

Computational Reduction for Parametrized PDEs: Strategies and Applications

Andrea Manzoni, Alfio Quarteroni and Gianluigi Rozza

Abstract. In this paper we present a compact review on the mostly used techniques for computational reduction in numerical approximation of partial differential equations. We highlight the common features of these techniques and provide a detailed presentation of the reduced basis method, focusing on greedy algorithms for the construction of the reduced spaces. An alternative family of reduction techniques based on surrogate response surface models is briefly recalled too. Then, a simple example dealing with inviscid flows is presented, showing the reliability of the reduced basis method and a comparison between this technique and some surrogate models.

Mathematics Subject Classification (2010). Primary 78M34; Secondary 49J20, 65N30, 76D55.

Keywords. Computational reduction; reduced basis methods; proper orthogonal decomposition; parametrized partial differential equations.

1. Introduction and historical perspective

Scientific computing and numerical simulations in engineering have gained an ever increasing importance during the last decades. In several fields, from aerospace and mechanical engineering to life sciences, numerical simulations of partial differential equations (PDE) provide nowadays a virtual platform ancillary to material/mechanics testing or *in vitro* experiments, useful either for (i) the prediction of input/output response or (ii) the design and optimization of a system [52]. A determinant factor leading to a strong computational speed-up is the constant increase of available computational power, which has gone with the progressive improvement of algorithms for solving large linear systems. Indeed, numerical simulation of turbulent flows, multiscale and multiphysics phenomena, are nowadays possible by means of discretization techniques such as finite elements/volumes or spectral methods, but are very demanding, involving up to $\mathcal{O}(10^6 - 10^9)$ degrees of freedom and several hours (or even days) of CPU time, also on powerful hardware parallel architectures. Nevertheless, it is still very difficult – and often impossible – to deal with *many*

query scenarios, such as those occurring in sensitivity analysis of PDE solutions with respect to parameters, optimization problems under PDE constraints (optimal control, shape optimization), or *real time* simulations and output evaluations. In all these situations, suitable reduction techniques enable the computation of a solution entailing an acceptable amount of CPU time and limited storage capacity.

The goal of a computational reduction technique is to capture the essential features of the input/output behavior of a system in a rapid and reliable way, i.e. *(i)* by improving computational performances and *(ii)* by keeping the approximation error between the reduced-order solution and the full-order one under control. In particular, we aim at approximating a PDE solution using a handful of degrees of freedom instead of millions that would be needed for a full-order approximation. In this way, we need to solve the full-order problem only for few instances of the input (through a demanding *Offline* stage), in order to be able to perform many low-cost reduced-order simulations (very inexpensive *Online* stage) for new instances of the input.

The idea standing at the basis of computational reduction strategies is the assumption (often verified) that the behavior of a system can be well described by a small number of dominant modes. Although reduction strategies have become a very popular research field in the last three decades, we may consider as the earliest attempt of reduction strategy the truncated Fourier series (1806) to approximate a function by means of a small number of trigonometric terms or *modes*, which can be seen as the foundation of the successive projection methods based on a set of snapshots. On the other hand, polynomial interpolation (Waring (1779), rediscovered by Euler in 1783, and published by Lagrange in 1795) can be seen as the earliest kernel of the *surrogate models* (or *metamodels*) used for predicting outputs of interest for combinations of input values which have not been simulated.

Proper orthogonal decomposition is probably the best known technique for computational reduction; it was firstly introduced in statistics as *principal component analysis* by Pearson (1901) [38], then developed independently by Hotelling (1936) [23], and still stands at the basis of statistical multivariate analysis. However, this kind of techniques was not widely exploited until the advent of electronic computers during the 50's, when the first steps towards computational reduction in linear algebra were moved thanks to Lanczos [28] and his iteration method (1950), at the basis of the so-called Krylov subspace expansions. Proper orthogonal decomposition (POD) was introduced for the first time as computational reduction tool by Lumley (1967) and Sirovich (1987) [53] in the analysis of complex turbulent fluid flows. Together with POD in fluid dynamics – a field which has always provided strong impulses to reduction strategies – the use of reduced basis methods was pioneered by Noor (1980) in computational nonlinear mechanics for the instability analysis of structures [35], and then studied by other researchers in the same decade [17, 40, 42].

Computational reduction is not the only available approach to speed-up complex numerical simulations; depending on the applications, we might consider instead *geometrical reduction* techniques or *model reduction* techniques, possibly coupled to computational reduction tools. In particular:

- *Geometrical reduction* techniques are based on a multiscale paradigm, yielding to coupling mathematical models at different *geometric scales* (3D, 2D, 1D and even 0D) through suitable interface conditions. A natural framework where suitable *geometric multiscaling* techniques have been developed is the numerical simulation of blood flows in the circulatory system [39, 47]. Due to different physiological and morphological aspects, three-dimensional models for blood flows in (local) portions of large vessels have been coupled with one-dimensional (global) models for the network of arteries and veins, or even to zero-dimensional (ODE) models for the capillary network. In this case, the geometrical downscaling involves time-dependent Navier-Stokes equations (3D), an Euler hyperbolic system (1D) or lumped parameter models based on ordinary differential equations (0D).
- *Model reduction* techniques can be regarded as strategies based on heterogeneous domain decomposition (DD) methods, which consider different mathematical models in different subdomains [46]. Hence, we *refine* the mathematical model wherever interested to a finer description of the phenomena, without increasing the complexity all over the domain, by treating the interface between the two models as unknown. Classical examples are advection-diffusion problems where a simpler pure advective model is considered except for the boundary layer, or potential flow models coupled to full Navier-Stokes models to describe e.g. fluid flows around obstacles. Like in the usual DD framework, coupling conditions give rise to an interface problem (governed by the Steklov-Poincaré operator), which can be solved using classical tools derived from optimal control problems (giving rise to the so-called *virtual control approach* [15, 18]). These techniques are suitable also for treating multiphysics problems, such as the coupling between hyperbolic and elliptic equations for boundary layers [18], or the Darcy-Stokes coupling for fluid flows through porous media [4].

The structure of the paper is as follows. We illustrate in Sect. 2 some features shared by several computational reduction approaches, focusing on the reduced basis-like techniques. Hence, we describe in Sect. 3 the most popular methods for computing snapshots and constructing reduced basis: greedy algorithms and proper orthogonal decomposition. The former is the kernel of reduced basis methods for parametrized PDEs, which are detailed in Sect. 4. Moreover, we provide a brief insight on surrogate models in Sect. 5. In the end, we discuss a simple application within the reduced basis framework and provide a comparison with some surrogate models in Sect. 6.

2. Computational reduction: main features

The focus of this paper is on parametrized PDEs where the *input* parameter vector $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p$ might describe physical properties of the system, as well as boundary conditions, source terms or the geometrical configuration. For the sake of space, we only focus on steady parametrized problems, which take the following form:

given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate the output of interest $s(\boldsymbol{\mu}) = J(u(\boldsymbol{\mu}))$ where the solution $u(\boldsymbol{\mu}) \in X = X(\Omega)$ satisfies

$$L(\boldsymbol{\mu})u(\boldsymbol{\mu}) = F(\boldsymbol{\mu}); \quad (2.1)$$

here $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ is a regular domain, X is a suitable Hilbert space, X^* its dual, $L(\boldsymbol{\mu}) : X \rightarrow X^*$ is a second-order differential operator and $F(\boldsymbol{\mu}) \in X^*$. Its weak formulation reads: find $u(\boldsymbol{\mu}) \in X = X(\Omega)$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X(\Omega), \quad (2.2)$$

where the bilinear form¹

$$a(u, v; \boldsymbol{\mu}) := {}_{X^*} \langle L(\boldsymbol{\mu})u, v \rangle_X, \quad \forall u, v \in X, \quad (2.3)$$

is continuous and coercive, i.e. for each $\boldsymbol{\mu} \in \mathcal{D}$:

$$\sup_{u \in X} \sup_{v \in X} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_X \|v\|_X} < +\infty, \quad \exists \alpha_0 > 0 : \inf_{u \in X} \frac{a(u, u; \boldsymbol{\mu})}{\|u\|_X^2} \geq \alpha_0.$$

If the coercivity assumption is not satisfied, stability is in fact fulfilled in the more general sense of the *inf-sup* condition. On the other hand,

$$f(v; \boldsymbol{\mu}) = {}_{X^*} \langle F(\boldsymbol{\mu}), v \rangle_X \quad (2.4)$$

is a continuous linear form. Further assumptions suitable for the effectivity of some reduction strategies will be introduced later on. We may distinguish between two general *paradigms* in computational reduction, that we will qualify as *projection* vs. *interpolation*, yielding the following families of techniques:

1. **Computational Reduction Techniques** (CRT) are *problem-dependent* methods which aim at reducing the dimension of the algebraic system arising from the discretization of a PDE problem. The reduced solution is thus obtained through a projection onto a small subspace made by *global* basis functions, constructed for the specific problem, rather than onto a large space of generic, *local* basis functions (like in finite elements);
2. **Surrogate Response Surfaces** (SRS), also known as *metamodels* or *emulators*, are instead *problem-transparent* methods, which provide an approximation of the input/output map by fitting a set of data obtained by numerical simulation. The PDEs connecting input and output are usually solved by full-order discretization techniques (based e.g. on finite elements). In this review, we will focus on CRTs, while a short description of some SRS methods will be provided in Sect. 5.

