Reduced-basis solutions of affinely-parametrized linear partial differential equations

Alexandre Janon

Reduced-basis solutions of affinely-parametrized linear partial differential equations

Alexandre Janon[1]

June 2010

Contents

[1]Laboratoire Jean Kuntzmann, Université Grenoble 1/INRIA, mailto:alexandre.janon@imag.fr

Introduction and overview

In some contexts (for example, in Monte-Carlo, or quasi-Monte-Carlo sensitivity analysis), it is necessary to compute the solution u_μ of some partial differential equation (PDE) depending on an input parameter vector μ (dependence can be obviously through the PDE coefficients, and also through the domain on which the equation is posed, or through initial/boundary data), for a large number (say, thousands) values of μ. When no analytical solution of the PDE is known, one has to use computationally-intensive numerical methods – such as finite elements or finite differences – to approximate the solution of the PDE. This leads to complex computer codes, which can take several hours to run – for a *single* value of μ. In this case, one easily sees that simply calling the numerical code for each required value of μ – one-by-one – is not practicable at all ; and that we need to use a computationally cheaper code giving a *surrogate* solution $\widetilde{u}_\mu$ (also called *metamodel*) from a value of μ. More specifically, we split our computation in two phases:

- an *offline* phase, where we make a reasonable number of calls to the expensive code in order to "learn" about the solutions of the PDE ;
- an *online* phase, where, given some value of μ, we use the information collected during the offline phase to produce a good approximation of the solution of the PDE for this specific value of μ, faster than if we had no information at all.

This strategy is efficient if the fixed additional cost of data collection during the offline phase (done once for every value of μ) is dominated by a strong reduction in the marginal cost (i.e., per value of μ) when one replaces the original, expensive numerical code by the cheaper "online" code.
In this report, we describe a procedure, named *reduced basis* approximation (RB), specifying the offline and online phases. The key to RB approximation is to look for (during the online phase) a surrogate $\widetilde{u}_\mu$ solution of the PDE as a linear combination of functions from a linearly-independent family (called the reduced basis) that has to be wisely chosen during the offline phase. For the resulting problem to be well-posed, one has to somewhat weaken the PDE, using a Galerkin-like method. An advantage of using such a procedure is that one can quantify the loss of information made when replacing the expensive numerical solution u_μ by $\widetilde{u}_\mu$, using an *a posteriori* fastly-computed *error bound*

ε_μ so that

$$||u_\mu - \widetilde{u}_\mu|| \leq \varepsilon_\mu$$

for some functional norm $||\cdot||$.

The report is organised as follows: in a first part, we present the necessary assumptions on the PDE for RB to be applicable to ; for the sake of simplicity, we restrict ourselves to the case of coercive, linear and "affinely parametrized" (we will explain what this means) PDEs, but it should be noted there are extensions to RB which somewhat relax some of these assumptions ; in a second part, we describe the core of the RB approximation, that are the offline and online phases of computation of the surrogate $\widetilde{u}_\mu$; in a third part, we give a procedure to compute the error bound ε_μ ; in a fourth part, we show two different ways (*greedy* and *POD*) to choose the reduced basis family ; in a fifth and final part, we address the question of using RB to approximate *outputs* of the solution, that are (scalar) functionals of u_μ.

1 Notations and assumptions

Our main reference for sections 1 to 3 is [5].

As said earlier, we denote by $\mu \in \mathscr{P} \subset \mathbf{R}^p$ our parameter vector.

Our unknown is a function u_μ — depending on μ, besides the usual (e.g. time and space) variables — belonging to some function space X, satisfying a μ-parametrized linear partial differential equation, which can be written under a variational (a.k.a. weak) form:

$$a_\mu(u_\mu, v) = f_\mu(v) \quad \forall v \in X \tag{1}$$

where a_μ is a μ-dependent (continuous) bilinear form on X, and f is a (continuous) linear form on X.

We further assume that our PDE is *affinely parametrized*, that is:

- $f_\mu \equiv f$ is μ-independent;
- and a_μ can be written as a sum of Q bilinear forms:

$$a_\mu(w, v) = \sum_{q=1}^{Q} \Theta_q(\mu) a_q(w, v)$$

  where:

- a_q are μ-independent bilinear forms on X ;
- Θ_q are arbitrary functions on $\mathscr{P}$, capturing all dependence in μ ; since these functions are to be evaluated many times during the online phase, we shall require these functions to be fast-computed.

2 Reduced basis approximation

2.1 Opportunity for a RB approximation

The function space X is typically a Sobolev space; such spaces, due to their infinite-dimensional nature, are not amenable to numerical computations. In order to discretize equation (1) and get an approximation of the actual mathematical solution, one can set X to some finite-element space (e.g., $\mathbb{P}^1$) and use (1) to get a linear system of equations satisfied by the coefficients of u_μ in a basis of X. In the sequel, X will always stand for such a space. This means that the reduced basis solution we are to compute is an approximation *to the discrete numerical solution*, and thus we rely on (and assume) the fact that the latter is a good approximation to the analytical solution. A good treatment of finite element discretizations can be found in [7].
The computational cost required to compute this approximation is an increasing function of the dimension of the chosen discrete function space, and unfortunately, it has to be very large for the approximation to be accurate. One explanation for this fact is that those spaces are "too general", that is, they contain too many different functions and so we need to deal with a large number of degrees of freedom for our unknown.
On the other hand, we expect that two solutions of (1), for two different values of μ, will exhibit common "features". We could "encode" those features into a space $\widetilde{X}$, spanned by some (μ-independent) functions $\zeta_1, \zeta_2, \ldots, \zeta_N \in X$, and replace X by $\widetilde{X}$ in (1). The family $\{\zeta_1, \ldots, \zeta_N\}$ is called the *reduced basis*. For a sufficiently regular-in-μ problem, we expect that taking $N << \dim X$ will still give – for a well-choosed reduced basis – an accurate enough solution, yet providing substantial computational economy.

