stage, online and very cheap, a small problem, of the size of the reduced basis, is solved. The offline stage is a learning one from which the online stage can proceed efficiently. In this paper we propose to exploit Machine Learning procedures in both offline and online stages to either tackle different classes of problems or increase the speed-up during the online stage. The method is presented through a simple flow problem — a flow past a backward step governed by the Navier Stokes equations — which shows, however, interesting features.

This work is part of the activity of the "*Institut Carnot Smiles*" .

P. Gallinari & O. Schwander
Sorbonne Universités, UPMC Univ. Paris 06 and CNRS, UMR 7606,
LIP6, 4, Place Jussieu, 75252, Paris Cedex 05, France.
E-mail: {Patrick.Gallinari, olivier.schwander}@lip6.fr
Y. Maday
Sorbonne Universités, UPMC Univ. Paris 06 and CNRS, UMR 7598,
Laboratoire Jacques-Louis Lions, 4, Place Jussieu, 75252, Paris Cedex 05, France
Institut Universtaire de France
Division of Applied Mathematics, Brown University, Providence, RI 02912, USA
E-mail: maday@ann.jussieu.fr
M. Sangnier
Sorbonne Universités, UPMC Univ. Paris 06, CNRS, FRE 3684,
Laboratoire de Statistique Théorique et Appliquée and CNRS, UMR 7606,
LIP6, 4, Place Jussieu, 75252, Paris Cedex 05, France.
E-mail: maxime.sangnier@upmc.fr
T. Taddei
Sorbonne Universités, UPMC Univ. Paris 06 and CNRS, UMR 7598,
Laboratoire Jacques-Louis Lions, 4, Place Jussieu, 75252, Paris Cedex 05, France
E-mail: taddei@ljll.math.upmc.fr
*Corresponding author: Yvon Maday, E-mail: maday@ann.jussieu.fr

Keywords: Reduced Basis, Geometry-as-a-Parameter, Learning Process, Numerical Methods, Navier Stokes Equations

1 Introduction

Reduced Basis Methods (RBM) is a class of model order reduction techniques for the approximation of the solution to parameter-dependent problems, stated e.g. as a Partial Differential Equation (PDE), when many-query and/or real-time simulations are required. They rely and exploit the fact that the solutions, as a function of the parameter, look quite alike, and — stated in a more rigorous way — constitute a manifold with a small Kolmogorov width. Most of the times, the RBM is deployed in two stages. The first one, called "offline", is a deterministic learning process where the reduced basis is constructed by learning the similarities and the differences between the various solutions for different values of the parameter. This learning process is generally either based on a singular value decomposition of (large enough and pragmatic discrete sample of) the manifold of all solutions or a greedy procedure based on an (pragmatically accurate and reliable) error estimator. The second stage, called "online", consists in solving a (very) low-dimensional problem associated to, e.g. a Galerkin approximation of the PDE on the space spanned by the reduced basis.

These approaches allow to lower significantly the complexity for approximating — online — the solution to the PDE by exploiting the knowledge that is acquired during the offline stage. The offline stage, on its own, requires classical approximations like finite element or spectral methods in order to determine the basis functions. These are written in terms of the (not reduced) finite element or spectral basis. During the offline step, together with the construction of the RB, basic computations are done (like construction of elementary mass and stiffness reduced matrices involving computation within the finite element or spectral framework). These are then used, online, for rapid construction and inversion of the Galerkin process either by affine assumption or use of the Empirical Interpolation Techniques. The online cost is much reduced with respect to the finite element case (typically scaling like N^3 where N is the size of the reduced basis, which is much smaller than the size $\mathcal{N}$ of the underlined discretization) leading to a rapid procedure for many-query or real-time simulations. We refer to [1] [11], [12], [14] and also to the recent books [5] and [13] for more on the subject of RBM.

Note that, actually, the reduced basis framework can even go beyond this basic case by allowing reduced approximations of problems that could not be even solved by classical approaches, because they would be too large. Examples are provided by the reduced basis element method (RBEM) [10,9,8], and by the static-condensation reduced basis element method (scRBEM) [6,7,15] that combine the domain decomposition framework with the RB techniques. In these methods indeed a "large" domain is decomposed into smaller "components" that, like in a "Lego-brick" construction, allow to get together and re-constitute the domain Ω of interest. These components are all deformations of few generic shapes. These generic shapes are "filled" with reduced bases that represent local solutions to the given phenomenon that has been modeled through the PDE on various deformations of the generic shapes. Hence, when transported back to each subdomain from the domain decomposition, these reduced bases are able to approximate well the local restriction of the solution defined on the whole domain. The (offline) construction of the local solutions does not require using a large domain for the finite element (say) simulation. Indeed, a moderate-sized domain, from which

the local solutions are extracted, is generally good enough for this purpose. Once the different "generic" local reduced bases are constructed, one can assemble the discrete global space by gluing together (and the RBEM and scRBEM differ from the way the glue is performed) the local solutions obtained by mapping on each subdomain the reduced basis from the relevant generic shape. The full (say finite element) discretization method is thus never deployed on the whole domain but the reduced system is the only one to be implemented. Note that during the online stage, the domains of interest can be much larger than the ones used during the offline stage to construct the local RB.

The above strategy allows to understand that, if the solution to some PDE does depend on the geometry, this dependence behaves in a reproducing way so that the same part of the solution can be found in another configuration.

Our project here is to exploit this fact and go one step further and is rooted in the following observations :

1. expert engineers generally envision what the solution of the system they know well can be, and they can also detect incoherence in numerical solutions that are proposed to them;[1]
2. users of numerical methods are able to recognize, among various numerical representations of solutions, those that correspond e.g. to fluid flows from those corresponding to wave propagation problems;
3. machine learning methods are now routinely able to classify pictures or features representing different objects or situations;
4. machine learning methods are also able to mimic "input/output" systems.

For our project, in the framework of fluid flows, we want:

1. to reproduce — through an automatic procedure — the ability of expert engineers of qualitatively approximating the solution to a given flow problem by assembling a number of regimes that they have seen before. These flow regimes (effectively, the pieces of the "puzzle") are close to the "generic" local reduced bases that are constructed in the reduced basis element method;
2. to simplify the construction of the reduced basis method by using a variety of previously-computed approximations, by automatically deconstructing the global solutions and by automatically classifying the different regimes;
3. to automatically identify, for a new configuration, a partition of the computational domain such that each element of the partition is characterized by one of the previously-observed flow regimes;
4. to propose an approximation by exploiting a library of flows.