The goal of a CRT for PDE problems is to compute, in a cheap way, a low-dimensional approximation of the PDE solution without using a high-fidelity, computationally expensive discretization scheme. The most common choices, like proper

¹In a more rigorous way, we should introduce the Riesz identification operator $R : V^* \rightarrow V$ by which we identify V and its dual, so that, given a third Hilbert space H such that $V \hookrightarrow H^*$ and $H^* \hookrightarrow V^*$, ${}_{X^*} \langle L(\boldsymbol{\mu})u, v \rangle_X = (RL(\boldsymbol{\mu})u, v)_H$. However, the Riesz operator will be omitted for the sake of simplicity.

orthogonal decomposition (POD) or reduced basis (RB) methods, seek for a reduced solution through a *projection* onto suitable low-dimensional subspaces². In particular, the essential components of a CRT can be summarized as follows:

- *High-fidelity discretization technique*: in any case, a CRT is premised upon, and does not replace completely, a high-fidelity (sometimes denoted as *truth*) discretization method, such as the finite element method (FEM), which in the steady case (2.1) reads as: given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate $s_h(\boldsymbol{\mu}) = J(u_h(\boldsymbol{\mu}))$, being $u_h(\boldsymbol{\mu}) \in X_h$ such that

$$L_h(\boldsymbol{\mu})u_h(\boldsymbol{\mu}) = F_h(\boldsymbol{\mu}), \quad (2.5)$$

where $X_h \subset X$ is a finite dimensional space of very large dimension \mathcal{N}_h and $L_h(\boldsymbol{\mu}), F_h(\boldsymbol{\mu})$ are discrete operators. In an abstract way, introducing the projection operators $\Pi_h : X \rightarrow X_h$ and $\Pi_h^* : X^* \rightarrow X_h^*$ onto X_h and X_h^* , respectively, and denoting $u_h(\boldsymbol{\mu}) = \Pi_h u(\boldsymbol{\mu})$, we have

$$\Pi_h^*(L(\boldsymbol{\mu})\Pi_h^{-1}u_h(\boldsymbol{\mu}) - f(\boldsymbol{\mu})) = 0, \quad (2.6)$$

identifying $L_h(\boldsymbol{\mu})$ and $F_h(\boldsymbol{\mu})$ as

$$L_h(\boldsymbol{\mu}) = \Pi_h^*L(\boldsymbol{\mu})\Pi_h^{-1}, \quad F_h(\boldsymbol{\mu}) = \Pi_h^*F(\boldsymbol{\mu}).$$

Equivalently, thanks to (2.3)-(2.4), the weak formulation of (2.6) is:

$$a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}), \quad \forall v_h \in X_h. \quad (2.7)$$

In particular, we assume that

$$\|u(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})\|_X \leq \mathcal{E}(h), \quad \forall \boldsymbol{\mu} \in \mathcal{D},$$

being $\mathcal{E}(h)$ an estimation of the discretization error, which can be made as small as desired by choosing a suitable discretization space. In practice, we will rely on a FEM approximation as *truth* discretization method.

- *(Galerkin) projection*: a CRT usually consists of selecting a reduced basis of few high-fidelity PDE solutions $\{u_h(\boldsymbol{\mu}^i)\}_{i=1}^N$ (called *snapshots*) and seeking a reduced approximation $u_N(\boldsymbol{\mu})$ expressed as a linear combination of the basis functions [7, 37]; the coefficients or *weights* of this combination are determined through a (Galerkin-like) projection of the equations onto the reduced space X_N , being $N = \dim(X_N) \ll \mathcal{N}_h$: given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate $s_N(\boldsymbol{\mu}) = J(u_N(\boldsymbol{\mu}))$, where $u_N(\boldsymbol{\mu}) \in X_N$ solves

$$L_N(\boldsymbol{\mu})u_N(\boldsymbol{\mu}) - F_N(\boldsymbol{\mu}) = 0; \quad (2.8)$$

the smaller is N , the cheaper will be the reduced problem. Equivalently,

$$a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}), \quad \forall v_N \in X_N. \quad (2.9)$$

²Indeed, we remark that several CRTs, like POD, have been originally introduced and developed in order to speed-up the solution of very complex time-dependent problems, like for turbulent flows, without being addressed to parametrized problems (i.e. time was considered as the only parameter).

As before, introducing the projectors onto the reduced space X_N and its dual X_N^* , $\Pi_N : X_h \rightarrow X_N$ and $\Pi_N^* : X_h^* \rightarrow X_N^*$, we have

$$\Pi_N^*(L_h(\boldsymbol{\mu})\Pi_N^{-1}u_N(\boldsymbol{\mu}) - F_h(\boldsymbol{\mu})) = 0, \quad (2.10)$$

so that we can identify

$$L_N(\boldsymbol{\mu}) = \Pi_N^*L_h(\boldsymbol{\mu})\Pi_N^{-1}, \quad F_N(\boldsymbol{\mu}) = \Pi_N^*f_h(\boldsymbol{\mu}).$$

Two possible strategies for sampling the parameter space and constructing the corresponding snapshots will be discussed in Sect. 3.1 and 3.2. Moreover, an algebraic perspective of projection/reduction stages based on matrix computation will be presented in Sect. 4, after introducing the reduced basis method formalism.

- *Offline/Online procedure*: under suitable assumptions (see Sect. 4) the extensive generation of the snapshots database can be performed *Offline* once, and is completely decoupled from each new subsequent input-output *Online* query [37]. Clearly, during the *Online* stage, the reduced problem can be solved for parameter instances $\boldsymbol{\mu} \in \mathcal{D}$ not selected during the *Offline* stage, and even extrapolating the solution for values $\boldsymbol{\mu} \in \mathcal{D}_{ext}$ belonging to a parameter superset $\mathcal{D}_{ext} \supseteq \mathcal{D}$ [9].

Not only, the expensive *Offline* computations have to be amortized over the *Online* stage; for instance, in the reduced basis context the break-even point is usually reached with $\mathcal{O}(10^2)$ online queries.

- *Error estimation procedure*: sharp, inexpensive bounds $\Delta_N(\boldsymbol{\mu})$ such that

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X \leq \Delta_N(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \quad N = 1, \dots, N_{max},$$

may be available [37], as well as output error bounds such that $|s_h(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})| \leq \Delta_N^s(\boldsymbol{\mu})$. These error estimators might also be employed for a *clever* parameter sampling during the construction of the reduced space. For the sake of brevity, we do not discuss the construction of *a posteriori* error estimators, extensively used within the reduced basis context: the interested reader may refer e.g. to [45, 49] and to references therein.

We remark that CRTs do not replace, but rather are built upon – and measured (as regards accuracy) relative to – a high-fidelity discretization technique, so that an algorithmic collaboration is pursued, expressed simply by means of a triangular inequality as follows, for all $\boldsymbol{\mu} \in \mathcal{D}$,

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X \leq \|u(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})\|_X + \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X \leq \mathcal{E}(h) + \Delta_N(\boldsymbol{\mu}).$$

3. Construction of reduced spaces

3.1. Greedy algorithms

A well-known strategy for constructing reduced subspaces is that of using *greedy* algorithms, based on the idea of selecting at each step the locally optimal element. In

an abstract setting, given a compact set \mathcal{K} in a Hilbert space \mathcal{X} , we seek for functions $\{f_0, f_1, \dots, f_{N-1}\}$ such that each $f \in \mathcal{K}$ is well approximated by the elements of the subspace $\mathcal{K}_N = \text{span}\{f_0, \dots, f_{N-1}\}$; starting from a first element $f_0 \in \mathcal{K}$ such that $\|f_0\|_{\mathcal{X}} = \max_{f \in \mathcal{K}} \|f\|_{\mathcal{X}}$, at the N -th step a greedy algorithm selects

$$f_N = \arg \max_{f \in \mathcal{K}} \|f - \Pi_N^{RB} f\|_{\mathcal{X}},$$

being Π_N^{RB} the orthogonal projection w.r.t. the scalar product inducing the norm $\|\cdot\|_{\mathcal{X}}$ onto \mathcal{K}_N . Hence f_N is the worst case element, which maximizes the error in approximating the subspace \mathcal{K} using the elements of \mathcal{K}_N . A more feasible variant of this algorithm – called *weak greedy algorithm* in [3] – replaces the true error $\|f - \Pi_N^{RB} f\|_{\mathcal{X}}$ by a surrogate $\eta_N(f)$ (in our case, the *a posteriori* error bound) satisfying

$$c_\eta \eta_N(f) \leq \|f - \Pi_N^{RB} f\|_{\mathcal{X}} \leq C_\eta \eta_N(f), \quad f \in \mathcal{X}.$$

In this way, $f_N = \arg \max_{f \in \mathcal{K}} \eta_N(f)$ can be computed more effectively, under the assumption that the surrogate error is cheap to evaluate. We refer the reader to [3] for more details and some results on convergence rates of these algorithms. In particular, the current procedure for constructing reduced subspaces in parametrized PDEs like (2.1) is based on the following *weak greedy* algorithm. For the sake of space, we bound ourselves to the case of time-independent problems. Moreover, we denote the particular samples which shall serve to select the RB space – or “train” the RB approximation – by Ξ_{train} , its cardinality by $|\Xi_{\text{train}}| = n_{\text{train}}$ and by $\varepsilon_{\text{tol}}^*$ a chosen tolerance for the stopping criterion of the algorithm. For a generic element $z: \mathcal{D} \rightarrow X^{\mathcal{N}}$, we denote

$$\|z\|_{L^\infty(\Xi_{\text{train}}; X)} \equiv \text{ess sup}_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \|z(\boldsymbol{\mu})\|_X.$$

Starting from $S_1 = \{\boldsymbol{\mu}^1\}$, we adopt the following procedure:

```

 $S_1 = \{\boldsymbol{\mu}^1\}$ ; compute  $u_h(\boldsymbol{\mu}^1)$ ;  $X_1^{RB} = \text{span}\{u_h(\boldsymbol{\mu}^1)\}$ ;
for  $N = 2 : N_{\text{max}}$ 
     $\boldsymbol{\mu}^N = \arg \max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_{N-1}(\boldsymbol{\mu})$ ;
     $\varepsilon_{N-1} = \Delta_{N-1}(\boldsymbol{\mu}^N)$ ;
    if  $\varepsilon_{N-1} \leq \varepsilon_{\text{tol}}^*$ 
         $N_{\text{max}} = N - 1$ ;
    end;
    compute  $u_h(\boldsymbol{\mu}^N)$ ;
     $S_N = S_{N-1} \cup \{\boldsymbol{\mu}^N\}$ ;
     $X_N^{RB} = X_{N-1}^{RB} \cup \text{span}\{u_h(\boldsymbol{\mu}^N)\}$ ;
end.
```

As already mentioned in Sect. 2, $\Delta_N(\boldsymbol{\mu})$ is a sharp, *inexpensive a posteriori* error bound for $\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X$. Hence, using the weak greedy algorithm only FE solutions corresponding to the selected snapshots have to be computed; instead, the

pure version of the greedy algorithm would select

$$\boldsymbol{\mu}^N = \arg \operatorname{ess\,sup}_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \|u_h(\boldsymbol{\mu}) - u_{N-1}(\boldsymbol{\mu})\|_X,$$

i.e. the element maximizing the true error $\|u_h(\boldsymbol{\mu}) - u_{N-1}(\boldsymbol{\mu})\|_{L^\infty(\Xi_{\text{train}}; X)}$, and would entail a much higher computational cost, requiring at each step the computation of n_{train} FE solutions. Greedy algorithms have been applied in several contexts, involving also other reduction issues; some recent applications deal e.g. with a simultaneous parameter and state reduction, in problems requiring the exploration of high-dimensional parameter spaces [6, 29].