2.2 Offline and online procedures

We suppose, for now, that we are given a reduced basis $\{\zeta_1, \ldots, \zeta_N\}$ of linearly independent functions in X (discussion about choice of this basis can be found

in section 4). We set $\widetilde{X} = \text{Span}\{\zeta_1, \ldots, \zeta_N\}$.
Given $\mu \in \mathscr{P}$, we look for $\widetilde{u}_\mu$ as a function belonging to $\widetilde{X}$, thus :

$$\widetilde{u}_\mu = \sum_{n=1}^{N} c_n(\mu)\zeta_n \tag{2}$$

where the unknowns are $c_n(\mu) \in \mathbf{R}$, $n = 1, \ldots, N$.
These unknowns are to be found by imposing that $\widetilde{u}_\mu$ satisfy a relaxed version of (1); instead of verifying (1) for every $v \in X$, $\widetilde{u}_\mu$ satisfies (1) for every $v \in \widetilde{X}$:

$$a_\mu(\widetilde{u}_\mu, v) = f(v) \quad \forall v \in \widetilde{X} \tag{3}$$

Since $\widetilde{X} \subset X$, (3) is weaker than (1). This standard technique is known as *Galerkin projection*.
Substituting (2) into (3), and setting $v = \zeta_p$ $(p = 1, \ldots, N)$ gives that the $c_n(\mu)$'s satisfy the following system of N linear equations:

$$\sum_{n=1}^{N} \left[\sum_{q=1}^{Q} \Theta_q(\mu) a_q(\zeta_n, \zeta_p) \right] c_n(\mu) = f(\zeta_p) \quad \forall p = 1, \ldots, N \tag{4}$$

This justifies the following offline/online procedure for the computation of $\widetilde{u}_\mu$:

- in the offline phase:
  - we choose a reduced basis $\{\zeta_1, \ldots, \zeta_N\}$ (more details about this choice later)
  - we assemble, and store, Q matrices of size $N \times N$:

$$\mathbf{a}_q = \left(a_q(\zeta_n, \zeta_p) \right)_{p,n=1,\ldots,N}$$

    and the N-sized vector:

$$\mathbf{f} = \left(f(\zeta_p) \right)_{p=1\ldots N}$$

- in the online phase, being given a value of μ:
  - we assemble the $N \times N$ matrix:

$$\mathbf{a}(\mu) = \sum_{q=1}^{Q} \Theta_q(\mu) \mathbf{a}_q$$

– our vector of unknowns $\mathbf{c}(\mu) = \left(c_n(\mu)\right)_{n=1...N}$ is given by :

$$\mathbf{c}(\mu) = \left(\mathbf{a}(\mu)\right)^{-1} \mathbf{f}$$

where the matrix inversion solves the system (4),

– and so we get:

$$\widetilde{u}_\mu = \sum_{n=1}^{N} c_n(\mu)\zeta_n$$

The complexity of the online phase is as follows: assembling of $\mathbf{a}(\mu)$ takes $O(QN^2)$ operations. Its inversion takes $O(N^3)$ operations (one should note that $\mathbf{a}(\mu)$ has no reason to be a sparse matrix – it is not, indeed). So the coefficients of $\widetilde{u}_\mu$ with respect to the reduced basis can be computed in less than $O(QN^2 + N^3)$ operations.

It is important to note that this complexity is *independent* of $\dim X$. Thus online computation time is independent of the "quality" of the underlying reference numerical approximation, which can be as good as wanted without impacting marginal cost. We notice that we neglected the cost of evaluations of $\Theta_q(\mu)$, hence the hypothesis of fast-evaluation of Θ_q we have made. We also notice that online complexity (in computational time and in storage requirements) is proportional to Q, the number of terms in the affine decomposition of a_μ; for our procedure to be efficient, Q should therefore be kept orders of magnitude smaller than $\dim X$.

3 Error bound

3.1 Derivation of the error bound

In this section, we design an error bound ε_μ so that:

$$||u_\mu - \widetilde{u}_\mu|| \leq \varepsilon_\mu$$

for all $\mu \in \mathscr{P}$.

We should note once again that, thanks to our assumption about the quality of the underlying approximation, we measure only the error between the reference, expensive numerical solution u and our surrogate $\widetilde{u}$; error between u and the analytical solution is neglected.

We endow X with an Hilbert structure: an inner product $< \cdot, \cdot >$ on $X \times X$

yielding a norm $||\cdot||$ on X. We need a crucial assumption about the bilinear form a_μ describing our PDE: the $\alpha(\mu)$ defined by:

$$\alpha(\mu) = \inf_{w \in X} \sup_{v \in X} \frac{a_\mu(w,v)}{||w||||v||}$$

has to be strictly positive: $\alpha(\mu) > 0 \ \ \forall \mu \in \mathscr{P}$.
The quantity $\alpha(\mu)$ is known as Babuška's "inf-sup" stability constant. The inf-sup stability hypothesis $\alpha(\mu) > 0$ is a sufficient condition for well-posedness of the underlying variational problem (1) (Babuška-Brezzi condition).
In the sequel, we suppose that a_μ is *symmetric*. The inf-sup stability hypothesis is then equivalent to the coercivity of a_μ (and the well-posedness of the variational problem comes from Lax-Milgram), and we have:

$$\alpha(\mu) = \inf_{w \in X} \frac{a_\mu(w,w)}{||w||^2}$$

Let's move on to the error bound. First we remember that:

$$a_\mu(u_\mu, v) = f_\mu(v) \quad \forall v \in X$$

Subtracting $a_\mu(\widetilde{u}_\mu, v)$ from both sides of this equation, we get, thanks to bilinearity of a_μ:

$$a_\mu(u_\mu - \widetilde{u}_\mu, v) = f_\mu(v) - a_\mu(\widetilde{u}_\mu, v) \quad \forall v \in X$$

So:

$$a_\mu(e_\mu, v) = r_\mu(v) \quad \forall v \in X \tag{5}$$

if we set:

$$e_\mu = u_\mu - \widetilde{u}_\mu$$

$$r_\mu(v) = f(v) - a_\mu(\widetilde{u}_\mu, v)$$

(e_μ is known as the *error*, and r_μ as the *residual* form).
Introducing $\alpha(\mu)$, we have the following inequality:

$$a_\mu(e_\mu, e_\mu) \geq \alpha(\mu)||e_\mu||^2$$

We also have:

$$||r_\mu(v)|| \leq ||r_\mu||_{X'}||v||$$

where

$$||r_\mu||_{X'} = \sup_{v \in X ||v||=1} r_\mu(v)$$

is the dual norm of the residual.
Thanks to these inequalities, and (5) with $v = e_\mu$, we get:

$$\alpha(\mu)||e_\mu||^2 \leq ||r_\mu||_{X'}||e_\mu||$$

and so, thanks to our stability assumption:

$$||e_\mu|| = ||u_\mu - \widetilde{u}_\mu|| \leq \frac{||r_\mu||_{X'}}{\alpha(\mu)} \tag{6}$$

So we have:

$$\varepsilon_\mu = \frac{||r_\mu||_{X'}}{\alpha(\mu)}$$

In order to practically exploit the error bound (6), it remains to explain how to efficiently work out $||r_\mu||$ and $\alpha(\mu)$. This question is addressed in the two following sections.