The project is ambitious at the global level but partial realizations are already interesting. The present paper is a first contribution in this direction, and deals with a flow problem past a backstep.

2 Description of the model problem

Let us consider the domain $\Omega = (0, L_1] \times (0, H_0) \cup (L_1, L) \times (H_0 - H_1, H_0)$ shown in Figure 1. We set $H_0 = 1$, and we consider L_1, L, H_1 as parameters of the model.

[1] note, however, that this does not imply that numerical methods are not worthy as they can lead to new solutions and also are able to be connected to optimization or control algorithms based on mathematical concepts.

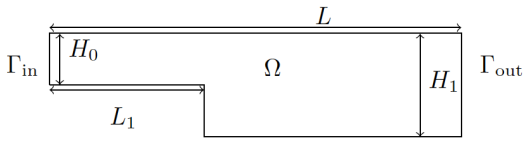


Fig. 1: Laminar flow past a backstep: computational domain.

The flow in such a domain is characterized by its velocity field $\boldsymbol{u} : \Omega \to \mathbb{R}^2$, and its pressure field $p : \Omega \to \mathbb{R}$ solution to the steady Navier Stokes equations with parabolic inflow at Γ_{in}:

$$\begin{cases} -\frac{1}{\text{Re}}\Delta \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = 0 \text{ in } \Omega, \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega \\ \boldsymbol{u}(\boldsymbol{x}) = \begin{bmatrix} 4\frac{x_2}{H_0}(1-\frac{x_2}{H_0}) \\ 0 \end{bmatrix} & \text{on } \Gamma_{\text{in}} \\ \frac{1}{\text{Re}}\partial_{\boldsymbol{n}}\boldsymbol{u} - p\boldsymbol{n} = 0 & \text{on } \Gamma_{\text{out}} \\ \boldsymbol{u} = 0 & \text{on } \partial\Omega \setminus \Gamma_{\text{out}} \end{cases} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2) \in \Omega$. We observe that the solution pair $(\boldsymbol{u}, p)$ is uniquely determined by the vector of parameters

$$\mu = [\text{Re}, L_1, H_1, L] \in \mathcal{P}, \tag{2}$$

where $\mathcal{P}$ denotes the parameter domain, set here to $\mathcal{P} = [30, 100] \times [3, 6] \times [1.5, 2] \times [10, 15]$. To stress this dependence, we adopt notation $(\boldsymbol{u}(.;\mu), p(.;\mu))$ to indicate the solution to (1) for a given $\mu \in \mathcal{P}$. Similarly, we use notation $\Omega = \Omega(\mu)$ to indicate the domain associated to the parameter $\mu \in \mathcal{P}$.

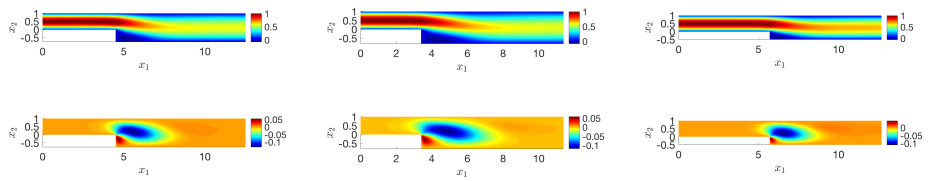
Figure 2 shows the first and the second components of the velocity field[2] for three values of the parameter $\mu \in \mathcal{P}$. We observe that the three solutions considered share significant features among each other, and more in general we observe the same qualitative physical behavior (nearly-parabolic flow, followed by a recirculating flow, followed by another nearly-parabolic flow) for a significant range of parameters $\mathcal{P}$.

The fact that these Figures look alike comes from the continuous dependence of the solutions with respect to a (small at least) variation of the geometric or flow parameters in $\mathcal{P}$ and implies a possible reduction from reduced basis method. As said in the introduction, we consider the manifold of all (velocities) solutions to problem (1) when the parameters vary.

Actually, as is the case in other situations where some knowledge on the flow can be used to diminish further the Kolmogorov dimension of the manifold of solutions (see e.g. ([2]) for hyperbolic flows with solutions with moving discontinuities), we are going to exploit the presence of a recirculating zone with varying length.

Let us make this more precise : we can separate the flow into three separate regions, or components similarly to RBEM or scRBEM, define two distinct flow regimes.

[2] From now on, we restrict our attention to the velocity field. The same discussion can be extended to the pressure field.



(a) μ^1 (b) μ^2 (c) μ^3

Fig. 2: Flow past a backstep: solution for three values of $\mu \in \mathcal{P}$: $\mu^1 = [65, 4.5, 1.75, 12.5]$, $\mu^2 = [87, 3.4, 1.8, 11.4]$, $\mu^3 = [93, 5.7, 1.5, 12.7]$.

– **Entrance:** before the step, the flow is roughly approximated by the Poiseuille flow, almost independent of x_1

$$u_1(\mathbf{x}) = \frac{4x_2}{H_0}\left(1 - \frac{x_2}{H_0}\right), \quad u_2(\mathbf{x}) \equiv 0.$$

– **Region close to the step:** the flow is here characterised by a laminar oblique flow on top, and by a recirculating flow on the bottom.
– **Exit:** after the recirculating zone, the flow is again roughly approximated by the Poiseuille solution:

$$u_1(\mathbf{x}) = 4\frac{x_2 + H_1 - H_0}{H_1}\left(1 - \frac{x_2 + H_1 - H_0}{H_1}\right), \quad u_2(\mathbf{x}) \equiv 0.$$

From a mathematical standpoint, we split the domain Ω in three subregions, $\Omega_1, \Omega_2, \Omega_3$ corresponding to the two regimes identified before; then, we define a local manifold for each subregion. In the remainder of the paper, we describe (i) how to practically identify the subregions, and (ii) how to determine the coefficients of the expansion during the online stage. In order to simplify the geometric transformation, the interface between the two last regions will be chosen perpendicular to the channel, at a proper abscissa $\xi(\mu)$ that will be defined later.

3 Construction of the solution manifolds

3.1 Decomposition of the solutions into Poiseuille component and recirculating one

In order to clarify the procedure, let us further discuss the application of the proposed algorithm to the particular model problem at hand. We state upfront that the extension of our approach to a larger class of problems requires some more investigation.