3.2. Alternative approaches

Another technique used for the construction of reduced spaces in computational reduction of parametrized systems is proper orthogonal decomposition (POD), a very popular approach used in several different fields such as multivariate statistical analysis (where it is called *principal component analysis*) or theory of stochastic processes (*Karhunen-Loève decomposition*). The first applications of POD were concerned with the analysis of turbulent flows and date back to the early '90s [1, 2]; more recent applications can be found, for instance, in [22, 25, 27, 30], as well as in [9, 12, 20] for parametrized flows.

POD techniques reduce the dimensionality of a system by transforming the original variables onto a new set of uncorrelated variables (that are called POD modes, or principal components) such that the first few modes retain most of the *energy* present in all of the original variables. This allows to obtain a reduced, modal representation through a spectral decomposition which requires basic matrix computations (a *singular value decomposition*) also for complex nonlinear problems. However, no a posteriori estimations for the error between the reduced and the full-order approximations are in general available, making the choice of the reduction size and the quality assessment of the reduced solution sometimes critical. Moreover, (ii) space reduction through spectral decomposition entails in general very large computational costs.

We shortly review the main features of the POD³ in the context of parametrized PDEs. Given a finite sample Ξ of points in \mathcal{D} , a train sample Ξ_{train} (which shall serve to select the POD space), for a generic element $z: \mathcal{D} \rightarrow X^N$, we denote

$$\|z\|_{L^2(\Xi; X)} \equiv \left(|\Xi|^{-1} \sum_{\boldsymbol{\mu} \in \Xi} \|z(\boldsymbol{\mu})\|_X^2 \right)^{1/2}.$$

The POD method seeks an N -dimensional subspace $X_N^{\text{POD}} \subset X_h$ approximating the data in an optimal *least-squares* sense; thus, we seek an orthogonal projector $\Pi_N^{\text{POD}}: X_h \rightarrow X_N^{\text{POD}}$, of prescribed rank N , as follows:

$$X_N^{\text{POD}} = \arg \inf_{X_N^{\text{POD}} \subset \operatorname{span}\{y^N(\boldsymbol{\mu}), \boldsymbol{\mu} \in \Xi_{\text{train}}\}} \|y^N(\boldsymbol{\mu}) - \Pi_N^{\text{POD}} y^N(\boldsymbol{\mu})\|_{L^2(\Xi_{\text{train}}; X)}. \quad (3.1)$$

³For a general and synthetic introduction to POD techniques in view of the reduction of a (time-dependent) dynamical system – which is the first (and most used) application of this strategy – the interested reader may refer to [41, 56].

Following the so-called *method of snapshots*, introduced by Sirovich [53], we compute the n_{train} full-order approximations $\{y(\boldsymbol{\mu}^m)\}_{m=1}^{n_{\text{train}}}$ corresponding to $\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^{n_{\text{train}}}$, the mean

$$\bar{y} = \frac{1}{n_{\text{train}}} \sum_{j=1}^{n_{\text{train}}} y(\boldsymbol{\mu}^j)$$

and the *correlation* matrix $\mathbb{C} \in \mathbb{R}^{n_{\text{train}} \times n_{\text{train}}}$ whose components are

$$\mathbb{C}_{ij} = \frac{1}{n_{\text{train}}} \sum_{m=1}^{n_{\text{train}}} (y(\boldsymbol{\mu}^i) - \bar{y}, y(\boldsymbol{\mu}^j) - \bar{y})_X, \quad 1 \leq i, j \leq n_{\text{train}}.$$

Then, we compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n_{\text{train}}} \geq 0$ (ordered by decreasing size) and the eigenvectors of the correlation matrix, which solve:

$$\mathbb{C}\boldsymbol{\psi}_k = \lambda_k \boldsymbol{\psi}_k, \quad k = 1, \dots, n_{\text{train}}.$$

The central result of POD states that the *optimal* subspace X_N^{POD} of dimension N minimizing (3.1) is given by

$$X_N^{\text{POD}} = \text{span}\{\zeta_n, 1 \leq n \leq N\}, \quad 1 \leq N \leq N_{\text{max}},$$

where the POD basis functions are defined as

$$\zeta_k = \frac{\tilde{\zeta}_k}{\|\tilde{\zeta}_k\|_X}, \quad \tilde{\zeta}_k = \sum_{m=1}^{n_{\text{train}}} \psi_{k,m} (y(\boldsymbol{\mu}^m) - \bar{y}), \quad 1 \leq k \leq n_{\text{train}},$$

being $\psi_{k,m} = (\boldsymbol{\psi}_k)_m$ the m -th component of the k -th eigenvector. In this way, the basis functions $\{\zeta_k\}_{k=1}^{n_{\text{train}}}$ are orthonormal, i.e. they are such that $(\zeta_n, \zeta_m)_X = \delta_{n,m}$, for $1 \leq n, m \leq n_{\text{train}}$. In particular, N_{max} is chosen as the smallest N such that $\varepsilon_N^{\text{POD}} = (\sum_{k=N+1}^{n_{\text{train}}} \lambda_k)^{1/2} \leq \epsilon_{\text{tol}}^*$, i.e. the energy $\varepsilon_N^{\text{POD}}$ retained by the last $n_{\text{train}} - N_{\text{max}}$ modes is negligible. Typically, this POD approach is much more expensive than the Greedy approach: in the latter, we only need to compute the N – typically very few – FE *retained* snapshots; in the POD approach, we must compute all n_{train} – typically/desirably very many – FE *candidate* snapshots, as well as the solution of an eigenproblem for the correlation matrix $\mathbb{C} \in \mathbb{R}^{n_{\text{train}} \times n_{\text{train}}}$.

Two additional techniques – indeed quite close to POD – for generating reduced spaces are the Centroidal Voronoi Tessellation [7–9] and the Proper Generalized Decomposition [10, 11, 16, 36].

We remark that the current approach for constructing reduced basis approximations of time-dependent parametrized PDEs exploits a combined POD-greedy procedure – POD in time to capture the causality associated with the evolution equation, greedy procedure for sampling the parameter space and treat more efficiently extensive ranges of parameter variation (see e.g. [21, 45]).

4. Reduced Basis Methods for parametrized PDEs

In this section we illustrate with more detail the general features presented in Sect. 2 in the case of *reduced basis methods* for parametrized PDEs, focusing on the steady

case. Reduced Basis (RB) discretization is, in brief, a Galerkin projection on an N -dimensional approximation space that focuses on the parametrically induced manifold $\mathcal{M}_h = \{u_h(\boldsymbol{\mu}) \in X_h : \boldsymbol{\mu} \in \mathcal{D}\}$. We restrict our attention to the Lagrange RB spaces, which are based on the use of “snapshot” FE solutions of the PDE, and review the construction of the RB approximation in the elliptic case. Moreover, we make the *ansatz* that the manifold \mathcal{M}_h given by the set of fields engendered as the input varies over the parameter domain \mathcal{D} , is sufficiently smooth.

In order to define a (hierarchical) sequence of Lagrange spaces X_N , $1 \leq N \leq N_{max}$, such that $X_1 \subset X_2 \subset \dots \subset X_{N_{max}} \subset X$, we first introduce a “master set” of properly selected parameter points $\boldsymbol{\mu}^n \in \mathcal{D}$, $1 \leq n \leq N_{max}$, and define, for given $N \in \{1, \dots, N_{max}\}$, the Lagrange parameter samples

$$S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}, \quad (4.1)$$

and associated Lagrange greedy-RB spaces

$$X_N^{RB} = \text{span}\{u_h(\boldsymbol{\mu}^n), 1 \leq n \leq N\}, \quad (4.2)$$

assembled by means of the *greedy* procedure presented in Sect. 3.1; in the rest of the section the superscript RB will be often omitted for clarity.

As already mentioned, the RB approximation of the PDE solution can be expressed as follows: given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate $s_N(\boldsymbol{\mu}) = J(u_N(\boldsymbol{\mu}))$, where $u_N(\boldsymbol{\mu}) \in X_N := X_N^{RB} \subset X_h$ satisfies

$$a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N), \quad \forall v_N \in X_N. \quad (4.3)$$

We immediately obtain the classical optimality result:

$$\| \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\| \|_{\boldsymbol{\mu}} \leq \inf_{w \in X_N} \| \|u_h(\boldsymbol{\mu}) - w\| \|_{\boldsymbol{\mu}}, \quad (4.4)$$

i.e. in the energy norm⁴ the Galerkin procedure automatically selects the *best* combination of snapshots; moreover, we have that

$$s_h(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) = \| \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\| \|_{\boldsymbol{\mu}}^2, \quad (4.5)$$

i.e. the output converges as the “square” of the energy error.

We now consider the discrete equations associated with the Galerkin approximation (4.3). First of all, we apply the Gram-Schmidt process with respect to the $(\cdot, \cdot)_X$ inner product to the snapshots $u_h(\boldsymbol{\mu}^n)$, $1 \leq n \leq N$, to obtain mutually $(\cdot, \cdot)_X$ -orthonormal basis functions ζ_n , $1 \leq n \leq N$. Then, the RB solution can be expressed as:

$$u_N(\boldsymbol{\mu}) = \sum_{m=1}^N u_{Nm}(\boldsymbol{\mu}) \zeta_m; \quad (4.6)$$

by plugging this expression in (4.3) and choosing $v = \zeta_n^N$, $1 \leq n \leq N$, we obtain the RB “stiffness” equations

⁴The energy norm $\| \| \cdot \| \|_{\boldsymbol{\mu}}$ is defined, for all $v \in X$, as $\| \|v\| \|_{\boldsymbol{\mu}} = a(v, v; \boldsymbol{\mu})$, provided that $a(\cdot, \cdot; \boldsymbol{\mu})$ is a symmetric and coercive bilinear form. The corresponding scalar product will be denoted as $((\cdot, \cdot))_{\boldsymbol{\mu}}$.

$$\sum_{m=1}^N a(\zeta_m, \zeta_n; \boldsymbol{\mu}) u_{Nm}(\boldsymbol{\mu}) = f(\zeta_n), \quad (4.7)$$

for the RB coefficients $u_{Nm}(\boldsymbol{\mu})$, $1 \leq m, n \leq N$; the RB output can be subsequently evaluated as

$$s_N(\boldsymbol{\mu}) = \sum_{m=1}^N u_{Nm}(\boldsymbol{\mu}) l(\zeta_m). \quad (4.8)$$