3.2 Dual norm of the residual $||r_\mu||_{X'}$

Riesz representation theorem ensures that there exist $y_\mu \in X$ so that

$$< y_\mu, v >= r_\mu(v) \quad \forall v \in X \tag{7}$$

Moreover, computing y_μ is sufficient because of the additional property:

$$||r_\mu||_{X'} = ||y_\mu||$$

Equation (7) can be rewritten, thanks to the definition of r_μ and the affine decomposition of a_μ:

$$< y_\mu, v >= f(v) - \sum_{q=1}^{Q} \Theta_q(\mu) a_q(\widetilde{u}_\mu, v) \quad \forall v \in X$$

and so, making use of the reduced basis decomposition of $\widetilde{u}_\mu$ (equation (2)):

$$< y_\mu, v >= f(v) - \sum_{q=1}^{Q} \sum_{n=1}^{N} c_n(\mu) a_q(\zeta_n, v) \quad \forall v \in X$$

This equation can be viewed as a linear variational problem whose unknown is $y_\mu \in X$. By Duhamel superposition principle, y_μ can be written as:

$$y_\mu = \gamma - \sum_{q=1}^{Q}\sum_{n=1}^{N} \Theta_q(\mu) c_n(\mu) \Gamma_{qn} \tag{8}$$

where $\gamma \in X$, and $\Gamma_{qn} \in X$ $(q = 1, \ldots, Q,\ n = 1, \ldots, N)$ satisfy:

$$< \gamma, v >= f(v) \quad \forall v \in X \tag{9}$$

$$< \Gamma_{qn}, v >= a_q(\zeta_n, v) \quad \forall v \in X \tag{10}$$

Law of cosines applied to (8) gives:

$$\begin{aligned} ||y_\mu||^2 = ||\gamma||^2 - 2\sum_{q=1}^{Q}\sum_{n=1}^{N} \Theta_q(\mu) c_n(\mu) < \gamma, \Gamma_{qn} > \\ + \sum_{q=1}^{Q}\sum_{q'=1}^{Q}\sum_{n=1}^{N}\sum_{n'=1}^{N} \Theta_q(\mu) c_n(\mu) \Theta_{q'}(\mu) c_{n'}(\mu) < \Gamma_{qn}, \Gamma_{q'n'} > \end{aligned} \tag{11}$$

Equation (11) will be our key to an offline/online procedure for computing $||r_\mu||_{X'}$. Each of those phases should be done after the phases of computation of $\widetilde{u}(\mu)$ described in section 2.2. The procedures read:

- in the offline phase:
  - we solve the $QN + 1$ linear variational problems (9) and (10) for γ and Γ_{qn} $(q = 1, \ldots, Q,\ n = 1, \ldots, N)$;
  - we compute and store $1 + QN + (QN)^2$ inner products:

    $$< \gamma, \gamma >,\ < \gamma, \Gamma_{qn} >,\ < \Gamma_{qn}, \Gamma_{q'n'} >$$

    $(q, q' = 1, \ldots, Q,\ n, n' = 1, \ldots, N)$
- in the online phase, we compute $||r_\mu||_{X'} = ||y_\mu||$ using equation (11).

We note that the vectors γ and Γ_{qn} computed during offline phase are *not* saved for the online phase, we store only $1 + QN + (QN)^2$ scalars. Thus in the online phase, the actual vector y_μ is never formed. Doing so would make complexity of the online phase depend on $\dim X$; this is something we would like to avoid.

3.3 Stability constant $\alpha(\mu)$

We now turn to the problem about estimating $\alpha(\mu)$.
Recall that

$$\alpha(\mu) = \inf_{v \in X || v || = 1} a_\mu(v, v)$$

$\alpha(\mu)$ is thus the optimal value of a minimization problem posed on X. The objective function in this problem is a quadratic form, we seek to minimize on the unit sphere. It is a well-known fact that $\alpha(\mu)$ is the smallest eigenvalue of a (symmetric) matrix associated to the quadratic form $a_\mu(\cdot, \cdot)$. Indeed, if for all v and w in X: $a_\mu(v, w) = v^T M_\mu w$ and if $< v, w >= v^T \Omega w$ for symmetric matrices M_μ and Ω, then $\alpha(\mu)$ is the smallest $\lambda \in \mathbf{R}$ so that equation :

$$M_\mu v = \lambda \Omega v$$

admits a nonzero solution v.
Due to the high-dimensionality of X, it would be too expensive to solve this (generalized) eigenproblem online, for each value of μ. However, we expect that affine decomposition of a_μ will provide us with an efficient offline/online procedure for estimating $\alpha(\mu)$. By estimating here we mean finding a lower bound for $\alpha(\mu)$, because the form of the error estimator (6) shows us we only need such a lower bound:

$$\widetilde{\alpha}(\mu) \leq \alpha(\mu)$$

because we have therefore:

$$||u - \widetilde{u}|| \leq \frac{||r||_{X'}}{\widetilde{\alpha}(\mu)}$$

3.3.1 An easy case

We should first address the case where $\Theta_q(\mu) \geq 0 \ \ \forall q = 1, \ldots, Q, \forall \mu \in \mathscr{P}$.

If this condition is met, then for any $\overline{\mu} \in \mathscr{P}, \mu \in \mathscr{P}$:

$$\begin{aligned}
\sum_{q=1}^{Q} \Theta_q(\mu) a_q(u,v) &\geq \sum_{q=1}^{Q} \frac{\Theta_q(\mu)}{\Theta_q(\overline{\mu})} \Theta_q(\overline{\mu}) a_q(u,v) \\
&\geq \left\{ \inf_{q'=1,\ldots,Q} \frac{\Theta_q(\mu)}{\Theta_q(\overline{\mu})} \right\} \sum_{q=1}^{Q} \Theta_q(\overline{\mu}) a_q(u,v) \\
&\geq \left\{ \inf_{q'=1,\ldots,Q} \frac{\Theta_q(\mu)}{\Theta_q(\overline{\mu})} \right\} \alpha(\overline{\mu}) ||u|| ||v||
\end{aligned}$$

This leads us to the following offline/online procedure for $\widetilde{\alpha}(\mu)$:

- Offline:
  - choose a "nominal" value $\overline{\mu} \in \mathscr{P}$ for which $\Theta_q(\overline{\mu}) \neq 0 \ \forall q = 1, \ldots, Q$;
  - compute and store $\alpha(\overline{\mu})$ by solving an eigenproblem on X.
- Online, given a value of μ, take

$$\widetilde{\alpha}(\mu) = \left\{ \inf_{q=1,\ldots,Q} \frac{\Theta_q(\mu)}{\Theta_q(\overline{\mu})} \right\} \alpha(\overline{\mu})$$

This procedure is simple and efficient, with a $\dim X$-independent online stage. However, if our positivity hypothesis on Θ_q is violated, we have to resort to a more complex method, which we describe below.