Motivated by the previous discussion, we wish to consider the global solution manifold as the combination of (here) two distinct local manifolds corresponding to the three regions (entrance, step, exit) of the flow taking into account that there are two flow regimes (Poiseuille flow and recirculating flow).

By the physical argument exposed before, we first choose

$$\Omega_1(\mu) = (0, L_1) \times (0, H_0), \qquad \Omega_2(\mu) = (L_1, L_1 + \xi(\mu)) \times (H_0 - H_1, H_0),$$
$$\Omega_3(\mu) = (L_1 + \xi(\mu), L) \times (H_0 - H_1, H_0), \tag{3}$$

and we characterize the flow regime in each of the subdomains $\Omega_k(\mu)$ by labeling them with an index $\ell_k(\mu)$. Here we have the choice between two flow regimes : $\ell_k(\mu) = 1$ for Poiseuille flow and $\ell_k(\mu) = 2$ for recirculating flow . Hence we set $\ell_1(\mu) = \ell_3(\mu) = 1$ and $\ell_2(\mu) = 2$. We observe that $\Omega_2(\mu)$ and $\Omega_3(\mu)$ depend on the quantity $\xi : \mathcal{P} \to (0, L-L_1)$ introduced in the previous section that is for the moment only heuristically stated; we discuss later how to choose it.

We then notice that the three regions are rectangles thus easily mapped to the unit square $\widehat{\mathcal{C}} = (0,1)^2$. We thus introduce the affine maps

$$T_k : \widehat{\mathcal{C}} \times \mathcal{P} \to \Omega_k(\mu). \tag{4}$$

$T_k(\hat{\boldsymbol{x}}; \mu) = \mathbb{B}_k(\mu)\hat{\boldsymbol{x}} + b_k(\mu)$, more precisely

$$T_1(\hat{\boldsymbol{x}}; \mu) = \begin{bmatrix} L_1 \hat{x}_1 \\ H_0 \hat{x}_2 \end{bmatrix}, \qquad T_2(\hat{\boldsymbol{x}}; \mu) = \begin{bmatrix} L_1 + \xi(\mu)\hat{x}_1 \\ H_1 \hat{x}_2 + H_0 - H_1 \end{bmatrix}, \tag{5}$$
$$T_3(\hat{\boldsymbol{x}}; \mu) = \begin{bmatrix} L - (L - (L_1 + \xi(\mu)))\hat{x}_1 \\ -H_1 \hat{x}_2 + (H_1 - H_0) \end{bmatrix}.$$

We observe that the choice of T_3 is motivated by a symmetry argument so that the sides corresponding to the true boundaries of $\Omega_1(\mu)$ and $\Omega_3(\mu)$ get mapped to the same side of $\widehat{\mathcal{C}}$.

Now, given the velocity field $\boldsymbol{u}(\mu) : \Omega(\mu) \to \mathbb{R}^2$, and the linear maps $T_k(\mu) : \widehat{\mathcal{C}} \to \Omega_k(\mu)$, $k = 1, 2, 3$, we employ a Piola transformation to map the velocity to the reference configuration in a convenient way[3]:

$$\boldsymbol{u}_k^{\text{ref}}(\hat{\boldsymbol{x}}; \mu) = |\det(\mathbb{B}(\mu))| \mathbb{B}_k(\mu)^{-1} \boldsymbol{u}|_{\Omega_k(\mu)}(T_k(\hat{\boldsymbol{x}}; \mu); \mu). \tag{6}$$

In section 4.4, we provide further details about the practical construction of $\boldsymbol{u}_k^{\text{ref}}$ for finite element/spectral discretizations.

All these pieces are now defined on a unique reference domain $\widehat{\mathcal{C}}$ and we can build the two manifolds :

– manifold of "Poiseuille-like" flows

$$\mathcal{Z}^{(1)} = \text{span}\{\boldsymbol{u}_1^{\text{ref}}(.; \mu), \boldsymbol{u}_3^{\text{ref}}(.; \mu), \quad \mu \in \mathcal{P}\}; \tag{7}$$

– manifold of "recirculation" flows

$$\mathcal{Z}^{(2)} = \text{span}\{\boldsymbol{u}_2^{\text{ref}}(.; \mu), \quad \mu \in \mathcal{P}\}. \tag{8}$$

With these notations, we can now explain how to define properly the map $\xi : \mathcal{P} \to (0, L - L_1)$: it is chosen to maximise the averaged correlation between (i) the restriction of $\boldsymbol{u}(x; \mu)$ to $\Omega_2(\mu)$ and $\boldsymbol{u}_2^{\text{ref}}([T_2(\mu)]^{-1}\boldsymbol{x}; \mu^{\text{ref}})$, and (ii) the restriction of $\boldsymbol{u}(\boldsymbol{x}; \mu)$ to $\Omega_3(\mu)$ and $\boldsymbol{u}_1^{\text{ref}}([T_3(\mu)]^{-1}\boldsymbol{x}; \mu^{\text{ref}})$. Here μ^{ref} is the centroid of $\mathcal{P}$, and we choose $\xi(\mu^{\text{ref}}) = 3$ by inspection. In formulas, $\xi(\mu)$ is given by

$$\xi(\mu) = \arg \max_{\xi \in (0, L-L_1)} \rho\left(\mathbf{u}_2^{\text{ref}}(\cdot; \mu^{\text{ref}}, \xi^{\text{ref}}), \mathbf{u}_2^{\text{ref}}(\cdot; \mu, \xi)\right) + \rho\left(\mathbf{u}_3^{\text{ref}}(\cdot; \mu^{\text{ref}}, \xi^{\text{ref}}), \mathbf{u}_3^{\text{ref}}(\cdot; \mu, \xi)\right) \tag{9a}$$

where the correlation coefficient $\rho : L^2(\widehat{C}) \times L^2(\widehat{C}) \to [-1, 1]$ is given by the following formula:

$$\rho(\mathbf{w}, \mathbf{v}) = \frac{(\mathbf{w}, \mathbf{v})_{L^2(\widehat{C})}}{\|\mathbf{w}\|_{L^2(\widehat{C})} \|\mathbf{v}\|_{L^2(\widehat{C})}}. \tag{9b}$$

[3] Remember that this Piola transformation preserves the incompressibility condition satisfied by the velocity field.

