



MOX-Report No. 35/2015

**A certified reduced basis method for PDE-constrained
parametric optimization problems by an adjoint-based
approach**

Manzoni, A.; Pagani, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<http://mox.polimi.it>

A certified reduced basis method for PDE-constrained parametric optimization problems by an adjoint-based approach*

A. Manzoni[†] S. Pagani[‡]

June 18, 2015

Abstract

In this paper we present a certified reduced basis (RB) framework for the efficient solution of PDE-constrained parametric optimization problems. We consider optimization problems (such as optimal control and optimal design) governed by elliptic PDEs and involving possibly non-convex cost functionals, assuming that the control functions are described in terms of a parameters vector. At each optimization step, the high-fidelity approximation of state and adjoint problems is replaced by a certified RB approximation, thus yielding a very efficient solution through an “optimize-then-reduce” approach. We develop a posteriori error estimates for the solutions of state and adjoint problems, for the cost functional, its gradient and the optimal parameters. We confirm our theoretical results in the case of optimal control/design problems dealing with potential and thermal flows.

1 Introduction

PDE-constrained optimization problems are widespread in applied sciences and engineering because of the ubiquitous need to control a system in order to reach some target. Two well-known cases are the optimal control of thermal flows in heating/cooling devices or the optimal design of airfoils. These problems, and more in general optimal control, optimal design and (deterministic) inverse identification problems, are usually cast within a constrained optimization framework in order to exploit a well-established theoretical setting and powerful approximation techniques [6, 15].

A very common assumption to handle the solution of PDE-constrained optimization problems is to express control variables, shapes, or unknown features to be recovered through a set of parameters, thus recasting these problems in a more affordable finite-dimensional framework. Nevertheless, even under this assumption, numerical optimization stands over iterative procedures which usually requires a large amount of PDE solutions, evaluations of the cost functional, and possibly of its gradient with respect to control/design variables, such as in the case of descent methods or nonlinear programming techniques [15]. Obtaining a significant computational speedup within this context has been one of the leading motivations behind the development of efficient reduced order models for PDE systems in the last decade. Reduced basis (RB) method and Proper Orthogonal Decomposition (POD) have been widely applied to the solution of optimal control and shape optimization problems; among many contributions, we cite [32, 13, 36, 9, 10, 38, 27, 12, 11, 42] regarding the former technique, and [20, 17, 18, 39, 1] concerning the latter. More recently, conversely to the *parametric optimization problems* considered in this paper, the case of *parametrized optimal control problems* has also been addressed.

*We are grateful to Prof. Sandro Salsa (Politecnico di Milano) for the scientific discussions and the support provided, and to Dr. Luca Dedè (EPFL) for the useful advices and help.

[†]CMCS-MATHICSE-SB, Ecole Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland, andrea.manzoni@epfl.ch (corresponding author)

[‡]MOX, Dipartimento di Matematica, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133 Milano, Italy, stefano.pagani@polimi.it

Within this latter class of problems, parameters affect either differential operators or data, *but not* the control/design variables, thus characterizing multiple scenarios where an (infinite dimensional) optimal control has to be determined; see e.g. [30, 29, 14, 19] and also the recent review presented in [5].

In this paper we consider *PDE-constrained parametric optimization problems* (μ -OPs) where control/design variables are expressed in terms of a vector of parameters $\mu \in \mathcal{P} \subseteq \mathbb{R}^d$, thus yielding *finite-dimensional control variables*. Parameters may affect boundary data, source terms, physical coefficients, or the geometrical configuration of the domain. Indeed, our framework is able to account for – apparently different – problems such as optimal control, inverse identification, optimal design, which can be unified under the parametric assumption and cast under the following form:

$$\min_{\mu \in \mathcal{P}_{ad}} J(\mu) \quad \text{s.t.} \quad a(u(\mu), v; \mu) = f(v; \mu) \quad \forall v \in X. \quad (\mu\text{-OP})$$

Here $u(\mu) \in X$ is the state solution, X is a suitable Sobolev space defined over a spatial domain $\Omega \subset \mathbb{R}^m$, $m = 1, 2, 3$, and the state constraint is given by a linear, steady, elliptic PDE. Moreover, $\mathcal{P}_{ad} \subseteq \mathcal{P} \subseteq \mathbb{R}^d$ denotes a closed set of admissible parameters. Furthermore, $J(\mu) = \bar{J}(u(\mu), \mu)$ is a suitable cost functional, being \bar{J} a quadratic function of the state u . Since the map $\mu \mapsto u(\mu)$ is usually nonlinear, J is not a quadratic functional of μ ; in general, J might not even be convex.