Although the system (4.7) is nominally of small size, yet it involves entities ζ_n , $1 \leq n \leq N$, associated with our \mathcal{N}_h -dimensional FE approximation space. Fortunately, a strong computational speedup can be achieved by making the crucial assumption of *affine parametric dependence*. The linear/bilinear forms can be expressed as a linear combination

$$a(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(w, v), \quad f(w; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) f^q(w) \quad (4.9)$$

for some finite Q_a, Q_f , where $\Theta_a^q : \mathcal{D} \rightarrow \mathbb{R}$, $1 \leq q \leq Q_a$, $\Theta_f^q : \mathcal{D} \rightarrow \mathbb{R}$, $1 \leq q \leq Q_f$ are smooth scalar functions depending on $\boldsymbol{\mu}$, and a^q , $1 \leq q \leq Q_a$, f^q , $1 \leq q \leq Q_f$ are bilinear/linear forms independent of $\boldsymbol{\mu}$. This property is in fact not exotic, actually it is quite naturally fulfilled in many kinds of applications in science and engineering. Under this assumption, (4.7)-(4.8) can be rewritten as

$$\left(\sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbb{A}_N^q \right) \mathbf{u}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) \mathbf{f}_N^q, \quad (4.10)$$

$$s_N(\boldsymbol{\mu}) = \mathbf{l}_N \mathbf{u}_N(\boldsymbol{\mu}), \quad (4.11)$$

where $(\mathbf{u}_N(\boldsymbol{\mu}))_m = u_{Nm}(\boldsymbol{\mu})$ and, for $1 \leq m, n \leq N$,

$$(\mathbb{A}_N^q)_{mn} = a^q(\zeta_m, \zeta_n), \quad (\mathbf{f}_N^q)_n = f^q(\zeta_n), \quad (\mathbf{l}_N)_n = l(\zeta_n).$$

Since each basis function ζ_n belongs to the FE space X_h , they can be written as a linear combination of the FE basis functions $\{\phi_i^h\}_{i=1}^{\mathcal{N}}$:

$$\zeta_n = \sum_{i=1}^{\mathcal{N}} \zeta_{ni} \phi_i^h, \quad 1 \leq n \leq N;$$

in matrix form, the basis can be represented as an orthonormal matrix

$$\mathcal{Z}_N := \mathcal{Z}_N^{RB} = [\boldsymbol{\zeta}_1 \mid \dots \mid \boldsymbol{\zeta}_N] \in \mathbb{R}^{\mathcal{N} \times N}, \quad 1 \leq N \leq N_{max},$$

being $(\mathcal{Z}_N)_{jn} = \zeta_{nj}$ for $1 \leq n \leq N$, $1 \leq j \leq \mathcal{N}$. Therefore, the RB ‘‘stiffness’’ matrix can be assembled once the corresponding FE ‘‘stiffness’’ matrix has been computed, and can be obtained as

$$\mathbb{A}_N^q = \mathcal{Z}_N^T \mathbb{A}_h^q \mathcal{Z}_N, \quad \mathbf{f}_N^q = \mathcal{Z}_N^T \mathbf{f}_h^q, \quad \mathbf{l}_N = \mathcal{Z}_N^T \mathbf{l}_h \quad (4.12)$$

being

$$(\mathbb{A}_h^q)_{ij} = a^q(\phi_j, \phi_i), \quad (\mathbf{f}_h^q)_i = f^q(\phi_i), \quad (\mathbf{l}_h)_i = l(\phi_i) \quad (4.13)$$

the FE algebraic structures. In this way, computation entails an expensive $\boldsymbol{\mu}$ -independent Offline stage performed only once and a very inexpensive Online stage for any chosen parameter value $\boldsymbol{\mu} \in \mathcal{D}$. During the former the FE structures $\{\mathbb{A}_h^q\}_{q=1}^{Q_a}$, $\{\mathbf{f}_h^q\}_{q=1}^{Q_f}$, \mathbf{l}_h , as well as the snapshots $\{u(\boldsymbol{\mu}^n)\}_{n=1}^{N^{max}}$ and the corresponding orthonormal basis $\{\zeta_n\}_{n=1}^{N^{max}}$, are computed and stored. In the latter, for any given $\boldsymbol{\mu}$, all the $\Theta_q^a(\boldsymbol{\mu})$, $\Theta_q^f(\boldsymbol{\mu})$ functions are evaluated, and the $N \times N$ linear system (4.10) is assembled and solved, in order to get the RB approximation $u_N(\boldsymbol{\mu})$. Then, the RB output approximation is obtained through the simple scalar product (4.8).

Although being dense (rather than sparse as in the FE case), the system matrix is very small, with a size independent of the FE space dimension \mathcal{N}_h .

A general formulation of RB methods can be found for example in [45,49]. Early applications to problems arising in computational fluid dynamics are described in [24, 43], while recent applications dealing with Stokes and Navier-Stokes equations are presented e.g. in [14,34,44,50,54]. More recent extensions to flow control problems by shape optimization can be found in [31,32]. The reader interested in time-dependent problems can instead refer to [19,21] or to the more recent review provided in [45].

We close this section by pointing out the connection between the RB approximation and the FE approximation from an algebraic standpoint. Let us denote by $\mathbf{u}_h \in \mathbb{R}^{\mathcal{N}}$ and $\mathbf{u}_N \in \mathbb{R}^N$ the vectors of degrees of freedom of the FEM and of the RB approximation, associated to the functions $u_h \in X_h$ and $u_N \in X_N$, respectively. Moreover, let $\mathbb{A}_h(\boldsymbol{\mu})$ and $\mathbb{A}_N(\boldsymbol{\mu})$ be the matrices corresponding to the FEM and to the RB discretization, respectively, for any given parameter value $\boldsymbol{\mu} \in \mathcal{D}$. From the relationships discussed in this section, the reduced linear system (4.10) can be rewritten as

$$\mathbb{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu}), \quad (4.14)$$

whereas the full-order FEM linear system would read

$$\mathbb{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}). \quad (4.15)$$

In order to make a connection between the RB and the FE linear systems, we can express (without considering a basis orthonormalization)

$$\mathbf{u}_h(\boldsymbol{\mu}) = \mathcal{Z}_N(\mathbf{u}_N(\boldsymbol{\mu}) + \boldsymbol{\delta}_N(\boldsymbol{\mu})), \quad (4.16)$$

where the error term $\boldsymbol{\delta}_N \in \mathbb{R}^N$ accounts for the fact that $\mathcal{Z}_N\mathbf{u}_N$ is not the exact solution of the full-order system and *a priori* is not vanishing.

By plugging the expression (4.16) into (4.15) and multiplying the system by \mathcal{Z}_N^T , we obtain

$$\mathcal{Z}_N^T \mathbb{A}_h(\boldsymbol{\mu}) \mathcal{Z}_N(\mathbf{u}_N(\boldsymbol{\mu}) + \boldsymbol{\delta}_N(\boldsymbol{\mu})) = \mathcal{Z}_N^T \mathbf{f}_h(\boldsymbol{\mu}).$$

Thanks to (4.13) and to (4.14), we thus find that

$$\mathbb{A}_N(\boldsymbol{\mu})\boldsymbol{\delta}_N(\boldsymbol{\mu}) = \mathbf{0},$$

i.e. the algebraic counterpart of the *Galerkin orthogonality* property, fulfilled by the RB approximation, that is:

$$a(u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}), v_N) = 0, \quad \forall v_N \in X_N.$$

On the other hand, setting $\boldsymbol{\delta}_h(\boldsymbol{\mu}) = \mathcal{Z}_N \boldsymbol{\delta}_N(\boldsymbol{\mu})$, we have

$$\mathbf{u}_h(\boldsymbol{\mu}) = \mathcal{Z}_N \mathbf{u}_N(\boldsymbol{\mu}) + \boldsymbol{\delta}_h(\boldsymbol{\mu}), \quad (4.17)$$

where now the error term $\boldsymbol{\delta}_h \in \mathbb{R}^N$ is represented in the reduced vector space \mathbb{R}^N . Plugging (4.17) into (4.15), we end up with

$$\mathbb{A}_h(\boldsymbol{\mu}) \boldsymbol{\delta}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbb{A}_h(\boldsymbol{\mu}) \mathcal{Z}_N \mathbf{u}_N(\boldsymbol{\mu}),$$

which is the algebraic counterpart of the error residual relationship:

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = r(v; \boldsymbol{\mu}), \quad \forall v \in X^N. \quad (4.18)$$

being $e(\boldsymbol{\mu}) := u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}) \in X^N$ the (reduced) approximation error, $r(v; \boldsymbol{\mu}) \in (X^N)'$ the residual, given by

$$r(v; \boldsymbol{\mu}) := f(v; \boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \quad \forall v \in X^N. \quad (4.19)$$

Together with a lower bound of the coercivity constant, equation (4.18) is a basic ingredient for *a posteriori* error bounds; see [45, 49] for further details.

5. Surrogate models: response surfaces and kriging

A different strategy to speedup numerical output evaluations related to parametrized systems can be based on suitable *data-driven, problem-transparent* methods, without attempting to reduce the computational cost related to PDE discretization. Surrogate models provide mathematical and statistical techniques for the approximation of an input/output relationship (e.g. implied by a computer simulation model). In *data-fitting* models, an approximation of the global behavior of an output with respect to the input variables is constructed using available data, produced by the numerical simulation of a full-order model. For these models the goal is twofold, since we might be interested either in *(i)* finding an explanation of the behavior of the simulation model (in terms of a functional relationship between the output and the input variables) or in *(ii)* evaluating the prediction of the expected simulation output for *scenarios* (or combination of input values, or factors) that have not yet been simulated. The final goals of surrogate models may be for instance validation and verification of the numerical model, sensitivity analysis and/or optimization. A review of prediction methodologies for analysis of computer experiments and their main features can be found e.g. in [51, 55].