3.2 Construction of the reduced velocity spaces

The two manifolds $\mathcal{Z}^{(1)}$ and $\mathcal{Z}^{(2)}$ being constructed — theoretically at least — we need now to extract the reduced bases that approximate them well. A classical approach is to use a Proper Orthogonal Decomposition (POD) of those over a sufficiently large sample. We thus design a sufficiently large training set

$$\mathcal{P}_{\text{train}} = \{\mu^p \in \mathcal{P}, p = 1, \dots, n_{\text{train}}\}, \tag{10}$$

where $\mu^1, \dots, \mu^{n_{\text{train}}}$ are randomly sampled from an uniform distribution over $\mathcal{P}$. This allows to define the training spaces $\mathcal{Z}^{(1)}_{\text{train}}$ and $\mathcal{Z}^{(2)}_{\text{train}}$ as follows

$$\mathcal{Z}^{(1)}_{\text{train}} = \text{span}\{\boldsymbol{u}_1^{\text{ref}}(.;\mu^p), \boldsymbol{u}_3^{\text{ref}}(.;\mu^p), \quad p = 1, \dots, n_{\text{train}}\} \tag{11}$$

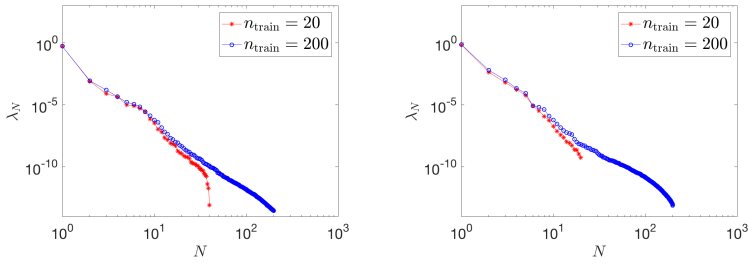
with dimension $2n_{\text{train}}$ and

$$\mathcal{Z}^{(2)}_{\text{train}} = \text{span}\{\boldsymbol{u}_2^{\text{ref}}(.;\mu^p), \quad p = 1, \dots, n_{\text{train}}\} \tag{12}$$

with dimension n_{train}.

We then perform a standard POD on the basis of these two spaces based on the L^2 – energy norm (i.e., $\|\boldsymbol{u}\|^2 := \int_\Omega \sum_{i=1}^2 u_i^2$) and consider the associated eigenmodes.

An indicator that the reduced approximation is consistent is that the eigenvalues (when ranked in a decreasing order) converge to zero pretty fast : this is what we notice indeed , look at Figure 3 where the eigenvalues are plotted for both spaces, computed out of $n_{\text{train}} = 20$ and $n_{\text{train}} = 200$. We notice the fast decay and the fact that, even for $n_{\text{train}} = 20$, we are able to estimate the first $N = 6, 7$ eigenvalues extremely accurately.



(a) $\mathcal{Z}^{(1)}_{\text{train}}$ (b) $\mathcal{Z}^{(2)}_{\text{train}}$

Fig. 3: Flow past a backstep: behavior of the POD modes for two datasets.

The reduced spaces $\mathcal{Z}^{(\ell)}_{N_\ell}$ ($\ell = 1, 2$) are then constructed by retaining the N_ℓ most energetic modes $\boldsymbol{\zeta}_n^{(\ell)}$ associated to the largest eigenvalues extracted from $\mathcal{Z}^{(\ell)}_{\text{train}}$, they produce the reference local reduced spaces for the velocity fields:

$$\mathcal{Z}^{(\ell)}_{N_\ell} = \text{span}\{\boldsymbol{\zeta}_n^{(\ell)} : \widehat{\mathcal{C}} \to \mathbb{R}^2 : \ n = 1, \dots, N_\ell\}, \quad \ell = 1, 2. \tag{13}$$

The spaces $\{\mathcal{Z}_{N_\ell}^{(\ell)}\}_{\ell=1}^2$ will then allow to approximate, through the maps (5), the flow in the three regions of the domains corresponding to the two flow regimes. Remember that they are defined over the reference configuration $\widehat{\mathcal{C}}$ and that $\mathcal{Z}_{N_1}^{(1)}$ corresponds to the "Poiseuille flow", while $\mathcal{Z}_{N_2}^{(2)}$ corresponds to the "recirculating flow".

We can thus expect to approximate for any $\mu \in \mathcal{P}$ the velocity as follows

$$\boldsymbol{u}(\boldsymbol{x};\mu) \approx \boldsymbol{u}_N(\boldsymbol{x};\mu) := \sum_{n=1}^{N_\ell} u_{n,k}(\mu)\, P_k\left(\boldsymbol{\zeta}_n^{(\ell)}\right)(\mathbf{x}), \quad \boldsymbol{x} \in \Omega_k(\mu); \tag{14}$$

where $P_k(\mathbf{w})(\mathbf{x}) = \frac{1}{\det(\mathbb{B}(\mu))}\mathbf{w}\left([T_k(\mu)]^{-1}(\mathbf{x})\right)$ is the Piola transformation for the map $T_k(\hat{\mathbf{x}};\mu) = \mathbb{B}(\mu)\hat{\mathbf{x}}+b(\mu)$. Like in any reduced basis method, the coefficients $\{u_{n,k}(\mu)\}_{n,k}$ must be estimated by resorting to some technique: for instance by projection based on the mathematical model (1) or on a simplified version of it. Another difficulty though appears as all the construction above is based on the definition of the quantity ξ associated with the geometric transformation: recalling (9), the computation of ξ requires the computation of the solution $\mathbf{u}(\mu)$. Assuming that we know the optimal coefficients $\{u_{n,k}(\mu)\}_{n,k}$ and the mapping parameter $\xi(\mu)$, we can appreciate the accuracy of the best approximation (14), Figures (6 (a)–(e)) below show that the mean relative $L^2(\Omega)$ error decreases rapidly with N as it should, and is about 10^{-3} for $N = 10$.