In several works, RB methods have been applied to parametric optimization problems just for the sake of state reduction, yielding a *reduce-then-optimize* approach, where the state system is solved in a reduced state space and numerical optimization is performed according to black-box (or derivative-free) methods. These latter simply require the solution of the state problem and the evaluation of J , relying on (e.g. finite differences) approximations of its derivatives. In this work we rely instead on RB methods for the reduction of the adjoint problem (or the sensitivity equations), too, thus enabling to perform the evaluation of the gradient of J at the reduced level. This can be seen as an *optimize-then-reduce* approach, meaning that first we express optimality conditions and derivatives in PDE form, then we construct a RB method for any μ -dependent PDE to be solved. Relying on this approach is also mandatory for deriving effective a posteriori error estimators on J , its gradient and the optimal solution.

Extending some results shown in [12, 24], in this work:

1. we state a well-posedness result, showing that the assumption of *affine parametric dependence* usually required in the RB context is crucial also for the sake of theoretical analysis of μ -OPs;
2. we derive optimality conditions by either evaluating the parametric sensitivities of the cost functional (involving the directional derivatives of the state solution with respect to parameter components), or applying an adjoint approach;
3. we implement a certified RB method for μ -OPs and show that the adjoint-based approach yields a stronger speedup than the sensitivity-based one (considered instead in [12]);
4. we extend previous results shown in [12] and provide a complete and detailed characterization of error bounds for the cost functional, its gradient and the optimal solution;
5. we take advantage of these error bounds to set descent methods where the stopping criteria are motivated by the reliability of the reduced-order approximation, showing that both optimization and reduction stages require a simultaneous (and careful) tuning.

Several problems which can be cast in this framework have been already tackled with RB methods – such as optimal control [30], optimal design [27, 22, 26], inverse parameter identification [13], model identification [2], uncertainty quantification problems [21, 23, 26] – but very often without relying on adjoint-based approaches, neither developing a rigorous error bound for the optimal solution. The general framework proposed in this paper allows a unified analysis (related to both well-posedness and a posteriori error analysis) and a possible extension to several applications also in view of cross-intersections among those class of problems (e.g., simultaneous control and design, or optimization under uncertainty problems).

The paper is organized as follows. In Sect. 2 we provide a general formulation of the class of problems we are interested to, whereas in Sect. 3 we show some well-posedness results and derive optimality conditions (of both first and second order). We set a RB method for the solution of μ -OPs in Sect. 4, by considering either the evaluation of parametric sensitivities, or the solution of an adjoint problem. In Sect. 5 we derive a posteriori error bounds for the quantities of interest in setting the optimization procedure. Finally, we confirm our theoretical results by showing some numerical results dealing with thermal and potential flows in Sect. 6.

2 Problem setting

We deal with steady scalar problems defined over a regular and bounded domain $\Omega \subset \mathbb{R}^m$, $m = 1, 2, 3$. Let us denote by X a Sobolev space such that $H_0^1(\Omega) \subseteq X \subseteq H^1(\Omega)$, taking into account boundary conditions over $\Gamma = \partial\Omega$; X^* denotes the dual of X . The parametrized bilinear form $a(\cdot, \cdot; \mu) : X \times X \rightarrow \mathbb{R}$ corresponds to the weak form of an elliptic, second-order parametrized differential operator; we require that, for any $\mu \in \mathcal{P} \subset \mathbb{R}^d$, a is continuous and coercive, i.e. there exist $\alpha(\mu) > 0$ and $M_a(\mu) < +\infty$ such that

$$\alpha(\mu) = \inf_{v \in X} \frac{a(v, v; \mu)}{\|v\|_X^2}, \quad M_a(\mu) = \sup_{u \in X} \sup_{v \in X} \frac{a(u, v; \mu)}{\|u\|_X \|v\|_X} \quad \forall \mu \in \mathcal{P}, \quad (1)$$

and that $f(\cdot; \mu) \in X^*$ is continuous for any $\mu \in \mathcal{P}$, that is, there exist $M_f(\mu) < +\infty$

$$M_f(\mu) = \sup_{v \in X} \frac{f(v; \mu)}{\|v\|_X} \quad \forall \mu \in \mathcal{P}. \quad (2)$$