3.3.2 The general case: successive constraints method (SCM)

The successive constraints method (SCM) is presented in [3].
$\alpha(\mu)$ can be rewritten

$$\alpha(\mu) = \inf_{w \in X, ||w||=1} a_\mu(w,w)$$

We further note that:

$$\alpha(\mu) = \inf_{w \in X, ||w||=1} \left\{ \sum_{q=1}^{Q} \Theta^q(\mu) a_q(w,w) \right\} = \inf_{y \in \mathscr{Y}} \left\{ \sum_{q=1}^{Q} \Theta^q(\mu) y_q \right\} \qquad (12)$$

where:

$$\mathscr{Y} = \left\{(y_1, \ldots, y_q) \in \mathbf{R}^Q \text{ st. } \exists w \in X, ||w|| = 1 \text{ and } y_q = a_q(w,w) \forall q = 1, \ldots, Q\right\}$$

To obtain a lower bound on $\alpha(\mu)$, we should replace $\mathscr{Y}$ by a $\widetilde{\mathscr{Y}} \supseteq \mathscr{Y}$; in other words, we should relax the constraints on the optimization problem (12).
To define $\widetilde{\mathscr{Y}}$, we begin by noticing that $\mathscr{Y}$ is bounded: let:

$$\sigma_q^- = \inf_{w \in X, ||w||=1} a_q(w,w), \quad \sigma_q^+ = \sup_{w \in X, ||w|||=1} a_q(w,w), \quad \mathscr{B} = \prod_{q=1}^{Q} [\sigma_q^-; \sigma_q^+].$$

Now let $y = (y_1, \ldots, y_Q) \in \mathscr{Y}$ and let w satisfying the properties in the definition of $\mathscr{Y}$. For every $\mu' \in \mathscr{P}$, we have:

$$\begin{aligned} \sum_q \Theta_q(\mu') y_q &= \sum_q \Theta_q(\mu') a_q(w,w) \\ &\geq \inf_{w' \in X, ||w'||=1} \sum_q \Theta^q(\mu') a_q(w',w') \\ &= \alpha(\mu') \end{aligned}$$

So far we have:

$$\mathscr{Y} \subset \left\{ y \in \mathscr{B} \mid \sum_{q=1}^{Q} \Theta_q(\mu') y_q \geq \alpha(\mu') \ \ \forall \mu' \in \mathscr{P} \right\}$$

The right-hand side of this inclusion would be a good candidate for $\widetilde{\mathscr{Y}}$; however it is defined by an infinite number of μ-independent constraints (each for one $\mu' \in \mathscr{P}$). We should relax one more time these constraints in order to get a finite number of (now, μ-dependent) constraints. We take:

$$\widetilde{\mathscr{Y}} = \left\{ y \in \mathscr{B} \mid \sum_{q=1}^{Q} \Theta_q(\mu') y_q \geq \alpha(\mu') \ \ \forall \mu' \in \mathscr{S}_M(\mu, \mathscr{C}) \right\}$$

Where $M \in \mathbf{N}$, $\mathscr{C}$ is a finite subset of $\mathscr{P}$ and $S_M(\mu, \mathscr{C})$ is the set of the M points in $\mathscr{C}$ that are closest (with respect to some metric) to μ.
So our SCM offline/online procedure is:

- offline:

- choose M; choose $\mathscr{C}$ (we postpone discussion about the choice of $\mathscr{C}$ until the next subsection) ;
- for each $\mu' \in \mathscr{C}$, compute and store $\alpha(\mu')$ by solving an eigenproblem on X; compute and store σ_q^- and σ_q^+ the same way, for each $q = 1, \ldots, Q$.

- online: solve optimization problem:

$$\widetilde{\alpha}(\mu) = \inf_{y=(y_1,\ldots,y_Q)\in\widetilde{\mathscr{Y}}} \left\{ \sum_{q=1}^{Q} \Theta^q(\mu) y_q \right\}$$

The online optimization problem is what is called a *linear programming problem*: the goal is to minimize a linear function under a finite number of linear inequalities constraints. In our case there are Q variables and $Q+M$ constraints. There exist efficient algorithms, such as the simplex algorithm (see [6] for instance), which solve such optimization problems under (on average) polynomial complexity with respect to the number of variables and number of constraints, even if they can be exponential in the worst cases. The key here, once again, is that online complexity is still independent of $\dim X$.
A last remark we can do on the algorithm is about the trade-off in the choice of M: whatever M is, we always get a certified bound on $\alpha(\mu)$, but increasing M will improve sharpness of this bound, at the expense of an increase in online computation time.

Choice of $\mathscr{C}$

We will use a so-called *greedy* procedure for choosing the finite subset $\mathscr{C} \subset \mathscr{P}$. Roughly speaking, a decision algorithm is said to be *greedy* if he makes the best possible choice at each step. This strategy does not always yield a globally optimal decision but often gives a good solution for cases in direct global optimization would be unpractical.
For our greedy choice of $\mathscr{C}$, we will start with $\mathscr{C} = \{\mu_1\}$, with an arbitrarily chosen $\mu_1 \in \mathscr{P}$. Then, at each step of the algorithm, we are going to add a point in $\mathscr{C}$ that gives the best possible expected improvement in the precision of our bound $\widetilde{\alpha}(\mu)$.
Thus we get the following procedure for choice of $\mathscr{C}$:

1. choose M ;
2. initialize $\mathscr{C} = \{\mu_1\}$ with some arbitrary $\mu_1 \in \mathscr{P}$;
3. choose a finite-sized sample set $\Xi \subset \mathscr{P}$;
4. repeat:
   - add
     $$\mu^* = \underset{\mu \in \Xi}{\operatorname{argmax}}\, \alpha(\mu) - \widetilde{\alpha}(\mu)$$
     to $\mathscr{C}$ (here $\widetilde{\alpha}(\mu)$ stands for the SCM lower bound computed using the "current" $\mathscr{C}$)

The repeat loop can be stopped either when $\#\mathscr{C}$ has reached a maximal value, or when $\max_{\mu \in \Xi} \alpha(\mu) - \widetilde{\alpha}(\mu)$ is less than a desired precision.
The greedy algorithm written below would require computation of $\alpha(\mu)$ for $\#\Xi \times (1 + 2 + \ldots + (\#\mathscr{C} - 1))$ values of μ. If one choose a large Ξ set, the cost of the choice of $\mathscr{C}$ might be too prohibitive. Rather, one could consider a cheap *upper* bound $\widetilde{\alpha}^{up}(\mu)$ for $\alpha(\mu)$ and replace $\alpha(\mu) - \widetilde{\alpha}(\mu)$ by the relative surrogate sharpness estimator: $\dfrac{\widetilde{\alpha}^{up}(\mu) - \widetilde{\alpha}(\mu)}{\widetilde{\alpha}^{up}(\mu)}$.
This cheap upper bound could be built using the following: let us define:

$$\widetilde{\mathscr{Y}}^{up} = \{y^*(\mu_k), k = 1, \ldots, K\}$$

where

$$\{\mu_1, \ldots, \mu_K\} = \mathscr{C}$$

and

$$y^*(\mu_k) = \underset{y \in \mathscr{Y}}{\operatorname{arginf}} \left\{ \sum_{q=1}^{Q} \Theta_q(\mu_k) y_q \right\} \quad (k = 1, \ldots, K)$$

We can define:

$$\widetilde{\alpha}^{up}(\mu) = \inf_{y \in \widetilde{\mathscr{Y}}^{up}} \left\{ \sum_{q=1}^{Q} \Theta_q(\mu) y_q \right\}$$

which satisfies $\widetilde{\alpha}^{up}(\mu) \geq \alpha(\mu)$ because $\widetilde{\mathscr{Y}}^{up} \subset \mathscr{Y}$.
Using $\widetilde{\alpha}^{up}(\mu)$ as a surrogate for $\alpha(\mu)$ reduces fixed complexity of the greedy algorithm to $K = \#\mathscr{C}$ eigenproblems on X (for computation of the $y^*(\mu_k)$) and

$\#\Xi$ trivial minimizations per step over reasonably-sized, finite, $\widetilde{\mathscr{Y}}^{up}$.