3.3 Algorithm for the Reduced Basis constructions

Algorithm 1 First offline stage : Construction of the Reduced Basis

Given the offline solutions $\{\boldsymbol{u}(\mu^p)\}_{p=1}^{n_{\text{train}}}$, for randomly-sampled parameters μ^p from an uniform distribution over $\mathcal{P}$

1: set μ^{ref} as the centroid of $\mathcal{P}$, set $\xi(\mu^{\text{ref}})$ by inspection (here $= 3$), and build the $K = 3$ reference solutions $\boldsymbol{u}_k^{\text{ref}}(.;\mu^{\text{ref}})$, $k = 1, 2, 3$ from (6).
2: For each $\mu^p \in \mathcal{P}_{\text{train}}$ determine $\xi(\mu^p)$, by computing (9).
3: For each $\mu^p \in \mathcal{P}_{\text{train}}$ determine the partition $\{\Omega_k(\mu^p)\}_k$ (3), and the maps $\{T_k(\cdot;\mu^p)\}_k$ (5), and consider the $L \equiv 2$ libraries of flow regimes $\{u_1^{\text{ref},}(\mu^p)\}_{p=1}^{2n_{\text{train}}}$ and $\{u_2^{\text{ref}}(\mu^p)\}_{p=1}^{n_{\text{train}}}$,
4: apply POD based on the $L^2(\widehat{\mathcal{C}})$ – energy norm to each library to obtain the reduced spaces $\mathcal{Z}_{N_1}^{(1)}$ and $\mathcal{Z}_{N_2}^{(2)}$.

The difficulty now is to derive a workable evaluation of the map ξ and of the coefficients $\{u_{n,k}(\mu)\}_{n,k}$ that does not require (as we did in the above test reported in Figures (6 (a)–(e)) to compute the solution $\boldsymbol{u}(\mu)$ before approximating it by its reduced basis representation (14).

We present in the next section a statistical learning procedure to evaluate both ξ and the coefficients $u_{n,k}(\mu)$.

4 Statistical learning approach for ξ and more

4.1 Automatic partitioning of the region after the step

Given the dataset $\{(\mu^p, \xi(\mu^p))\}_{p=1}^{n_{\text{train}}}$ generated applying Algorithm 1, we fit a regression tree[4] (using the Matlab function `fitrtree`) $\xi^{\text{learned}} : \mathcal{P} \to \mathbb{R}$ that approximates the offline map $\xi : \mathcal{P} \to \mathbb{R}$ (9). Decision trees is a class of statistical learning methods used both in classification and regression. The last situation is the one that meets our expectations since we aim at estimating a real-valued (in opposition to binary-valued) function $\xi : \mathcal{P} \to \mathbb{R}$. In a nutshell, a regression tree is an iterative algorithm that fits a piecewise constant function to the data.The constant cells are determined in the space of parameters $\mathcal{P}$ in order that they are as homogeneous as possible regarding the value to predict [4]. Let us note that the cells are rectangular since, at each iteration, a regression tree chooses a cell, along with a single variable to split, so that homogeneity is maximized on each side. In our implementation, a cell cannot be split if there are less than 10 training points inside.

In order to assess the accuracy and determine whether or not to collect new data (i.e., run new offline simulations), we resort to a cross-validation procedure. We randomly partition the offline dataset into a subset of size $0.9n_{\text{train}}$ and a subset of size $0.1n_{\text{train}}$. The former is used to learn the regressor, the latter is used to measure performance. In this work, we compute the coefficient of determination R^2:

$$R^2(\xi) = 1 - \frac{\sum_{p=1}^{n_{\text{test}}} (\xi^{\text{learned}}(\tilde{\mu}^p)) - \xi(\tilde{\mu}^p))^2}{\sum_{p=1}^{n_{\text{test}}} (\bar{\xi}_{\text{train}} - \xi(\tilde{\mu}^p))^2}, \qquad \bar{\xi}_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{p=1}^{n_{\text{train}}} \xi(\mu^p).$$

The coefficient of determination measures the proportion of the variance in the dependent variable that is predictable from the independent variables: a value close to zero means that the regressor is highly-inaccurate and/or the independent variables are not informative about the quantity we wish to predict, while a value close to one proves that the regressor is accurate. We consider B independent random splits in order to reduce the effect of the particular split considered. Figure 4 shows the histogram of R^2 for $B = 2000$ and $n_{\text{train}} = 200$. The averaged value of the R^2 is equal to 0.93.

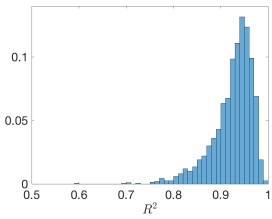


Fig. 4: Flow past a backstep: histogram of the out-of-sample coefficient of determination (R^2) based on B 0.9-0.1 random split of the dataset ($B = 2000$, $n_{\text{train}} = 200$). The averaged R^2 is equal to 0.93.

[4] Note that in this paper regression trees have been chosen because it is a classical regressor, but any other regressor could have been used.

Another approach — which is not computationally practical — to measure the performance of the regressor is to consider the behavior of the exact projection on the reduced basis space defined from the use of ξ^{learned}. We can check in Figures (6 (b)–(f)) below that we need many more samples for the estimate of the mapping parameter ξ than for the estimate of the first POD eigenvalues, as we are only able to get fair approximations with $n_{\text{train}} = 200$.

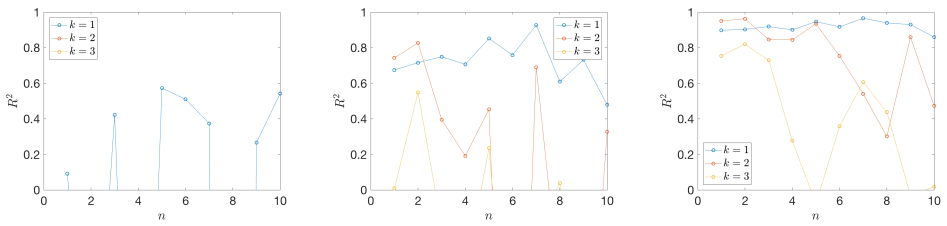
4.2 Prediction of expansion coefficients

We define the optimal expansion coefficients $\{u_{n,k}(\mu)\}_n$ as follows:

$$u_{n,k}(\mu) = \left(\boldsymbol{\zeta}_n^{\ell_k}, \mathbf{u}_k^{\text{ref}}(\cdot;\mu)\right)_{L^2(\widehat{\mathcal{C}})}, \quad n = 1, \ldots, N, \quad k = 1, 2, 3. \tag{15}$$

Then, we apply the same procedure to learn and predict the coefficients $\{u_{n,k}\}_n$ (for each region indexed by k) in the RB expansion (14) for new values of μ. As for the regressor associated with ξ, also in this case, we can compute the coefficient of determination $R^2(u_{n,k})$.