As already remarked, *control/design* parameters μ might be related to: (i) distributed and/or boundary control, and then affect the right-hand side $f(\cdot; \mu)$; (ii) physical coefficients of the PDE operator, and then affect the left-hand side $a(\cdot, \cdot; \mu)$; (iii) the shape of the domain, and then affect both terms of the state equation. Through these choices, we can deal with optimal control, identification or optimal design problems, or even possible combinations among them. When dealing with optimal design, we rely on a *fixed domain* approach, that is, the state problem originally posed on a parametrized domain $\tilde{\Omega} = \tilde{\Omega}(\mu)$ is mapped back onto a reference configuration Ω , by introducing a suitable, μ -dependent geometric map (or collection of maps) from Ω to $\tilde{\Omega}$. This is a standard procedure when dealing with RB methods (see e.g. [35, 37] for further details), which also enables a strong speedup of optimal design problems by avoiding remeshing when the domain undergoes shape deformations. Moreover, we assume that $J : \mathcal{P} \rightarrow \mathbb{R}$ is given by

$$J(\mu) = \tilde{J}(u(\mu), \mu) = \frac{1}{2} \|Cu(\mu) - z_d\|_{\mathcal{Z}}^2 + \frac{\beta}{2} \|\mu - \mu_d\|_{\mathbb{R}^d}^2 \quad (3)$$

being $C : X \rightarrow \mathcal{Z}$ a continuous observation operator, that is, $\|Cv\|_{\mathcal{Z}} \leq \gamma \|v\|_X \quad \forall v \in X$, with \mathcal{Z} a suitable (Hilbert) space of observations defined over Ω . Here $z_d \in \mathcal{Z}$ is the desired target, whereas $\beta \geq 0$ and $\mu_d \in \mathcal{P}$ is a prescribed parameter value. Here \tilde{J} is a quadratic functional of the state $u(\mu)$, whereas in the more standard optimal control problems, the quadratic penalization term shall enforce the cost functional to be convex. However, we set $\beta = 0$ from now on, to take into account the more general case of possibly nonconvex functionals. Note that the cost functional might depend on the control/design parameters through both the domain and the state solutions; hence, we denote by $J(\mu) = \tilde{J}(u(\mu), \mu)$ where no ambiguity occurs.

Furthermore, we require the following *affine decomposition* (or parametric separability) for the forms appearing in (μ -OP), by expressing for any $\mu \in \mathcal{P}$

$$a(u, v; \mu) = \sum_{q=1}^{Q_a} \Theta_q^a(\mu) a_q(u, v), \quad f(v; \mu) = \sum_{q=1}^{Q_f} \Theta_q^f(\mu) f_q(v), \quad (4)$$

for Q_a, Q_f (real) functions $\Theta_q^a, \Theta_q^f : \mathcal{P} \rightarrow \mathbb{R}$, and (continuous) bilinear (resp. linear) forms $a_q(\cdot, \cdot) : X \times X \rightarrow \mathbb{R}$, $q = 1, \dots, Q_a$ (resp. $f_q(\cdot) : X \rightarrow \mathbb{R}$, $q = 1, \dots, Q_f$). We denote by M_a^q , $q = 1, \dots, Q_a$ (resp. M_f^q , $q = 1, \dots, Q_f$) the continuity factor of the μ -independent bilinear (resp. linear) forms.