5.1. Response surfaces

One of the most common surrogate models is the polynomial response surface method (RSM), which is based on low-order *polynomial regression* and aims at representing the output (or *response*) surface as a polynomial function of the input parameters. Another class of techniques, more suitable than low-order polynomial regression for data fitting in wider parameter spaces, are the so-called *kriging methods*. We present in this section the former, focusing on the simplest case of a single (univariate, scalar) simulation output w , which can be expressed in general as a

function of p input variables:

$$w = s(\mu_1, \dots, \mu_p);$$

here $s : \mathbb{R}^p \rightarrow \mathbb{R}$ denotes the function implicitly defined by the numerical approximation of the output w ; $\mathbb{D} = (\mu_{ij})$ denotes the design matrix for the numerical simulation experiment, with $j = 1, \dots, p$ and $i = 1, \dots, n$, being n the number of input combinations evaluated during the experiment. If the response is well-modeled by a linear function of the input variables, the approximating function is a *first-order* polynomial regression model:

$$w_{reg,1}(\boldsymbol{\mu}) = \beta_0 + \sum_{i=1}^p \beta_i \mu_i + \varepsilon_{reg},$$

where $\boldsymbol{\beta} = (\beta_0, \dots, \beta_p) \in \mathbb{R}^{p+1}$ is the parameter vector of the surrogate model and ε_{reg} is the error term including the lack of fit of the surrogate model. In matrix form, considering the n combinations evaluated on the lines, we have

$$\mathbf{w}_{reg,1} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_{reg},$$

being $\mathbb{X} = [\mathbf{1} \mid \mathbb{D}] \in \mathbb{R}^{n \times (p+1)}$, $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^n$, and $\boldsymbol{\varepsilon}_{reg} \in \mathbb{R}^n$ the vector of the residuals in the n input combinations. The estimation $\hat{\boldsymbol{\beta}}$ of the parameters of the model is usually obtained through the *least squares* method, giving $\hat{\boldsymbol{\beta}} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X} \mathbf{w}$, where $\mathbf{w} = (w_1, \dots, w_n) \in \mathbb{R}^n$ is the vector containing the n output values obtained through the numerical simulations corresponding to the input combinations \mathbb{D} . Hence, for a new input combination $\boldsymbol{\mu}_{new}$, the response surface prediction is given by $\hat{w}_{reg,1}(\boldsymbol{\mu}_{new}) = \mathbf{x}_{new}^T \hat{\boldsymbol{\beta}} = \mathbf{x}_{new}^T (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X} \mathbf{w}$, being $\mathbf{x}_{new} = (1; \boldsymbol{\mu}_{new})^T$; the response surface analysis is then performed using the fitted surface. Several design optimality criteria are available for choosing the input combinations \mathbb{X} where we intend to simulate the output values, on the basis of which the response surface is fitted, in order to get an accurate estimation $\hat{\boldsymbol{\beta}}$ (in terms of minimum variance); a popular option is based for instance on the minimization of $|\det(\mathbb{X}^T \mathbb{X})^{-1}|$. If there is a *curvature* effect in the system, then a polynomial of higher degree must be used, such as a *second-order* polynomial regression model:

$$w_{reg,2}(\boldsymbol{\mu}) = \beta_0 + \sum_{i=1}^p \beta_i \mu_i + \sum_{i=1}^p \beta_{ii} \mu_i^2 + \sum_{\substack{i=1, j=1 \\ i < j}}^p \beta_{ij} \mu_i \mu_j + \varepsilon_{reg},$$

for which the analysis can be performed in the same way. In the case where the output function $s(\cdot)$ has several local maxima and minima, and the problem involves high dimensions and/or scattered data in the parameter space, radial basis functions (RBF) have been found very accurate to generate response surface models [33]. An interpolation model based on RBFs is a linear combination of the form

$$w_{rbf}(\boldsymbol{\mu}) = \sum_{i=1}^n \gamma_i \phi(\|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|),$$

where $\phi(r)$ is a function depending on the radial distance $r = \|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n)^T$ is a vector of coefficients determined from the interpolation conditions $w_{rbf}(\boldsymbol{\mu}_i) = s(\boldsymbol{\mu}_i)$, i.e. the RBF surface matches the output function $s(\cdot)$ at all data points $\boldsymbol{\mu}_i$. The equivalent matrix formulation is

$$\mathbb{P}\boldsymbol{\gamma} = \mathbf{s},$$

where $\mathbf{s} = (s(\boldsymbol{\mu}_1), \dots, s(\boldsymbol{\mu}_n))^T$ and $(\mathbb{P})_{ij} = \phi(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|)$. Typical choices for the radial basis $\phi(r)$ are, for example, the *thin plate spline* $\phi(r) = r^2 \ln(r)$, the Gaussian function $\phi(r) = \exp(-cr^2)$, the multiquadric function $\phi(r) = \sqrt{r^2 + c^2}$, where $c > 0$ is a chosen scaling constant. Other options, such as a combination of low-order polynomial models and RBFs, may result more convenient depending on the problem to be solved (see e.g. [5]).

5.2. Kriging models

Although the polynomial RSM provide in general an acceptable trend of the global behavior of the response, they might fail in capturing local minima or maxima. Kriging models – which can be seen as a further generalization of the RSM based on low-order polynomial regression – provide better results for nonlinear prediction in multi-dimensional parameter domains involving a more complex response behavior. In particular, these models yield the best linear unbiased prediction in (a given, in our case input parameter) space using observations taken at known nearby locations. A general reference is given by [13], while a compact survey of these techniques can be found, for instance, in [26]. Within this class of models, we treat the output of some numerical experiment as a realization from a stochastic process, i.e.

$$w(\boldsymbol{\mu}) = m + \delta(\boldsymbol{\mu}), \quad \boldsymbol{\mu} \in \mathcal{D}$$

where $\delta(\boldsymbol{\mu})$ is a zero-mean stochastic process ($\mathbb{E}(\delta(\boldsymbol{\mu})) = 0$) with known covariance function C such that

$$C_{ii} = \text{var}(\delta(\boldsymbol{\mu}_i)) = \sigma^2, \quad C_{ij} = \text{cov}(\delta(\boldsymbol{\mu}_i), \delta(\boldsymbol{\mu}_j)) = \sigma^2 \rho(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|),$$

being the covariance $C_{ij} = \text{cov}(\delta(\boldsymbol{\mu}_i), \delta(\boldsymbol{\mu}_j))$ dependent only on the difference $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$ – i.e. $\rho(\delta(\boldsymbol{\mu}_i), \delta(\boldsymbol{\mu}_j)) = \rho(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|)$. If we assume that m is known, the *simple kriging* predictor can be obtained as the linear predictor

$$w_{sk}(\boldsymbol{\mu}) = k + \sum_{i=1}^n l_i w(\boldsymbol{\mu}_i)$$

of $w(\bar{\boldsymbol{\mu}})$ (output at an unexperimented combination $\bar{\boldsymbol{\mu}}$) minimizing (w.r.t. $\mathbf{l} = (l_1, \dots, l_n)$) the mean-squared prediction error $\mathbb{E}[(w(\bar{\boldsymbol{\mu}}) - w_{sk}(\boldsymbol{\mu}))^2]$, being $\{w(\boldsymbol{\mu}_i)\}_{i=1}^n$ the computed values of the output. This gives $\mathbf{l} = \bar{\mathbf{c}}C^{-1}$ and $k = (1 - \bar{\mathbf{c}}C^{-1}\mathbf{1})m$, where $(\bar{\mathbf{c}})_i = (\text{cov}(\delta(\bar{\boldsymbol{\mu}}), \delta(\boldsymbol{\mu}_i)))$.

If m is unknown (*ordinary kriging*), the previous expression is no longer a predictor; one possibility is to restrict the solution to the class of homogeneous linear predictors

$$w_{ok}(\boldsymbol{\mu}) = \sum_{i=1}^n \lambda_i w(\boldsymbol{\mu}_i), \quad \text{s.t.} \quad \sum_{i=1}^n \lambda_i = 1$$

and to look for the best linear unbiased predictor obtained by minimizing

$$\hat{\boldsymbol{\lambda}} = \arg \min_{\boldsymbol{\lambda}} \mathbb{E}[(w(\bar{\boldsymbol{\mu}}) - w_{ok}(\boldsymbol{\mu}))^2].$$

In any case, the weights λ_i are not constant (whereas the $\boldsymbol{\beta}$ coefficients in RSM models are) but decrease with the distance between the input $\boldsymbol{\mu}$ to be predicted and the input combinations $\{\boldsymbol{\mu}_i\}_{i=1}^n$ used in the numerical experiment. The kriging weights $\boldsymbol{\lambda}$ obviously depend on the correlations

$$\rho(w(\boldsymbol{\mu}_r), w(\boldsymbol{\mu}_s)) = \prod_{j=1}^p \rho(h_j(r, s)), \quad h_j(r, s) = |\mu_{rj} - \mu_{sj}|$$

with $j = 1, \dots, p$, $r, s = 1, \dots, n$, between the simulation outputs in the kriging model; usual choices are exponential or Gaussian covariance functions, under the form $\rho(h_j(r, s)) = \exp(-\theta_j h_j^\alpha(r, s))$, being θ_j a measure of the importance of the input $\boldsymbol{\mu}_j$ (the higher θ_j is, the less effect input j has) and $\alpha = 1, 2$ for the exponential or the Gaussian case, respectively. In particular, by construction the kriging predictor is uniformly unbiased, i.e. $w_{ok}(\boldsymbol{\mu}_i) = w(\boldsymbol{\mu}_i)$, $\forall i = 1, \dots, n$.

More general models (*universal kriging*) employ a regression model for estimating m ; e.g. for a *first-order* regression model we consider the expression

$$w_{uk}(\boldsymbol{\mu}) = \beta_0 + \sum_{i=1}^p \beta_i \mu_i + \sum_{i=1}^n \lambda_i w(\boldsymbol{\mu}_i),$$

where the weights $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ are obtained (as before) as the generalized least squares solution, and depend on the covariance function; an optimal expression for the coefficients θ_j in the covariance function can also be computed.

6. A simple application of interest

We present in this section an application of the RB method to a simple problem of interest in ideal computational fluid dynamics – the description of the flow around parametrized airfoils – as well as some comparisons between the RB results and those obtained by applying the surrogate models of Sect. 5.

In particular, we consider a potential flow model, describing steady, laminar, inviscid, irrotational flows in two-dimensional domains. Usual contexts where the potential flow model is used are, for instance, aerodynamics in the so-called *panel method* (for the simulation of flows around aircrafts and the outer flow fields for airfoils) and hydrodynamics (for example, in water waves and groundwater flows). However, this model is too simplistic, since it is not able to describe e.g. flows in presence of boundary layers or strong vorticity effects; a common strategy to take them into account consists in the coupling of a potential flow model (outside the boundary layer) with more accurate models (e.g. Navier-Stokes equations) inside the boundary layer [46].