As illustrated in Figure 5, we realize also that the estimate of the coefficients $u_{n,k}(\mu)$ requires many more samples than the estimate of the first POD eigenvalues. For $n_{\text{train}} = 200$, we are able to estimate well only 3 coefficients for the third region. There is a practical explanation of why the coefficients in the first region are easier to learn: the choice of ξ does not influence the first region while it certainly influences the second and the third regions. Therefore, the estimate of the coefficients in the second and third regions faces two sources of uncertainty; (i) the intrinsic uncertainty dictated by variations with μ of the expansion coefficients $\{u_{n,k}(\mu)\}_n$, and (ii) the uncertainty dictated by the errors associated with the estimate of the mapping parameter ξ.



(a) out-of-sample, $n_{\text{train}} = 20$ (b) out-of-sample, $n_{\text{train}} = 50$ (c) out-of-sample, $n_{\text{train}} = 200$

Fig. 5: Flow past a backstep: behavior of the coefficient of determination (R^2) for each regressor. Out-of-sample R^2 is computed based on $B = 200$ $0.9 - 0.1$ random train/test partitions.

4.3 Full learning algorithm

Let us summarize the procedure through Algorithm 2.

Algorithm 2 Second offline stage : Learning process

Given the offline solutions $\{\boldsymbol{u}(\mu^p)\}_{p=1}^{n_{\text{train}}}$,

1: compute $\{\xi(\mu^p)\}_p$ and $\{u_{n,k}(\mu^p)\}_{n,k,p}$, $n = 1, \dots, N$, $k = 1, 2, 3$, $p = 1, \dots, n_{\text{train}}$ using (9) and (15);
2: learn the maps ξ^{learned} for ξ, and $\{u_{n,k}^{\text{learned}}\}_{n,k}$ for $\{u_{n,k}\}_{n,k}$ using regression trees.

4.4 From the finite element solutions over $\Omega(\mu)$ to the Learned Reduced Basis

The solution to the Navier Stokes equations (1) in the physical domain $\Omega(\mu)$, $\mu \in \mathcal{P}$ is obtained using a classical Taylor-Hood P3-P2 continuous Finite Element solver. The meshes for different values of the parameters have been generated independently. The number of degrees of freedom associated with the FE problem is in the order of $2 \cdot 10^5$ for all values of μ considered.

We emphasize that for this problem it is certainly possible to use the same reference mesh for all domains $\{\Omega(\mu) : \mu \in \mathcal{P}\}$ by introducing a suitable mapping. However, we envision that the definition of a single reference mesh is only feasible for relatively-simple parametrizations of the domain. Furthermore, even in our case, we should have set the dependence of the mapping parameter ξ on μ *a priori*, rather than learning it using simulations: *a priori* choices of ξ would lead to larger Kolmogorov widths, an are thus unfavorable for the subsequent reduction.

It is convenient over $\widehat{\mathcal{C}}$ to use a polynomial representation that we choose by using a 30 by 30 tensor product approximation based on Legendre polynomials which gives $\widehat{\mathcal{N}} = 2 \times 900$ degrees of freedom

$$\widehat{V} = \left[\text{span}\{\mathcal{L}_n(\mathbf{x}) = \mathcal{L}_{n_1}(x_1)\mathcal{L}_{n_2}(x_2) : n_1, n_2 = 1, \dots, \sqrt{\widehat{\mathcal{N}}}\}\right]^2,$$

Given the velocity field $\mathbf{u}(\mu) : \Omega(\mu) \to \mathbb{R}^2$, and the linear maps $T_k : \widehat{\mathcal{C}} \times \mathcal{P} \to \Omega_k(\mu)$ $T_k(\hat{\mathbf{x}}; \mu) = \mathbb{B}_k(\mu)\hat{\mathbf{x}} + b_k(\mu)$, we employ the Piola transformation (6) to map the velocity to the reference configuration. Since we consider inconsistent approximation spaces between the reference domain and the physical domain $\Omega(\mu)$, the definition of $\mathbf{u}_k^{\text{ref}}$ requires some care. Here, we define $\mathbf{u}_k^{\text{ref}}$ as follows:

$$\boldsymbol{u}_k^{\text{ref}}(\hat{\boldsymbol{x}}; \mu) = \mathit{\Pi}_{\widehat{V}} \left[|\det(\mathbb{B}(\mu))| \mathbb{B}_k(\mu)^{-1} \boldsymbol{u}|_{\Omega_k(\mu)}(T_k(\hat{\boldsymbol{x}}; \mu); \mu) \right], \tag{16}$$

where $\mathit{\Pi}_{\widehat{V}}$ is the L^2 – projection operator over $\widehat{V}$ (which is a obtained through a simple matrix multiplication).

Finally, the mapping parameter ξ (9) is approximated as

$$\xi_{\text{train}} = \arg \max_{\xi \in \Xi_{\text{train}}} \rho\left(\mathbf{u}_2^{\text{ref}}(\cdot; \mu^{\text{ref}}, \xi^{\text{ref}}), \mathbf{u}_2^{\text{ref}}(\cdot; \mu, \xi)\right) + \rho\left(\mathbf{u}_3^{\text{ref}}(\cdot; \mu^{\text{ref}}, \xi^{\text{ref}}), \mathbf{u}_3^{\text{ref}}(\cdot; \mu, \xi)\right) \tag{17}$$

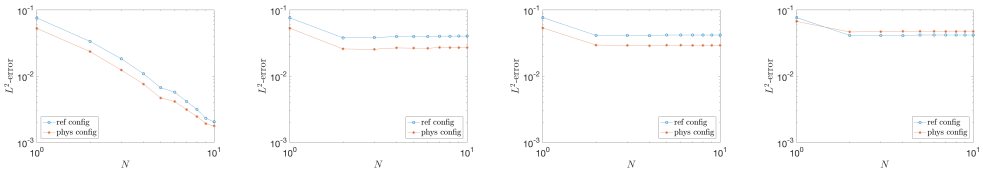
where $\Xi_{\text{train}} \subset (0, L - L_1)$, $|\Xi_{\text{train}}| = 15$ is a set of candidate values for ξ.