Moreover, we express the cost functional (3) as

$$J(\boldsymbol{\mu}) = g(u(\boldsymbol{\mu}), u(\boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \left(\frac{1}{2} s_q(u(\boldsymbol{\mu}), u(\boldsymbol{\mu})) + l_q(u(\boldsymbol{\mu})) \right), \quad (5)$$

for Q_g functions $\Theta_q^g(\boldsymbol{\mu}) : \mathcal{P} \rightarrow \mathbb{R}$, $q = 1, \dots, Q_g$ and bilinear (resp. linear) forms $s_q(\cdot, \cdot) : X \times X \rightarrow \mathbb{R}$ (resp. $l_q(\cdot) : X \rightarrow \mathbb{R}$). In the remainder, we consider two classes of functionals which can be cast under the form (3), by choosing either $\mathcal{Z} = H^1(\Omega_{obs})$ or $L^2(\Omega_{obs})$, C as the restriction operator over a prescribed observation region $\Omega_{obs} \subset \Omega$ and z_d a prescribed target function. Thus, we end up with the following formulation of problem ($\boldsymbol{\mu}$ -OP):

$$\begin{aligned} \min_{\boldsymbol{\mu} \in \mathcal{P}_{ad} \subseteq \mathcal{P}} \quad & \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \left(\frac{1}{2} s_q(u(\boldsymbol{\mu}), u(\boldsymbol{\mu})) + l_q(u(\boldsymbol{\mu})) \right) \\ \text{s.t.} \quad & \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) a_q(u(\boldsymbol{\mu}), v) = \sum_{q=1}^{Q_f} \Theta_q^f(\boldsymbol{\mu}) f_q(v) \quad \forall v \in X. \end{aligned} \quad (6)$$

3 Well-posedness analysis and optimality conditions

Problem (6) is a special case of PDE-constrained optimization problem, where the control variable is set in a closed subset of \mathbb{R}^d rather than to an infinite dimensional (Banach or Hilbert) space. Although the existence of a minimizer follows from Weierstrass theorem (see e.g. [40]) in a straightforward way, its uniqueness is not ensured. A natural requirement for uniqueness would be the strict convexity of the cost functional with respect to the control variable $\boldsymbol{\mu}$; however, this is in general hard to satisfy, because of the nonlinear $\boldsymbol{\mu}$ -dependency of the state solution, already in the simplest case of a quadratic functional of the state variable and a linear state problem.

After analyzing the well-posedness of problem (6) and the differentiability of the state solution with respect to $\boldsymbol{\mu}$, we derive a set of first and second order optimality conditions.

3.1 Existence of a minimizer

A further assumption concerning the map $\boldsymbol{\mu} \rightarrow u(\boldsymbol{\mu})$, which also affects the cost functional, is required in order to apply Weierstrass theorem to problem ($\boldsymbol{\mu}$ -OP), as stated in the following

Proposition 3.1. *Under the following assumptions:*

1. $\mathcal{P}_{ad} \subset \mathbb{R}^d$ is a closed set;
2. Θ_q^a , $q = 1, \dots, Q_a$ and Θ_q^f , $q = 1, \dots, Q_f$ are Lipschitz continuous functions, with constants Λ_q^a , $\Lambda_q^f > 0$, respectively;
3. J is sequentially lower continuous (with respect to $\boldsymbol{\mu}$),

problem ($\boldsymbol{\mu}$ -OP) admits (at least) a minimization point $(\hat{u}, \hat{\boldsymbol{\mu}}) = (u(\hat{\boldsymbol{\mu}}), \hat{\boldsymbol{\mu}})$.

Proof. Thanks to Lax-Milgram lemma (see e.g. [34]), the properties (1)-(2) of the bilinear and linear forms, and assumption 2., we have that the map $\boldsymbol{\mu} \rightarrow u(\boldsymbol{\mu})$ defines a bounded injective operator; moreover,

$$\|u(\boldsymbol{\mu}) - u(\bar{\boldsymbol{\mu}})\|_X \leq \frac{1}{\alpha(\boldsymbol{\mu})} \left(\sum_{q=1}^{Q_f} M_f^q \Lambda_q^f + \frac{M_f(\bar{\boldsymbol{\mu}})}{\alpha(\bar{\boldsymbol{\mu}})} \sum_{q=1}^{Q_a} M_a^q \Lambda_q^a \right) \|\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}\|_{\mathbb{R}^d}. \quad (7)$$