Let us consider a rectangular domain $D \subset \mathbb{R}^2$ and a parametrized NACA airfoil $B_o(\boldsymbol{\mu})$ and denote $\Omega_o(\boldsymbol{\mu}) \subset \mathbb{R}^2$ the parametrized fluid domain given by $\Omega_o(\boldsymbol{\mu}) = D \setminus B_o(\boldsymbol{\mu})$. Denote (\mathbf{u}, p) the velocity and the pressure, respectively, of a fluid flow: under the previous assumptions, $\mathbf{u} = \nabla \phi$, can be described as the gradient of a scalar function ϕ , which is called *velocity potential* and satisfies – in the incompressible case – the Laplace equation:

$$\begin{aligned} -\Delta \phi &= 0 && \text{in } \Omega_o(\boldsymbol{\mu}) \\ \frac{\partial \phi}{\partial \mathbf{n}} &= 0 && \text{on } \Gamma_w(\boldsymbol{\mu}) \\ \frac{\partial \phi}{\partial \mathbf{n}} &= \phi_{\text{in}} && \text{on } \Gamma_{\text{in}}(\boldsymbol{\mu}) \\ \phi &= \phi_{\text{ref}} && \text{on } \Gamma_{\text{out}}(\boldsymbol{\mu}), \end{aligned}$$

where homogeneous Neumann conditions describe non-penetration on the walls $\Gamma_w(\boldsymbol{\mu})$, inhomogeneous Neumann conditions are used to impose the velocity \mathbf{u}_{in} on the inflow boundary $\Gamma_{\text{in}}(\boldsymbol{\mu})$ (being $\phi_{\text{in}} = \mathbf{u}_{\text{in}} \cdot \mathbf{n}$ and \mathbf{n} the outward normal vector to $\Gamma_{\text{in}}(\boldsymbol{\mu})$) and Dirichlet conditions are employed to prescribe the level of the potential on the outflow boundary $\Gamma_{\text{out}}(\boldsymbol{\mu})$; see Fig. 1 for the geometrical configuration used in this example.

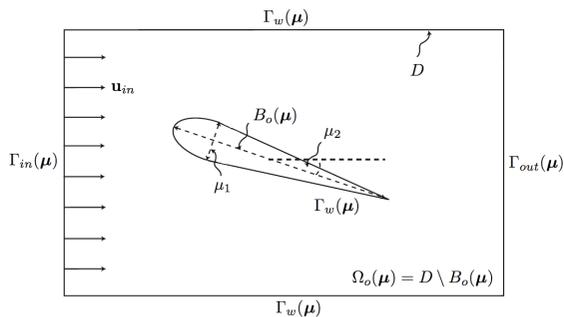


FIGURE 1. Geometrical configuration and parameters for the potential flow example.

The pressure p can be obtained by Bernoulli's equation:

$$p + \frac{1}{2} \rho |\mathbf{u}|^2 = p_{\text{in}} + \frac{1}{2} \rho |\mathbf{u}_{\text{in}}|^2, \quad \text{in } \Omega_o(\boldsymbol{\mu}),$$

whereas the pressure coefficient c_p – useful to study the aerodynamical performances of the airfoil – can be defined as

$$c_p(p) = \frac{p - p_{\text{in}}}{\frac{1}{2} \rho |\mathbf{u}_{\text{in}}|^2} = 1 - \left(\frac{|\mathbf{u}|^2}{|\mathbf{u}_{\text{in}}|^2} \right),$$

where p_{in} and \mathbf{u}_{in} are the pressure and the velocity of the undisturbed flow on the inflow boundary, respectively. The weak formulation of this problem on the parametrized domain $\Omega_o(\boldsymbol{\mu})$ is given by: find $u \in X(\Omega_o(\boldsymbol{\mu}))$ s.t.

$$a_o(u, v; \boldsymbol{\mu}) = f_o(v; \boldsymbol{\mu}), \quad \forall v \in X(\Omega_o(\boldsymbol{\mu}))$$

being

$$a_o(w, v; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} \nabla w \cdot \nabla v d\Omega_o, \quad f_o(w; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} w \phi_{in} d\Omega_o,$$

and assuming $\mathbf{u}_{in} = (1, 0)$, $\phi_{ref} = 0$. In particular, we consider the flow around a symmetric airfoil profile parametrized w.r.t. thickness $\mu_1 \in [4, 24]$ and the angle of attack $\mu_2 \in [0.01, \pi/4]$; the profile is rotated according to μ_2 , while the external boundaries remain fixed and the inflow velocity is parallel to the x axis (see Fig. 1). A possible parametrization (NACA family) is

$$\begin{aligned} \mathbf{x}_o &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} \cos \mu_2 & -\sin \mu_2 \\ \sin \mu_2 & \cos \mu_2 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & \pm \mu_1/20 \end{pmatrix} \begin{pmatrix} 1-t^2 \\ \varphi(t) \end{pmatrix}, \quad t \in [0, \sqrt{0.3}] \\ \mathbf{x}_o &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \cos \mu_2 & -\sin \mu_2 \\ \sin \mu_2 & \cos \mu_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \pm \mu_1/20 \end{pmatrix} \begin{pmatrix} t^2 \\ \varphi(t) \end{pmatrix}, \quad t \in [\sqrt{0.3}, 1], \end{aligned}$$

being $\varphi(t) = 0.2969t - 0.1260t^2 - 0.3520t^4 + 0.2832t^6 - 0.1021t^8$ the parametrization of the boundary. By means of this geometrical map, an automatic affine representation based on domain decomposition (where basic subdomains are either straight or curvy triangles) can be built within the `rbMIT` library [49]; in this way, we can recover an affine decomposition, which in this case consists of $Q_a = 45$, $Q_f = 1$ terms. For the derivation of the parametrized formulation (2.7) and of the affinity assumptions, we refer the reader to [48, 49].

Starting from a *truth* FE approximation of size $\mathcal{N} \approx 3,500$ elements, the greedy procedure for the construction of the RB space selects $N_{max} = 7$ snapshots with a stopping tolerance of $\varepsilon_{tol}^{RB} = 10^{-2}$, thus yielding a reduction of 500 in the dimension of the linear system; in Fig. 2 the convergence of the greedy procedure, as well as the selected snapshots, are reported.

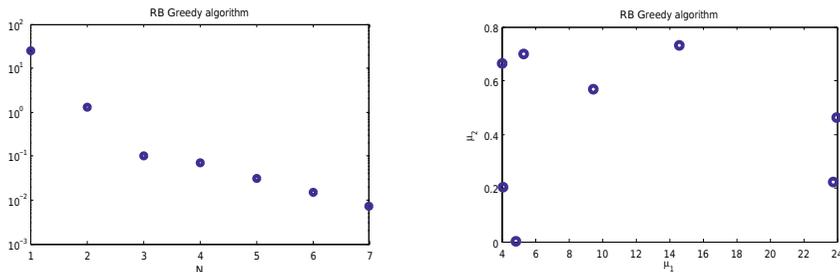


FIGURE 2. Convergence of the greedy procedure ($\max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_N(\boldsymbol{\mu})$, $N = 1, \dots, N_{max}$) and corresponding selected snapshots in the parameter space \mathcal{D} .

Concerning the computational performances, the FE *offline* stage (involving the automatic geometry handling and affine decomposition, and the FE structures assembling) takes about $^5 t_{FE}^{offline} = 8h$ on a single-processor desktop; the most expensive

⁵Computations have been executed on a personal computer with 2×2 GHz Dual Core AMD Opteron(tm) processors 2214 HE and 16 GB of RAM.

stage is the construction of the automatic affine domain decomposition. The construction of the RB space with the greedy algorithm and the algebraic structures for the efficient evaluation of the error bounds takes about $4h$, giving a total RB *offline* time of $t_{RB}^{offline} = 12h$. Concerning the *online* stage, the computation of the field solution for 100 parameter values takes $t_{FE}^{online} = 5.2s$ with the FE discretization and $t_{RB}^{online} = 2.12 \times 10^{-2}s$ with the RB approximation, entailing a computational speedup of $t_{RB}^{online}/t_{FE}^{online} = 250$. In Fig. 3 some representative solutions are shown. We can point out how, in presence of positive angles of attack, the flow fields are no longer symmetric on the two sides of the airfoil, showing increasing pressure peaks with increasing angles of attack.

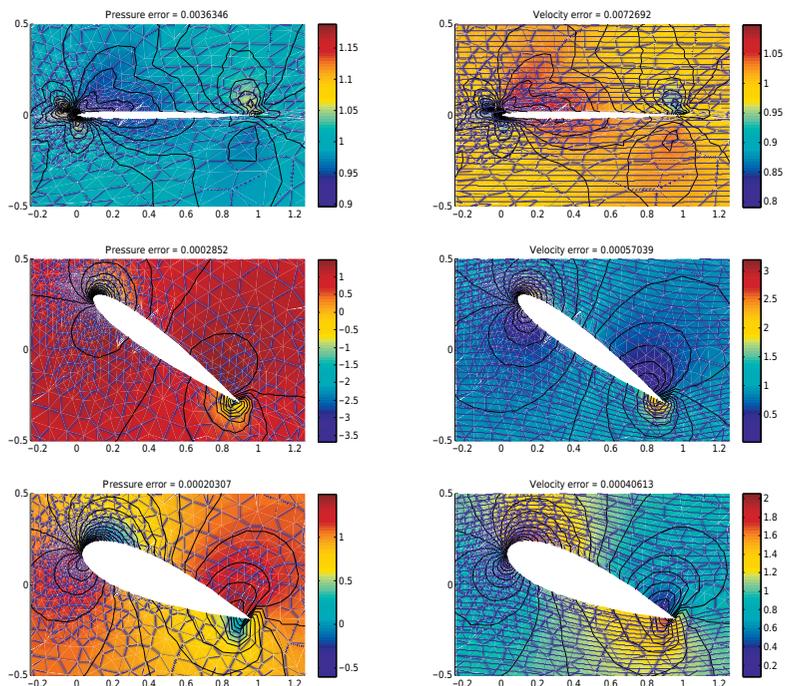


FIGURE 3. RB solutions to the potential flow problems: pressure field (left), velocity magnitude and streamlines (right) for $\mu = [4, 0]$, $\mu = [14, \pi/5]$, $\mu = [24, \pi/8]$ (from top to bottom).

We can also appreciate the limit of the model, since the streamlines on the upper and lower sides of the airfoil are not parallel to the trailing edge (thus not obeying to the Kutta condition) but form a rear stagnation point on the upper side of the profile.