5 Numerical results

Figure 6 shows the behavior of the mean relative L^2 error in the reference and in the physical configurations. We consider four different cases corresponding to two

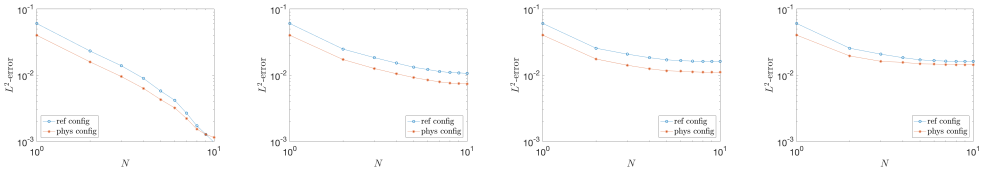
choices for the expansion coefficients and for the mapping parameter ξ: the coefficients $\{u_{n,k}(\mu)\}_{n,k}$ are computed either by projection (15) (*opt.* $u_{n,k}$) or by resorting to the learning process (*learned* $u_{n,k}$); similarly, the parameter ξ is computed either by applying (9) (*opt.* ξ) or by the learning process (*learned* ξ). We have already commented the two first columns: we observe that going from $n_{\text{train}} = 20$ to $n_{\text{train}} = 200$ is not particularly beneficial for the projection error, while it is more relevant for the error in the estimated field. This is consistent with the results of Figure 3 and 5.

$n_{\text{train}} = 20$



(a) opt. $u_{n,k}$, ξ (b) opt. $u_{n,k}$, learned ξ (c) learned $u_{n,k}$, opt. ξ (d) learned $u_{n,k}$, ξ

$n_{\text{train}} = 200$



(e) opt. $u_{n,k}$, ξ (f) opt. $u_{n,k}$, learned ξ (g) learned $u_{n,k}$, opt. ξ (h) learned $u_{n,k}$, ξ

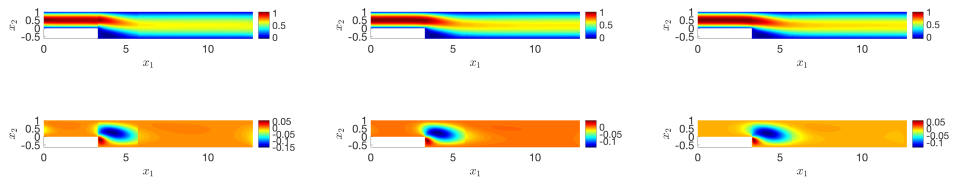
Fig. 6: Flow past a backstep: behavior of the mean relative L^2 error for $n_{\text{test}} = 10$ datapoints for optimal and learned expansion coefficients $\{u_{n,k}\}$ and mapping parameter ξ, for two values of n_{train}.

Figure 7 shows the behavior of the predicted field for three values of N, for $n_{\text{train}} = 200$. We observe convergence (in the eyeball norm) to the FE solution. We further observe that, particularly for small values of N, artifacts clearly appear at the interface between the subdomains. This can be explained by recalling that our learning procedure does not impose continuity between components at the interfaces.

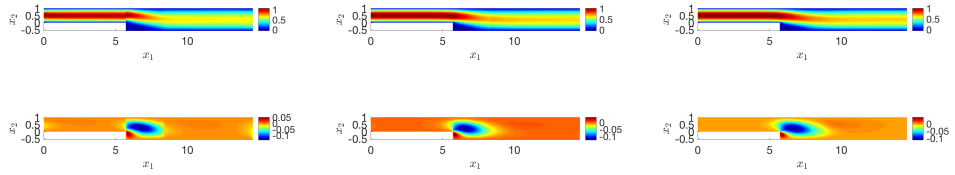
Next, Figures 8 and 9, obtained using $N = 10$ reduced basis modes, show that the estimated field (in the eyeball norm) is close to the one predicted using the FE method. Again the largest errors are at the interface between the different domains, and at the tip of the backstep where the gradient of the velocity is large.

Then, Figure 10 shows the behavior of the relative mean L^2 error with n_{train} for $N = 2$ and $N = 4$. We generate a dataset based on $n_{\text{tot}} = 250$ offline solves, then for several values of n_{train} we consider $B = 10$ random n_{train} - n_{test} train/test splits ($n_{\text{train}} + n_{\text{test}} = n_{\text{tot}}$): the training set is used to learn the model, while the test set is used to assess performance. The relative error is computed as follows:

$$\frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{n_{\text{test}}}\sum_{j=1}^{n_{\text{test}}}\frac{\|u(\mu^{b,j}) - \hat{u}(\mu^{b,j})\|_{L^2(\Omega)}}{\|u(\mu^{b,j})\|_{L^2(\Omega)}}\right)$$



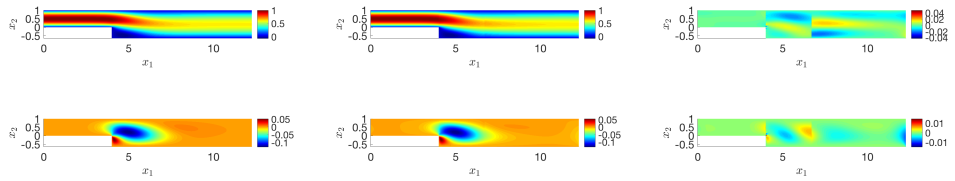
(a) $\mu = \mu^1$, $N = 1$ (b) $\mu = \mu^1$, $N = 2$ (c) $\mu = \mu^1$, $N = 10$



(d) $\mu = \mu^2$, $N = 1$ (e) $\mu = \mu^2$, $N = 2$ (f) $\mu = \mu^2$, $N = 10$

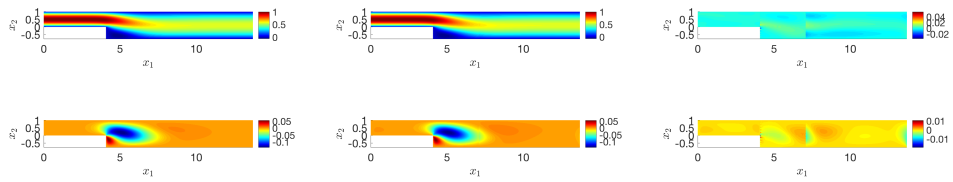
Fig. 7: Flow past a backstep: visualisation of the predicted field for three values of N and $n_{\text{train}} = 200$. $\mu^1 = [87.0307, 5.7174, 1.5635, 14.5669]$, $\mu^2 = [74.2651, 3.2926, 1.6392, 12.7344]$.