4 Choice of a reduced basis

In this section, we present two methods for choosing our reduced basis $\{\zeta_1, \ldots, \zeta_N\}$. These two methods have different offline complexities and lead to different bases. Both nicely fit into our reduced basis framework, and can benefit from the same procedure for error estimation, detailed in section 3.

4.1 Greedy procedure

The greedy procedure is treated in [5].
The following algorithm is based on the same heuristic than the one described earlier for choosing $\mathscr{C}$ in the SCM: at each step of the algorithm, we put one new function in the reduced basis; this function is the solution u_{μ^*} of the PDE for a specific value of the parameter choosed so as to give the best expected improvement in the online error bound, i.e. the value of the parameter for which we make the greatest (estimated) error when using "current" reduced basis.
The greedy algorithm for choice of the reduced basis is the following: given $N \in \mathbf{N}$, the desired final size of the reduced basis,

- Choose a finite-sized, random, large $\Xi \subset \mathscr{P}$ sample of parameters.
- Choose $\mu_1^* \in \mathscr{P}$ at random, and set $\zeta_1 = u_{\mu_1^*}$.
- Repeat, for n from 2 to N:
  - For each $\mu \in \Xi$, compute the online error bound for $\widetilde{u}_\mu$ when using $\{\zeta_1, \ldots, \zeta_{n-1}\}$ as reduced basis.
  - Find $\mu_n^* \in \Xi$ with the greatest online error bound.
  - Set $\zeta_n = u_{\mu_n^*}$.

This procedure requires computation of N actual solutions, and $\#\Xi \times N$ online error bounds. It relies crucially on the existence of a cheap online error bound. Of course, instead of fixing a target size for the reduced basis, one can also keep iterating and adding functions to the reduced basis until the greatest online error bound (on the Ξ sample of parameters) gets below some prescribed value.

4.2 Proper Orthogonal Decomposition (POD)-driven procedure

To our knowledge, the following presentation of the Proper Orthogonal Decomposition is new. We refer to [8] for a different presentation, based on singular value decomposition (SVD).

First mode

Suppose, for a soft start, that we want to find a reduced basis containing only one item: $\{\zeta_1\}$. Without loss of generality we can always suppose that $||\zeta_1|| = 1$. It seems reasonable to make the choice which makes the least summed squares error when projecting orthogonally the solutions u_μ (for all $\mu \in \mathscr{P}$) onto the reduced space Span $\{\zeta_1\}$. Said with formulas, we seek ζ_1 satisfying:

$$\zeta_1 = \underset{p\in X, ||p||=1}{\operatorname{arginf}} \int_{\mu\in\Xi} \left|\left|u_\mu - < u_\mu, p > p\right|\right|^2 \mathrm{d}\mu$$

The function ζ_1 will be called our *first POD mode*.
Unfortunately, the integral appearing in the objective function below is, in general, analytically untractable. We can approximate it using a finite sample of parameters $\Xi \subset \mathscr{P}$; the integral then gets replaced by a finite sum over Ξ:

$$\zeta_1 = \underset{p\in X, ||p||=1}{\operatorname{arginf}} \sum_{\mu\in\Xi} \left|\left|u_\mu - < u_\mu, p > p\right|\right|^2$$

The set of solutions $\{u_\mu, \mu \in \Xi\}$ is called the *snapshots* ensemble.
The objective function can then be rewritten:

$$\begin{aligned}\sum_{\mu\in\Xi} \left|\left|u_\mu - < u_\mu, p > p\right|\right|^2 &= \sum_{\mu\in\Xi} \left|\left|u_\mu\right|\right|^2 - 2\sum_{\mu\in\Xi} < u_\mu, p >^2 + \sum_{\mu\in\Xi} < u_\mu, p >^2 ||p||^2 \\ &= \sum_{\mu\in\Xi} \left|\left|u_\mu\right|\right|^2 - \sum_{\mu\in\Xi} < u_\mu, p >^2 \text{ using that } ||p||^2 = 1\end{aligned}$$

And so:

$$\zeta_1 = \underset{p\in X, ||p||=1}{\operatorname{argmax}} \sum_{\mu\in\Xi} < u_\mu, p >^2 \tag{13}$$

The objective function in (13) is a quadratic form in p, that has to be maximized under a "unity norm" constraint. We know this can be reduced to an eigenproblem.
More specifically, let us introduce a basis $\{\phi_1, \ldots, \phi_{\mathcal{N}}\}$ of X. As we stressed

in the introduction, $\mathcal{N}$ is large. We enumerate the snapshots parameters: $\Xi = \{\mu_1, \ldots, \mu_{N_{snap}}\}$.
Each snapshot $u(i) = u_{\mu_i}$ $(i = 1, \ldots, N_{snap})$ has an expansion in the basis of X:

$$u(i) = \sum_{j=1}^{\mathcal{N}} u(i)_j \phi_j$$

and we can write down M, the matrix of the $u(i)_j$ coefficients:

$$M = \begin{pmatrix} u(1)_1 & u(2)_1 & \ldots & u(N_{snap})_1 \\ u(1)_2 & u(2)_2 & \ldots & u(N_{snap})_2 \\ \vdots & \vdots & \ldots & \vdots \\ u(1)_{\mathcal{N}} & u(2)_{\mathcal{N}} & \ldots & u(N_{snap})_{\mathcal{N}} \end{pmatrix}$$

Denote by Ω, the (symmetric, positive definite) matrix of the scalar product $<,>$ in X.
We can write down our objective function as:

$$\sum_{\mu \in \Xi} < u_\mu, p >^2 = p^T \Omega^T M M^T \Omega p$$

The condition of optimality for (13) is then given by the Lagrange multipliers theorem: there has to exist $\lambda \in \mathbf{R}$, so that:

$$\Omega^T M M^T \Omega \zeta_1 = \lambda \Omega \zeta_1 \tag{14}$$

and λ is the optimal value of the problem (as can be seen by left-multiplying (14) by ζ_1^T, and using $||\zeta_1||^2 = 1$) – which should be made as large as possible.
Now left-multiply (14) by $\Omega^{-T} = \Omega^{-1}$:

$$M M^T \Omega \zeta_1 = \lambda \zeta_1 \tag{15}$$

so that ζ_1 is the unit eigenvector associated with the greatest eigenvalue of $M M^T \Omega$.
So we have reduced our problem as an eigenproblem of dimension $\mathcal{N}$, because $M M^T \Omega$ is of size $\mathcal{N} \times \mathcal{N}$. Since $\mathcal{N}$ can be large, this could turn out to be highly unpractical. Fortunately, we are to see that we can solve the same problem using eigenproblem of size $N_{snap} \times N_{snap}$.
The key is to look for ζ_1 as a linear combination *of the snapshots*, instead of the