Moreover, in order to evaluate the aerodynamic performance along different airfoil sections, we can evaluate the pressure coefficient c_p on the airfoil boundary – our output of interest. Thanks to the *offline/online* strategy, given a new configuration (corresponding to a new parameter combination $\mu = (\mu_1, \mu_2)$), the evaluation of $c_p(\mu)$ can be performed in almost a real time. As expected, pressure at the leading

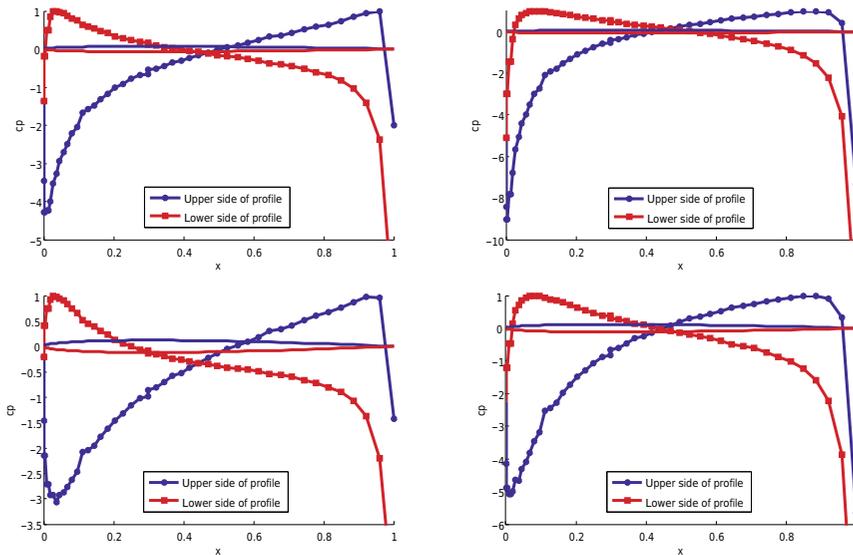


FIGURE 4. Some representative RB output computations of the pressure coefficient c_p for different values of the geometrical parameters: from top, left to bottom, right, $\boldsymbol{\mu} = [14, \pi/8]$, $\boldsymbol{\mu} = [14, \pi/5]$, $\boldsymbol{\mu} = [24, \pi/8]$, $\boldsymbol{\mu} = [24, \pi/5]$.

edge depends on both the angle of attack μ_2 and the thickness μ_1 of the profile: the smaller the angle of attack and the thinner the profile, the larger is the positive pressure. Moreover, the thickest profile shows smaller positive pressure on the lower side, while the stagnation point (corresponding to $c_p = 1$, i.e. to the point of maximum – or stagnation – pressure, where $\mathbf{u} = \mathbf{0}$) is close to the leading edge, and moves towards the midchord as μ_2 increases.

6.1. Comparison between RB approximation and surrogate models

Next we compare the performance of the RB method and the surrogate models presented in Sec. 5 in evaluating the output $s(\boldsymbol{\mu}) \equiv c_p(\boldsymbol{\mu})$. In particular, we consider a set of K points located on the upper part of the profile $\{\mathbf{x}_k\}_{k=1}^K$, for which we compute:

- the RB approximation $s_N(\boldsymbol{\mu}; \mathbf{x}_k) = c_p(p_N(\boldsymbol{\mu}); \mathbf{x}_k)$, obtained using the RB approximation $p_N(\boldsymbol{\mu})$ of the pressure;
- the surrogate outputs for $k = 1, \dots, K$, obtained through RSM

$$s_N^{\text{RSM},1}(\boldsymbol{\mu}; \mathbf{x}_k) = \beta_0^k + \sum_{j=1}^2 \beta_j^k \mu_j,$$

$$s_N^{\text{RSM},2}(\boldsymbol{\mu}; \mathbf{x}_k) = \beta_0^k + \sum_{i=1}^2 \beta_i^k \mu_i + \sum_{i=1}^2 \beta_{ii}^k \mu_i^2 + \sum_{i < j}^2 \beta_{ij}^k \mu_i \mu_j,$$

with polynomial regression of order 1 and 2, respectively;

- the surrogate outputs obtained through a RBF interpolation

$$s_N^{\text{RBF}}(\boldsymbol{\mu}; \mathbf{x}_k) = \beta_0^k + \sum_{j=1}^2 \beta_j^k \mu_j + \sum_{i=1}^n \gamma_i \phi(\|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|),$$

being here $\phi(r) = r^2 \log(r)$ the so-called *thin-plate spline* RBF, while the polynomial function enforces the well posedness of the interpolation problem (see e.g. [5, 33]);

- the surrogate outputs

$$s_N^{\text{KRI}}(\boldsymbol{\mu}; \mathbf{x}_k) = \beta_0^k + \sum_{i=1}^2 \beta_i^k \mu_i + \sum_{i=1}^n \lambda_i^k(\theta; \boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^n) w(\boldsymbol{\mu}_i)$$

obtained through a (universal) kriging model, where, for all $k = 1, \dots, K$, a different covariance function $\rho(h_j^k(r, s)) = \exp(-\theta_j^k h_j^2(r, s))$, $j = 1, 2$, has been considered; here $h_j(r, s) = |\mu_{rj} - \mu_{sj}|$, for $r, s = 1, \dots, n$. In particular, optimal values (maximum likelihood estimation) for the coefficients $\theta_j^k \in [0.1, 20]$ have been computed, for all k and j .

For the construction of the surrogate models, we consider a set of n experimented combinations of parameter values $\{\boldsymbol{\mu}^i\}_{i=1}^n$, in the cases $n = 7$ (corresponding to the dimension N of the RB space) and $n = 100$, and the corresponding n output values obtained through a FE simulation, for each point \mathbf{x}_k , $k = 1, \dots, K$. The sample $\{\boldsymbol{\mu}^i\}_{i=1}^n$ corresponding to the experimented parameter combinations has been randomly selected in the parameter space \mathcal{D} according to a bivariate uniform distribution. In order to compare the results, we introduce a fine test sample $\Xi_{\text{train}} \subset \mathcal{D}$ of dimension $n_{\text{train}} = 500$, selected according to a bivariate uniform distribution too. Error bounds computed directly on velocity and/or pressure solution (see Fig. 3) are available in [48].

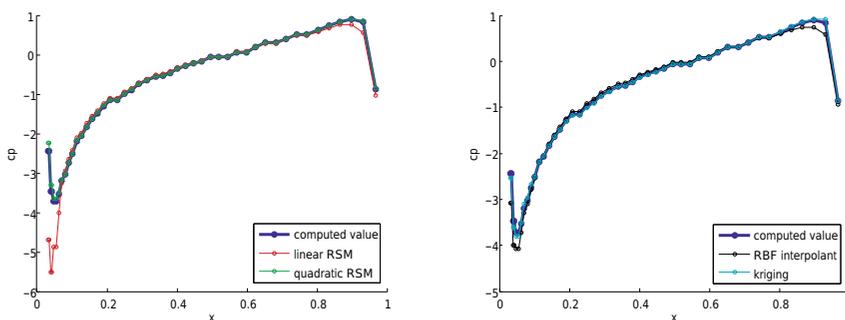


FIGURE 5. High-fidelity FEM and surrogate (based on $n = 7$ computed values) c_p distributions on the upper part of the airfoil corresponding to $\bar{\boldsymbol{\mu}} = [14.3451, 0.3655]$ (left: RSM with a first (red) and second (green) order regression models; right: RBF interpolation (black) and kriging model (cyan)).

In Fig. 5 we show the distribution of the pressure coefficient obtained through a FE high-fidelity approximation for a new, randomly chosen, parameter combination

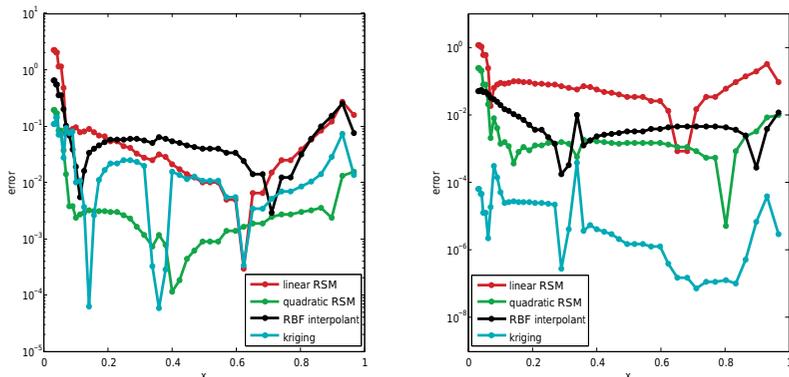


FIGURE 6. Errors between the high-fidelity and the surrogate c_p distributions corresponding to $\bar{\boldsymbol{\mu}} = [14.3451, 0.3655]$, for $N = 7$ (left) and $N = 100$ (right) computed output values.

$\bar{\boldsymbol{\mu}}$, as well the distributions obtained through the surrogate models presented, using $n = 7$ computed values. The errors between the high-fidelity approximation and the surrogate approximations are represented in Fig. 6, for the cases $n = 7$ (left) and $n = 100$ (right).

We can remark that for surrogate models built upon a small number of computed outputs, RSM with low-order polynomial regression give a result which is comparable to the one obtained with more advanced techniques, such as RBF or kriging. In particular, the error is about 10^{-1} for linear RSM and RBF, 10^{-2} (with some lower peaks) for the kriging approximation and 10^{-3} for quadratic RSM. Increasing the number of computed outputs, we find that the kriging model is the one giving the best performance (errors about $10^{-5} \div 10^{-6}$), approximation through RSM does not show a remarkable improved quality, while RBF performs in a sensibly better way. In the last Fig. 7 we compare the errors between FEM and surrogate (built in the case $n = 7$) output approximations, as well as the error estimation for the RB approximation of the output, obtained averaging the results over the train sample. We observe that the quadratic interpolant can be in fact as accurate as the RB output – and in any case more efficient if $n \approx N$ since the evaluation $\boldsymbol{\mu} \rightarrow s_N^{\text{RSM},1}(\boldsymbol{\mu})$ requires just $\mathcal{O}(n^2)$ operations whereas the online RB evaluation $\boldsymbol{\mu} \rightarrow s_N^{\text{RB}}(\boldsymbol{\mu})$ entails $\mathcal{O}(N^3)$ operations – while the RBF and the kriging outputs may be even more accurate than the RB output. However, as already remarked in [49], in higher parameter dimensions it is not possible to perform efficient approximations based on surrogate models, mainly due to:

- (i) the difficulty arising from the sampling stage for the construction of the surrogate model – whereas the greedy algorithm for the RB space construction seeks for the best candidate snapshot automatically;

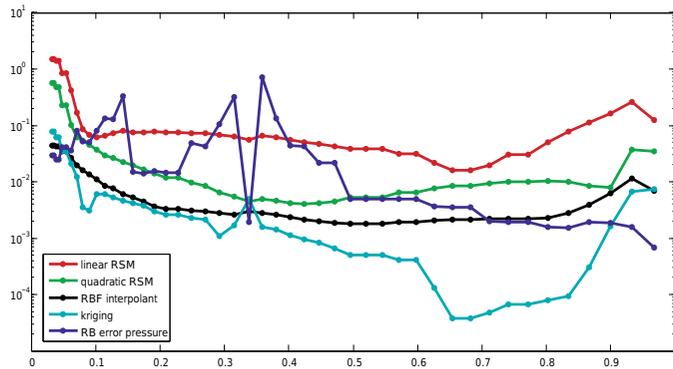


FIGURE 7. Error bound between the reduced basis and the high-fidelity approximations (in blue), and true errors between the high-fidelity and the surrogate distributions in the case $\bar{\mu} = [14.3451, 0.3655]$ for $N = 7$ computed output values.