Here, $b = 1, \ldots, B$ is the index associated with the training/test split, while $j = 1, \ldots, n_{\text{test}}$. $\mu^{b,j}$ is the j-th value of the parameter in the test set associated with the b-th split. We observe that the error stagnates to roughly $2.5 \cdot 10^{-2}$ for $N = 2$ and $1.5 \cdot 10^{-2}$ for $N = 4$: this can be explained by observing that in this test we consider a fixed value of N.



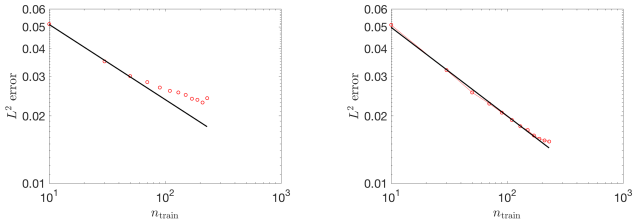
(a) u (b) $\hat{u}$ (c) $u - \hat{u}$

Fig. 8: Flow past a backstep: visualisation of the true and predicted fields for $\mu = [71.82, 4, 1.65, 12.26]$ ($n_{\text{train}} = 200$), here $N = 10$.



(a) u (b) $\hat{u}$ (c) $u - \hat{u}$

Fig. 9: Flow past a backstep: visualisation of the true and predicted fields for $\mu = [59.59, 4.08, 1.78, 13.71]$ ($n_{\text{train}} = 200$), here $N = 10$.



(a) $N = 2$ (b) $N = 4$

Fig. 10: Flow past a backstep: behavior of the relative mean L^2 error with n_{train} for $N = 2$ and $N = 4$. The black trend lines correspond to the curves $y(n) = 0.111n^{-0.335}$ for $N = 2$, and $y(n) = 0.124n^{-0.396}$ for $N = 4$.

6 Conclusion and challenges

We have presented in this paper a first attempt to employ Machine Learning techniques during the offline and online stages of the Reduced Basis method. We wish to employ Machine Learning procedures

- to automatically recognize in a given solution to a fluid problem specific flow regimes (like the Poiseuille flow and recirculation) by using classifiers, possibly informed by experts' knowledge;
- to build, for each such identified regimes, a Reduced Basis of each flow regime;
- to learn how to define the partition $\{\Omega_k(\mu)\}_{k=1}^K$ of the domain $\Omega(\mu)$ in such a way each subdomain reflects a specific flow regime;
- to learn (or at least compute from a Reduced Basis Galerkin representation of the problem) the coefficients of the solution to be determined in its proper RB.

This approach is certainly not working for problems where the classical Reduced Basis is not appropriate, i.e. for turbulent flows (see however [3] for a recent contribution in this direction), but there are many applications of interest in Mechanics, where this method could be developed.

Note that, in the current framework, we assume that the reference flow regimes are known *a priori*. We believe that this is reasonable for almost any practical application. However, it would be certainly important to be able to detect new regimes and properly augment the library of examples.

Note also that, in this (very) simple example, the dimensionality of μ is moderate. This may not be the case for other applications: we have indeed that geometry is typically described by a collection of edges (or triangles in 3D) that describe the surface. Although in certain cases it is possible to parametrize the geometry in terms of a modest amount of parameters, we envision that the procedure should be designed to handle high-dimensional inputs.

In order to proceed, we need to propose an efficient way to automatically segment a set of velocity flows $\{u(\mu^p)\}_p$ (considered as images) following some patterns $\{\mathbf{u}_k^{\text{ref}}\}_k$ corresponding to flow regimes. This will be achieved thanks to convolutional architectures.

In the problem at hand, we expect our learning procedure to automatically discover the regions $\{\Omega_1(\mu^p)\}_p$, $\{\Omega_2(\mu^p)\}_p$ and $\{\Omega_3(\mu^p)\}_p$, as well as to recognize that $\{\Omega_1(\mu^p)\}_p$ and $\{\Omega_3(\mu^p)\}_p$ correspond to the same physical regime.

For each regime indexed by k, we shall learn the map $T_k : \widehat{\mathcal{C}} \times \{\mu^p\}_p \to \cup_p \Omega_k(\mu^p)$. Note that for relatively-simple configurations, these maps can be determined (at least in part) based on prior knowledge, and may consist in a simple dilation according to the axes x_1 and x_2. In a more complicated situation, the maps $\{T_k\}_k$ may also include rotations and other transformations. In the problem at hand, estimating the maps $\{T_k\}_k$ boils down to learning the cutting point ξ as a function of μ.

References

1. Bo Östen Almroth, Perry Stern, and Frank A Brogan. Automatic choice of global shape functions in structural analysis. *Aiaa Journal*, 16(5):525–528, 1978.
2. Nicolas Cagniart, Yvon Maday, and Benjamin Stamm. Model Order Reduction for Problems with large Convection Effects. working paper or preprint, October 2016.
3. Lambert Fick, Yvon Maday, Anthony T Patera, and Tommaso Taddei. A Reduced Basis Technique for Long-Time Unsteady Turbulent Flows. working paper or preprint, 2017.
4. T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.
5. Jan S Hesthaven, Gianluigi Rozza, and Benjamin Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics. Springer*, 2016.
6. Dinh Bao Phuong Huynh, David J Knezevic, and Anthony T Patera. A static condensation reduced basis element method: approximation and a posteriori error estimation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.
7. Dinh Bao Phuong Huynh, David J Knezevic, and Anthony T Patera. A static condensation reduced basis element method: Complex problems. *Computer Methods in Applied Mechanics and Engineering*, 259:197–216, 2013.
8. Alf Emil Løvgren, Yvon Maday, and Einar M Rønquist. A reduced basis element method for complex flow systems. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.
9. Alf Emil Løvgren, Yvon Maday, and Einar M Rønquist. A reduced basis element method for the steady stokes problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 40(3):529–552, 2006.
10. Yvon Maday and Einar M Ronquist. The reduced basis element method: application to a thermal fin problem. *SIAM Journal on Scientific Computing*, 26(1):240–258, 2004.
11. Ahmed K Noor and Jeanne M Peters. Reduced basis technique for nonlinear analysis of structures. *Aiaa journal*, 18(4):455–462, 1980.
12. Christophe Prud'homme, Dimitrios V. Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T Patera, and G Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Eng*, 124(1):70–80, 2002.

13. Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.
14. Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229, 2008.
15. Kathrin Smetana and Anthony T Patera. Optimal local approximation spaces for component-based static condensation procedures. *SIAM Journal on Scientific Computing*, 38(5):A3318–A3356, 2016.