"generic" basis items $\{\phi_i\}$.
We write:

$$\zeta_1 = \sum_{i=1}^{N_{snap}} z_{1i} u(i) = M z_1$$

where $z_1 = (z_1 1, \ldots, z_{1,N_{snap}})^T$. Setting $\zeta_1 = M z_1$ into (15) yields:

$$M M^T \Omega M z_1 = \lambda M z_1$$

and a sufficient condition for this to hold is:

$$M^T \Omega M z_1 = \lambda z_1$$

so that our new unknown $z_1 \in \mathbf{R}^{N_{snap}}$ is an eigenvector of $M^T \Omega M$ (which of size $N_{snap} \times N_{snap}$) associated with its greatest eigenvalue.
The absence of optimality loss when looking for ζ_1 in Range(M) (in other words, $\zeta_1 = M z_1$ with z_1 as defined below satisfies (13)) follows clearly from these facts:

1. if z is a nonzero eigenvector of $M^T \Omega M$ (associated with $\lambda \neq 0$), then $\zeta = Mz$ is a nonzero eigenvector of $M M^T \Omega$ associated with λ ;
2. if ζ is a nonzero eigenvector of $M M^T \Omega$ (associated with $\lambda \neq 0$), then $z = M^T \Omega \zeta$ is a nonzero eigenvector of $M^T \Omega M$ associated with λ;
3. $M^T \Omega M$ and $M M^T \Omega$ have the same nonzero eigenvalues, each with the same multiplicity in one and the other.

Proof: To see 1., first note that if we had $Mz = 0$, then $M^T \Omega M z = 0$ and so z could not be an eigenvector associated with a nonzero eigenvalue of $M^T \Omega M$; thus $Mz \neq 0$. Now $M M^T \Omega (Mz) = M(M^T \Omega M z) = M \lambda z = \lambda (Mz)$ so that Mz is an eigenvector of $M M^T \Omega$ associated with λ. Point 2. is 1. *mutatis mutandis*. Point 3.: it is clear from 1. and 2. that the two matrices have the same nonzero eigenvalues. To get the statement about the multiplicities, suppose that Span$\{z_1, \ldots, z_k\}$ is the eigenspace of $M^T \Omega M$ associated with eigenvalue λ. Since $M^T \Omega M$ is self-adjoint with respect to the Euclidean inner product (i.e., is symmetric), we can suppose that $\{z_1, \ldots, z_k\}$ is orthogonal (for the Euclidean inner product). Now, from 2., Span$\{Mz_1, \ldots, Mz_k\}$ is contained in the eigenspace of $M M^T \Omega$ associated with λ. Moreover, for every $i \neq j$:

$$< Mz_i, Mz_j > = z_j^T M^T \Omega M z_i = \lambda z_j^T z_i = 0$$

so that $\{Mz_1, \ldots, Mz_k\}$ is $<,>$-orthogonal, and therefore linearly independent, and so λ has multiplicity k in $MM^T\Omega$. $\square$

Second mode

Now let's say we are looking for a set of two functions as our reduced basis: $\{\zeta_1, \zeta_2\}$. As earlier, we want (ζ_1, ζ_2) to minimize:

$$\sum_{\mu\in\Xi} \left|\left|u_\mu - (< u_\mu, \zeta_1 > \zeta_1 + < u_\mu, \zeta_2 > \zeta_2)\right|\right|^2$$

among the $(\zeta_1, \zeta_2) \in X \times X$ satisfying: $||\zeta_1|| = ||\zeta_2|| = 1$ and $< \zeta_1, \zeta_2 >= 0$.
In other words, we aim to minimize the overall error (over our sample of parameters Ξ) made when orthogonally projecting u_μ onto Span$\{\zeta_1, \zeta_2\}$, with $\{\zeta_1, \zeta_2\}$ an orthonormal family.
Mimicking what we have done earlier, we rewrite our objective function as:

$$\begin{aligned}
\sum_{\mu\in\Xi} \left|\left|u_\mu - (< u_\mu, \zeta_1 > \zeta_1 + < u_\mu, p > p)\right|\right|^2 &= \sum_{\mu\in\Xi} ||u_\mu||^2 - 2\sum_{\mu\in\Xi} < u_\mu, \zeta_1 >^2 - 2\sum_{\mu\in\Xi} < u_\mu, p >^2 \\
&\quad + \sum_{\mu\in\Xi} < u_\mu, \zeta_1 > ||\zeta_1||^2 + \sum_{\mu\in\Xi} < u_\mu, p >^2 ||p||^2 \\
&\quad + \sum_{\mu\in\Xi} < u_\mu, \zeta_1 >< u_\mu, \zeta_2 >< \zeta_1, \zeta_2 > \\
&= \text{Constant} - \sum_{\mu\in\Xi} < u_\mu, \zeta_1 >^2 - \sum_{\mu\in\Xi} < u_\mu, \zeta_2 >^2
\end{aligned}$$

because of the constraints on ζ_1 and ζ_2.
The objective function, apart from the terms that are independent of ζ_1 and ζ_2, is a sum of a function dependent on ζ_1 only and a function dependent on ζ_2 only. Thus the joint optimization on (ζ_1, ζ_2) is equivalent to successive optimization on ζ_1, then on ζ_2. Optimization on ζ_1 has been done previously. Optimization in ζ_2 is the following:

$$\zeta_2 = \underset{p\in X, ||p||=1, <p,\zeta_1>=0}{\text{argmax}} \sum_{\mu\in\Xi} < u_\mu, p >^2$$

The objective function has exactly the same matrix expression as the one of the problem defining the first POD mode:

$$\sum_{\mu\in\Xi} < u_\mu, p >^2 = p^T\Omega^T MM^T\Omega p$$

The Lagrange multiplier theorem now gives that there exist $\lambda_2 \in \mathbf{R}$ so that:

$$\Omega^T M M^T \Omega \zeta_2 = \lambda_2 \Omega \zeta_2 \tag{16}$$

As before, left multiplying by ζ_2^T gives that λ_2 is the optimal value of the problem, which should be made as large as possible.
Left-multiplying (16) by Ω^{-T} we get that (16) is equivalent to:

$$M M^T \Omega \zeta_2 = \lambda \zeta_2$$

so λ_2 is an eigenvalue of $M M^T \Omega$.
Now we have, for every $x, y \in X$:

$$\begin{aligned} < M M^T \Omega x, y > &= y^T \Omega M M^T \Omega x \\ &= y^T \left(M M^T \Omega \right)^T \Omega x \\ &= < x, M M^T \Omega y > \end{aligned}$$

so that $M M^T \Omega$ is self-adjoint with respect to $<,>$. So we can find a unit eigenvector ζ_2, orthogonal to ζ_1 associated with the second greatest eigenvalue (note that this eigenvalue is equal to λ_1 if λ_1 has multiplicity > 1).
As earlier, we can use the fact that $M M^T \Omega$ and $M^T \Omega M$ have the same nonzero eigenvalues, to get $\zeta_2 = M z_2$ where z_2 is an eigenvector of the smaller matrix $M^T \Omega M$. The fact that multiplicities of nonzero eigenvalues in $M M^T \Omega$ and $M^T \Omega M$ are the same ensures that this procedure is correct even if $\lambda_1 = \lambda_2$.