Assumption 3. is verified since J is given by the composition of a continuous function and a semi-continuous one. The existence of a minimizer follows thanks to the Weierstrass theorem. \square

3.2 Differentiability of the cost functional

Let us now investigate the differentiability of the cost functional, for which we require that the functions Θ_q^a , $q = 1, \dots, Q_a$ and Θ_q^f , $q = 1, \dots, Q_f$ are of class $C^1(\mathcal{P})$. Numerical optimization procedures such as descent methods require the evaluation of the derivatives of the cost functional $J(\boldsymbol{\mu}) = \tilde{J}(u(\boldsymbol{\mu}), \boldsymbol{\mu})$ with respect to the parameters, given by

$$\nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu}) \cdot \mathbf{e}_i = \frac{\partial J}{\partial \mu_i}(\boldsymbol{\mu}) = \frac{\partial \tilde{J}}{\partial u} \frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu}) + \frac{\partial \tilde{J}}{\partial \mu_i}(\boldsymbol{\mu}). \quad (8)$$

To ensure that J is differentiable, we require that the map $\boldsymbol{\mu} \mapsto u(\boldsymbol{\mu})$ is differentiable, too. This property results from the *implicit function theorem*¹ (see e.g. [41]), according to the following

Theorem 3.2. *Let $A \subset X \times \mathcal{P}$ be an open subspace and $\Phi : A \rightarrow X^*$ such that:*

- $\Phi \in C^1(A)$,
- $\Phi(\hat{\boldsymbol{\mu}}, u(\hat{\boldsymbol{\mu}})) = 0$ and $\Phi_u(\hat{\boldsymbol{\mu}})$ is an isomorphism from X to X^* .

Then, $\boldsymbol{\mu} \rightarrow u(\boldsymbol{\mu})$ is a $C^1(U)$ map in a neighbourhood U of $\hat{\boldsymbol{\mu}}$, and $\Phi(\boldsymbol{\mu}, u(\boldsymbol{\mu})) = 0$ for any $\boldsymbol{\mu} \in U$.

Proof. In order to apply the implicit function theorem to the problem at hand, we rewrite the state problem appearing in (6) by isolating a $\boldsymbol{\mu}$ -independent operator Π corresponding to an injective operator². Hence, we rewrite the state problem under strong form in X^* as: find $u \in X$ such that

$$\Pi u + \sum_{q=2}^{Q_a} \hat{\Theta}_q^a(\boldsymbol{\mu}) A_q u - \sum_{q=1}^{Q_f} \hat{\Theta}_q^f(\boldsymbol{\mu}) F_q = 0 \quad \text{in } X^*.$$

Then, we define the operator

$$\Phi : X \times \mathcal{P} \rightarrow X^* : (u, \boldsymbol{\mu}) \rightarrow \sum_{q=1}^{Q_f} \hat{\Theta}_q^f(\boldsymbol{\mu}) F_q - \sum_{q=2}^{Q_a} \hat{\Theta}_q^a(\boldsymbol{\mu}) A_q u.$$

Thanks to the continuity of the forms $a^q(\cdot, \cdot)$ and $f_q(\cdot)$ and since the functions Θ_q^a , $q = 1, \dots, Q_a$ and Θ_q^f , $q = 1, \dots, Q_f$ are of class $C^1(\mathcal{P})$, Φ is a $C^1(X \times \mathcal{P})$ map. For any $\boldsymbol{\mu}$, let us denote by $\Pi^{-1} : X^* \rightarrow X$ the inverse of the injective operator yielding the unique solution of $\Pi y = v$ in X^* . We can formulate the state problem as

$$F(u, \boldsymbol{\mu}) = u - \Pi^{-1}(\Phi(u, \boldsymbol{\mu})) = 0; \quad (9)$$