- (ii) the complexity of any interpolation procedure, which in general is not an easy task, as well as the lack of sharp and rigorous error bounds for the output interpolants.

Acknowledgment

We thank Prof. Anna Maria Paganoni (MOX-Politecnico di Milano) for her valuable feedbacks and suggestions, as well as Ms. Claudia Günther for some parts of the code used to carry out the geometrical parametrization. We acknowledge the use of the `rbMIT` package developed by the group of A.T. Patera (MIT) as a basis for the numerical RB simulations.

We also acknowledge the use of the `DACE` toolbox developed by H.B. Nielsen and co-workers (Technical University of Denmark) as a basis for computing kriging approximations.

This work has been supported in part by the Swiss National Science Foundation (Project 200021-122136).

References

- [1] N. Aubry. On the hidden beauty of the proper orthogonal decomposition. *Theor. Comp. Fluid. Dyn.*, 2:339–352, 1991.
- [2] G. Berkooz, P. Holmes, and J.L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.*, 25(1):539–575, 1993.
- [3] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.*, 43(3):1457–1472, 2011.
- [4] P. Blanco, M. Discacciati, and A. Quarteroni. Modeling dimensionally-heterogeneous problems: analysis, approximation and applications. *Numer. Math.*, 119:299–335, 2011.

- [5] M.D. Buhmann. *Radial Basis Functions*. Cambridge University Press, UK, 2003.
- [6] T. Bui-Thanh, K. Willcox, and O. Ghattas. Parametric reduced-order models for probabilistic analysis of unsteady aerodynamics applications. *AIAA J.*, 46(10), 2008.
- [7] J. Burkardt, Q. Du, and M. Gunzburger. Reduced order modeling of complex systems, 2003. Proceedings of NA03, Dundee.
- [8] J. Burkardt, M. Gunzburger, and H.C. Lee. Centroidal voronoi tessellation-based reduced-order modeling of complex systems. *SIAM J. Sci. Comput.*, 28(2):459–484, 2006.
- [9] J. Burkardt, M. Gunzburger, and H.C. Lee. POD and CVT-based reduced-order modeling of Navier-Stokes flows. *Comp. Methods Appl. Mech. Engrg.*, 196(1-3):337–355, 2006.
- [10] M. Chevreuil and A. Nouy. Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics. *Int. J. Numer. Methods Engng.*, 89(2):241–268, 2012.
- [11] F. Chinesta, P. Ladeveze, and E. Cueto. A short review on model order reduction based on proper generalized decomposition. *Arch. Comput. Methods Engrg.*, 18:395–404, 2011.
- [12] E.A. Christensen, M. Brøns, and J.N. Sørensen. Evaluation of proper orthogonal decomposition–based decomposition techniques applied to parameter-dependent non-turbulent flows. *SIAM J. Sci. Comput.*, 21:1419, 1999.
- [13] N.A.C. Cressie. *Statistics for spatial data*. John Wiley & Sons, Ltd, UK, 1991.
- [14] S. Deparis and G. Rozza. Reduced basis method for multi-parameter-dependent steady Navier-Stokes equations: Applications to natural convection in a cavity. *J. Comput. Phys.*, 228(12):4359–4378, 2009.
- [15] M. Discacciati, P. Gervasio, and A. Quarteroni. Heterogeneous mathematical models in fluid dynamics and associated solution algorithms. In G. Naldi and G. Russo, editors, *Multiscale and Adaptivity: Modeling, Numerics and Applications (Lecture notes of the C.I.M.E. Summer School, Cetraro, Italy 2009)*, Lecture Notes in Mathematics, Vol. 2040. Springer, 2010.
- [16] A. Dumon, C. Allery, and A. Ammar. Proper general decomposition (PGD) for the resolution of Navier-Stokes equations. *J. Comput. Phys.*, 230:1387–1407, 2011.
- [17] J.P. Fink and W.C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *Z. Angew. Math. Mech.*, 63(1):21–28, 1983.
- [18] P. Gervasio, J.-L. Lions, and A. Quarteroni. Heterogeneous coupling by virtual control methods. *Numer. Math.*, 90:241–264, 2001.
- [19] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 41(3):575–605, 2007.
- [20] M.D. Gunzburger, J.S. Peterson, and J.N. Shadid. Reducer-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comput. Methods Appl. Mech. Engrg.*, 196:1030–1047, 2007.
- [21] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math. Modelling Numer. Anal.*, 42:277–302, 2008.
- [22] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge Univ. Press, 1998.

- [23] H. Hotelling. Simplified calculation of principal components. *Psychometrika*, 1:27–35, 1936.
- [24] K. Ito and S.S. Ravindran. A reduced order method for simulation and control of fluid flows. *J. Comput. Phys.*, 143(2), 1998.
- [25] P.S. Johansson, H.I. Andersson, and E.M. Rønquist. Reduced-basis modeling of turbulent plane channel flow. *Compu. Fluids*, 35(2):189–207, 2006.
- [26] J. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal Of Operational Research*, 192(3):707–716, 2009.
- [27] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2003.
- [28] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand.*, 45:255–282, 1950.
- [29] C. Lieberman, K. Willcox, and O. Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM J. Sci. Comput.*, 32(5):2523–2542, 2010.
- [30] X. Ma and G.E.M. Karniadakis. A low-dimensional model for simulating three-dimensional cylinder flow. *J. Fluid. Mech*, 458:181–190, 2002.
- [31] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *Int. J. Numer. Methods Biomed. Engng.*, 2011. In press (DOI: 10.1002/cnm.1465).
- [32] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization of cardiovascular geometries by reduced basis methods and free-form deformation techniques. *Int. J. Numer. Methods Fluids*, 2011. In press (DOI: 10.1002/flid.2712).
- [33] D.B. McDonald, W.J. Grantham, W.L. Tabor, and M.J. Murphy. Global and local optimization using radial basis function response surface models. *Applied Mathematical Modelling*, 31(10):2095–2110, 2007.
- [34] N.C. Nguyen, K. Veroy, and A.T. Patera. Certified real-time solution of parametrized partial differential equations. In: *Yip, S. (Ed.). Handbook of Materials Modeling*, pages 1523–1558, 2005.
- [35] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA J.*, 18(4):455–462, 1980.
- [36] A. Nouy. Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Methods Engng.*, 17:403–434, 2010.
- [37] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equation*. Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2009.
- [38] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.
- [39] J. Peiró and A. Veneziani. Reduced models of the cardiovascular system. In: *Formaggia, L.; Quarteroni, A; Veneziani, A. (Eds.), Cardiovascular Mathematics*, Springer, 2009.
- [40] J.S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM J. Sci. Stat. Comput.*, 10:777–786, 1989.

- [41] R. Pinnau. Model reduction via proper orthogonal decomposition. In W.H.A. Schilder and H. van der Vorst, editors, *Model Order Reduction: Theory, Research Aspects and Applications*,, pages 96–109. Springer, 2008.
- [42] T.A. Porsching and M.Y. Lin Lee. The reduced-basis method for initial value problems. *SIAM Journal of Numerical Analysis*, 24:1277–1287, 1987.
- [43] C. Prud’homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bounds methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.
- [44] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. *Numer. Methods Partial Differential Equations*, 23(4):923–948, 2007.
- [45] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations in industrial applications. *J. Math. Ind.*, 1(3), 2011.
- [46] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, 1999.
- [47] A. Quarteroni and A. Veneziani. Analysis of a geometrical multiscale model based on the coupling of pdes and odes for blood flow simulations. *SIAM J. on Multiscale Model. Simul.*, 1(2):173–195, 2003.
- [48] G. Rozza. Reduced basis approximation and error bounds for potential flows in parametrized geometries. *Comm. Comput. Phys.*, 9:1–48, 2011.
- [49] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Engrg.*, 15:229–275, 2008.
- [50] G. Rozza and K. Veroy. On the stability of reduced basis methods for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.
- [51] T.J. Santner, B.J. Williams, and W. Notz. *The design and analysis of computer experiments*. Springer-Verlag, New York, 2003.
- [52] W. Schilder. Introduction to model order reduction. In W. Schilder and H. van der Vorst, editors, *Model Order Reduction: Theory, Research Aspects and Applications*,, pages 3–32. Springer, 2008.
- [53] L. Sirovich. Turbulence and the dynamics of coherent structures, part i: Coherent structures. *Quart. Appl. Math.*, 45(3):561–571, 1987.
- [54] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Methods Fluids*, 47(8-9):773–788, 2005.
- [55] F.A.C. Viana, C. Gogu, and R.T. Haftka. Making the most out of surrogate models: tricks of the trade. In *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pages 587–598, 2010.
- [56] S. Volkwein. Model reduction using proper orthogonal decomposition, 2011. Lecture Notes, University of Konstanz, www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Vorlesung.pdf.

Andrea Manzoni
CMCS – Modelling and Scientific Computing
MATHICSE – Mathematics Institute of Computational Science and Engineering
EPFL – Ecole Polytechnique Fédérale de Lausanne
Station 8, CH-1015 Lausanne
Switzerland
e-mail: andrea.manzoni@epfl.ch

Alfio Quarteroni
CMCS – Modelling and Scientific Computing
MATHICSE – Mathematics Institute of Computational Science and Engineering
EPFL – Ecole Polytechnique Fédérale de Lausanne
Station 8, CH-1015 Lausanne
Switzerland

and

MOX – Modellistica e Calcolo Scientifico
Dipartimento di Matematica “F. Brioschi” – Politecnico di Milano
P.za Leonardo da Vinci 32, I-20133 Milano
Italy
e-mail: alfio.quarteroni@epfl.ch

Gianluigi Rozza
CMCS – Modelling and Scientific Computing
MATHICSE – Mathematics Institute of Computational Science and Engineering
EPFL – Ecole Polytechnique Fédérale de Lausanne
Station 8, CH-1015 Lausanne
Switzerland
e-mail: gianluigi.rozza@epfl.ch

Received: March 7, 2012.