Following modes, and summary for POD

We can readily repeat what we have done for ζ_2 to prove:
Let $\Xi = \{\mu_1, \ldots, \mu_{N_{snap}}\}$ be a finite part of $\mathscr{P}$. Let M be the matrix whose columns are the coefficients of $u_{\mu_1}, \ldots, u_{\mu_{N_{snap}}}$ with respect to some basis of X.
Let Ω be the matrix of the inner product in X.
Then, the family $(\zeta_1, \ldots, \zeta_N)$ of vectors in X that minimizes:

$$\sum_{\mu \in \Xi} \left|\left| u(\mu) - \left(< u_\mu, \zeta_1 > \zeta_1 + < u_\mu, \zeta_2 > \zeta_2 + \ldots + < u_\mu, \zeta_N > \zeta_N \right) \right|\right|^2$$

under the constraints $< \zeta_i, \zeta_j > = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ else.} \end{cases}$

is defined by:

$$\begin{cases} \zeta_1 &= \dfrac{1}{||Mz_1||} Mz_1 \\ &\vdots \\ \zeta_N &= \dfrac{1}{||Mz_n||} Mz_n \end{cases}$$

where z_k $(k = 1, \ldots, N)$ is a nonzero eigenvector, associated with k-th largest eigenvalue (counting repeated nonsimple eigenvalues) of $M^T \Omega M$.
The statement above holds for every $N \in \mathbf{N}^*$ so that the N-th largest eigenvalue of $M^T \Omega M$ is nonzero.
The functions $(\zeta_1, \ldots, \zeta_N)$ defined above are called the first N POD modes associated with the snapshot ensemble Ξ.
A note about the (offline) complexity of the POD: for a N-sized reduced basis, it requires computation of N_{snap} "expensive" solutions u_μ and extraction of the N leading eigenelements of an $N_{snap} \times N_{snap}$ symmetric matrix.

5 Case of a scalar output

One is often not interested in the full solution u_μ, but rather in an "output" calculated from u_μ:

$$s_\mu = \eta(u_\mu)$$

where $\eta : X \to \mathbf{R}$ is a (μ-independent) functional.
We present here some approaches for computation of a surrogate output $\widetilde{s}_\mu$ from $\widetilde{u}_\mu$.

5.1 Primal approach

A trivial approach to the question is to compute $\widetilde{u}_\mu$, and then, online:

$$\widetilde{s}_\mu = \eta(\widetilde{u}_\mu) \tag{17}$$

Done naively, this procedure might be $\dim X$-*dependent*, since evaluation of η may require access to every component of $\widetilde{u}_\mu$ in the original X space.
A way of overcoming this problem, if η is a linear functional, is to write that:

$$\widetilde{s}_\mu = \eta\left(\sum_{n=1}^{N} c_n(\mu)\zeta_n\right) = \sum_{n=1}^{N} c_n(\mu)\eta(\zeta_n)$$

and the offline/online procedure becomes:

- offline: compute and store $\eta(\zeta_n)$, $n = 1, \ldots, N$
- online: find $c_n(\mu)$ $(n = 1, \ldots, N)$ satisfying (2) using procedure described in 2.2 and compute

$$\widetilde{s}_\mu = \sum_{n=1}^{N} c_n(\mu)\eta(\zeta_n) \tag{18}$$

And now the online phase is still $\dim X$-independent.
For a quadratic output functional, one proceeds the same way using:

$$\widetilde{s}_\mu = \sum_{n=1}^{N}\sum_{n'=1}^{N} c_n(\mu)c_{n'}(\mu)\eta_b(\zeta_n, \zeta_{n'})$$

where η_b is the symmetric bilinear form associated with η.
This time, one has to compute and store (offline) $\eta_b(\zeta_n, \zeta_{n'})$, for $n, n' = 1, \ldots, N$.
The same procedure can easily be generalized for polynomial outputs.

Error bound for the output

A straightforward majoration for $|s_\mu - \widetilde{s}_\mu|$ can be found under a Lipschitz hypothesis for η: let $L > 0$ be so that, whatever $v, w \in X$:

$$|\eta(v) - \eta(w)| \leq L||v - w||$$

(for example, if η is a continuous linear functional, L can be chosen as the dual norm of η).
Then we have:

$$|s_\mu - \widetilde{s}_\mu| \leq L\varepsilon_\mu \tag{19}$$

5.2 Primal/dual approach

For linear outputs, error bound (19) can be improved, at the expense of, in some cases, an increase in computation time and program complexity. The idea, present in [5], is to introduce the *dual* problem of (1) associated with η:

$$a_\mu(v, \psi_\mu) = -\eta(v) \quad \forall v \in X \tag{20}$$

whose unknown is $\psi_\mu \in X$.

This dual problem is an affine parametric PDE which we can apply the RB framework on. That is, given a reduced basis $\{\zeta_1^{du}, \dots, \zeta_{N_{du}}^{du}\}$ and a reduced space $\widetilde{X}^{du} = \text{Span}\{\zeta_1^{du}, \dots, \zeta_{N_{du}}^{du}\}$, find an approximation $\widetilde{\psi}_\mu \in \widetilde{X}^{du}$ to ψ_μ satisfying:

$$a_\mu(v, \widetilde{\psi}_\mu) = -\eta(v) \quad \forall v \in \widetilde{X}^{du}$$

with error bound:

$$||\widetilde{\psi}_\mu - \psi_\mu|| \leq \varepsilon_\mu^{du}$$

After having found reduced basis solutions for the primal and dual problems, the following surrogate output can be taken:

$$\widetilde{s}_\mu = \eta(\widetilde{u}_\mu) - r_\mu(\widetilde{\psi}_\mu) \tag{21}$$

The error bound on the output now reads:

$$||\widetilde{s}_\mu - s_\mu|| \leq \alpha(\mu)\varepsilon_\mu\varepsilon_\mu^{du} \tag{22}$$

Using (21) instead of (17) gives us the quadratic error bound (22) instead of the linear (19) for only (roughly) twice the price in computation and storage requirements. And if the problem is self-dual (i.e. a_μ is symmetric and $\eta = -f$), we can take $N = N_{du}$, $\zeta_i = \zeta_i^{du} \ \forall i = 1, \dots, N$ and get this "squared" effect for free, by just adding up the corrective term in (21).