Π^{-1} is continuously differentiable (because it is linear) and, for a reference value $\bar{\boldsymbol{\mu}}$ ($\bar{u} = u(\bar{\boldsymbol{\mu}})$), we obtain

$$F_u(\bar{u}, \bar{\boldsymbol{\mu}})v = v - \Pi^{-1}(\Phi_u(\bar{u}, \bar{\boldsymbol{\mu}})v) \quad \forall v \in X,$$

by applying the chain rule to F . Here $F_u(\bar{u}, \bar{\boldsymbol{\mu}})$ is an isomorphism in X , since the problem

$$v - \Pi^{-1}\left(\sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) A_q v\right) = w \quad (10)$$

has a unique solution $\forall w \in X$. Thanks to the implicit function theorem, (9) defines $u = u(\boldsymbol{\mu})$ with $u \in C^1(\mathcal{P})$. \square

By deriving (9) w.r.t. $\boldsymbol{\mu}$ we can obtain the expressions of the first-order *parametric sensitivities* of the state solution. In fact, the directional derivative $\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}$, for any $\mathbf{k} \in \mathbb{R}^d$, is given by the solution of

$$a(\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} (\nabla_{\boldsymbol{\mu}} \Theta_q^f(\boldsymbol{\mu}) \cdot \mathbf{k}) f_q(v) - \sum_{q=1}^{Q_a} (\nabla_{\boldsymbol{\mu}} \Theta_q^a(\boldsymbol{\mu}) \cdot \mathbf{k}) a_q(u(\boldsymbol{\mu}), v) \quad \forall v \in X. \quad (11)$$

¹We follow this approach also in view of possible extensions to nonlinear state problems.

²If one of the bilinear forms $a_q(\cdot, \cdot)$ is coercive, identifying Π is straightforward thanks to the Riesz representation theorem, by taking $\hat{\Theta}_q^a(\boldsymbol{\mu}) = \Theta_q^a(\boldsymbol{\mu})/\Theta_1^a(\boldsymbol{\mu})$, $q = 2, \dots, Q_a$, and $\hat{\Theta}_q^f(\boldsymbol{\mu}) = \Theta_q^f(\boldsymbol{\mu})/\Theta_1^f(\boldsymbol{\mu})$, $q = 1, \dots, Q_f$.

By choosing $\mathbf{k} = \mathbf{e}_i$, where \mathbf{e}_i denotes the i -th unit vector of the canonical basis of \mathbb{R}^d , (11) yields d equations for the first-order parametric sensitivities $\partial u(\boldsymbol{\mu})/\partial \mu_i$, $i = 1, \dots, d$, required to express the first order optimality conditions. The derivation of these latter expressions is the goal of the following section.

3.3 First order optimality condition

Thanks to Propositions 3.1 and 3.2 we can show the following result yielding a first-order optimality condition to characterize a minimizer of (5); see, e.g., [15] for the proof.

Proposition 3.3. *Let $\hat{\boldsymbol{\mu}} \in \mathcal{P}_{ad} \subseteq \mathcal{P}$ such that $J(\hat{\boldsymbol{\mu}}) = \min_{\boldsymbol{\mu} \in \mathcal{P}_{ad}} J(\boldsymbol{\mu})$. Under the following assumptions:*

1. $\mathcal{P}_{ad} \subset \mathbb{R}^d$ is a convex bounded and closed set;
2. J is $\boldsymbol{\mu}$ -differentiable,

the following variational inequality holds

$$\nabla_{\boldsymbol{\mu}} J(\hat{\boldsymbol{\mu}}) \cdot (\hat{\boldsymbol{\mu}} - \boldsymbol{\eta}) \geq 0 \quad \forall \boldsymbol{\eta} \in \mathcal{P}_{ad}. \quad (12)$$