Proof of (22)**:** We omit the μ subscripts.

$$\begin{aligned}
|s - \widetilde{s}| &= |\eta(u - \widetilde{u}) - a(\widetilde{u}, \widetilde{\psi}) + f(\widetilde{\psi})| \\
&= |-a(u - \widetilde{u}, \psi) - a(\widetilde{u}, \widetilde{\psi}) + f(\widetilde{\psi})| \text{ from (20)} \\
&= |-a(u, \psi) + f(\widetilde{\psi}) + a(\widetilde{u}, \psi - \widetilde{\psi})| \text{ by reordering terms} \\
&= |f(\widetilde{\psi} - \psi) + a(\widetilde{u}, \psi - \widetilde{\psi})| \text{ from (1)} \\
&= |r(\psi - \widetilde{\psi})| \text{ by the very definition of } r \\
&\leq ||r||_{X'}||\psi - \widetilde{\psi}|| \leq ||r||_{X'}\varepsilon^{du} = \alpha\varepsilon\varepsilon^{du} \text{ from the definition of } \varepsilon. \qquad \square
\end{aligned}$$

We shall now pass to the offline/online procedure: we first require computation of $r_\mu(\widetilde{\psi}_\mu)$ in (21).
Expanding $\widetilde{\psi}_\mu$ on the dual reduced basis:

$$\widetilde{\psi}_\mu = \sum_{p=1}^{N_{du}} q_p(\mu)\zeta_p^{du}$$

we have:

$$r_\mu(\widetilde{\psi}_\mu) = \sum_{p=1}^{N_{du}} q_n(\mu) r_\mu(\zeta_p^{du}) \tag{23}$$

and, for each $p = 1, \ldots, N_{du}$, we have, from definition of r_μ, affine expansion of a_μ and reduced basis expansion (2) of $\widetilde{u}_\mu$:

$$r_\mu(\zeta_p^{du}) = f(\zeta_p^{du}) - \sum_{q=1}^{Q} \Theta_q(\mu) \sum_{n=1}^{N} c_n(\mu) a_q(\zeta_n, \zeta_p^{du}) \tag{24}$$

So our full offline/online procedure for primal/dual evaluation and error bound on s_μ reads:

- offline:
  - perform offline-stage RB approximation (part 2.2) and error estimation (part 3) for the two problems (1) and (20) ;
  - compute and store $\eta(\zeta_n)$ for $n = 1, \ldots, N$;
  - compute and store $f(\zeta_p^{du})$ $(p = 1, \ldots, N_{du})$ and $a_q(\zeta_n, \zeta_p^{du})$ $(q = 1, \ldots, Q; n = 1, \ldots, N; p = 1, \ldots, N_{du})$;
- online:
  - perform online-stage RB approximation and error estimation for the two problems (1) and (20); this gives respective expansions $\{c_n(\mu)\}_{n=1,\ldots,N}$ and $\{q_p(\mu)\}_{p=1,\ldots,N_{du}}$ of $\widetilde{u}_\mu$ and $\widetilde{\psi}_\mu$ in, respectively primal reduced basis $\{\zeta_1, \ldots, \zeta_N\}$ and dual reduced basis $\{\zeta_1^{du}, \ldots, \zeta_{N_{du}}\}$, together with $\alpha(\mu)$, ε_μ (error bound for primal problem) and ε_μ (error bound for dual problem) ;
  - compute $\eta(\widetilde{\mu})$ as left-hand side of (18) ;
  - compute $r_\mu(\widetilde{\psi}_\mu)$ using (23) and (24) ;
  - output $\widetilde{s}_\mu$ using (21) ;
  - give error bound (22).

5.3 Output-aware choice of basis: goal-oriented POD

When one is interested in a special output of u_μ, it could be wise to choose an *adapted* reduced basis for this output. For instance, if the output depends strongly on the values of $u_\mu(x)$ for x in some region R_1 contained in the domain

of calculus, and much less strongly for x in some region R_2, then a well-chosen reduced basis should lead to a precise $\widetilde{u}_\mu$ approximation in R_1, which could be much less precise in R_2.

The goal-oriented POD method, presented in [1] in the context of linear-time-invariant (LTI) dynamical systems and balanced truncation, is a variation on POD taking the output functional in account.

Goal-oriented POD proceeds by finding $(\zeta_1, \ldots, \zeta_N) \in X$ minimizing the sum of squared differences (for different values of a parameter in a finite sample $\#\Xi$) between the actual output and the output evaluated on the RB approximation taking $\{\zeta_1, \ldots, \zeta_N\}$ so as to minimize

$$\inf \sum_{\mu \in \Xi} \left|\left| \eta(u_\mu) - \eta(\widetilde{u}_\mu) \right|\right|^2 \tag{25}$$

under the constraints:

$$\begin{cases} \widetilde{u}_\mu = \sum_{n=1}^{N} c_n(\mu)\zeta_n \quad \forall \mu \in \Xi \\ \sum_{n=1}^{N} \left[\sum_{q=1}^{Q} \Theta_q(\mu) a_q(\zeta_n, \zeta_p) \right] c_n(\mu) = f(\zeta_p) \quad \forall \mu \in \Xi, \forall p = 1, \ldots, N \\ < \zeta_n, \zeta_m > = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{else} \end{cases} \quad \forall n, m = 1, \ldots, N \end{cases}$$

We believe it is possible to adapt the optimization framework described in [1] to solve (a weaker variant) of the above problem for $(\zeta_1, \ldots, \zeta_N)$. However this optimization problem seems rather difficult (non-convex, with local non-global minima), thus requiring much more computational burden than classic POD.

Conclusion

We have presented a PDE-based, deterministic approach for model reduction called reduced basis approximation. The advantages of this approach are: a certified, computationally-efficient bound for reduction error, and a wide flexibility in the reduced basis chosen so as to enable output-aware choices of basis. Its main drawback is the need to make assumptions on the model that is being reduced. However work have been done to extend the reduced basis, and its associate procedures (error estimation, basis selection) to non-affinely

parametrized PDEs ([2]) or non-linear, time-dependent PDEs (see [4] for instance).
Future work should include further extensions to other PDEs with different parameters, and more output-aware basis choice procedures.

References

[1] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896, 2007.

[2] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.

[3] D.B.P. Huynh, G. Rozza, S. Sen, and A.T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus Mathematique*, 345(8):473–478, 2007.

[4] N.C. Nguyen, G. Rozza, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46(3):157–185, 2009.

[5] N.C. Nguyen, K. Veroy, and A.T. Patera. Certified real-time solution of parametrized partial differential equations. *Handbook of Materials Modeling*, pages 1523–1558, 2005.

[6] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Verlag, 1999.

[7] A.M. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer, 2008.

[8] S. Volkwein. Proper orthogonal decomposition and singular value decomposition. 1999.