We remark that (12) is a first-order necessary optimality condition. Therefore, we can apply the chain rule in order to evaluate

$$\nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu}) \cdot \mathbf{k} = \frac{\partial \tilde{J}(u(\boldsymbol{\mu}), \boldsymbol{\mu})}{\partial u} \nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k} + \nabla_{\boldsymbol{\mu}} \tilde{J}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) \cdot \mathbf{k} \quad \forall \mathbf{k} \in \mathcal{P}, \quad (13)$$

where

$$\begin{aligned} \frac{\partial \tilde{J}(u(\boldsymbol{\mu}), \boldsymbol{\mu})}{\partial u} \nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k} &= \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) (s_q(u(\boldsymbol{\mu}), \nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}) + l_q(\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k})) \\ &= dg(u(\boldsymbol{\mu}), \nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}; \boldsymbol{\mu}) \end{aligned} \quad (14)$$

$$\nabla_{\boldsymbol{\mu}} \tilde{J}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) \cdot \mathbf{k} = \sum_{q=1}^{Q_g} \nabla_{\boldsymbol{\mu}} \Theta_q^g(\boldsymbol{\mu}) \cdot \mathbf{k} \left(\frac{1}{2} s_q(u(\boldsymbol{\mu}), u(\boldsymbol{\mu})) + l_q(u(\boldsymbol{\mu})) \right) \quad (15)$$

where we have denoted by

$$dg(u, \psi; \boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) (s_q(u, \psi) + l_q(\psi)) \quad (16)$$

the Fréchet derivative (with respect to u) of $g(u, u; \boldsymbol{\mu})$, being $g(\cdot, \cdot; \boldsymbol{\mu})$ the bilinear form defined in (5). In order to exploit formula (13), we need to evaluate $\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}$ for any $\mathbf{k} \in \mathcal{P}$. As shown in the previous section, sensitivities entail in principle the solution of d further parametrized PDEs (11), thus implying an extensive computational effort, although the left-hand side is still equal to the state operator. A more feasible approach is based on the solution of an *adjoint problem*, in addition to the state problem. In fact, by adding (11) to (13) and choosing $v = p(\boldsymbol{\mu})$ in this latter equation, we can make the evaluation of $\nabla_{\boldsymbol{\mu}} J$ independent of $\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}$. This goal is achieved by choosing $p = p(\boldsymbol{\mu}) \in X$ as the solution of the adjoint problem

$$\sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) a_q^*(p(\boldsymbol{\mu}), \psi) = \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \{s_q(u(\boldsymbol{\mu}), \psi) + l_q(\psi)\} \quad \forall \psi \in X \quad (17)$$

being for any $q = 1, \dots, Q_a$ $a_q^*(p, \psi) = a_q(\psi, p) \quad \forall p, \psi \in X$

q $\boldsymbol{\mu}$ -independent adjoint bilinear forms. In this way, for any $\mathbf{k} \in \mathcal{P}$ we obtain

$$\nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu}) \cdot \mathbf{k} = \nabla_{\boldsymbol{\mu}} \tilde{J}(\boldsymbol{\mu}) \cdot \mathbf{k} - \sum_{q=1}^{Q_a} \nabla_{\boldsymbol{\mu}} \Theta_q^a(\boldsymbol{\mu}) \cdot \mathbf{k} a_q(u(\boldsymbol{\mu}), p(\boldsymbol{\mu})) + \sum_{q=1}^{Q_f} \nabla_{\boldsymbol{\mu}} \Theta_q^f(\boldsymbol{\mu}) \cdot \mathbf{k} f_q(p(\boldsymbol{\mu})). \quad (18)$$

Remark 3.1. *Our approach does not need the introduction of the Lagrangian formalism associated to the sensitivity-based approach. However, we can always recast the adjoint-based approach in the Lagrangian functional formalism; see, e.g. [15].*

When relying on the adjoint problem, the evaluation of $\nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu})$ for any $\mathbf{k} \in \mathbb{R}^d$ requires the solution of just two PDEs, in contrast to $d + 1$ PDEs which would be required when considering the solution of p sensitivity equations. This is the main reason why we consider an adjoint-based approach, instead of the sensitivity-based approach proposed e.g. in [12]. We can summarize the results obtained so far in the following

Proposition 3.4. *Let $\hat{\boldsymbol{\mu}} \in \mathcal{P}_{ad}$ be an optimal solution of ($\boldsymbol{\mu}$ -OP). Then, under the assumptions of Proposition 3.3, there exists an adjoint state $p = p(\hat{\boldsymbol{\mu}})$ such that $(u(\hat{\boldsymbol{\mu}}), p(\hat{\boldsymbol{\mu}}), \hat{\boldsymbol{\mu}})$ satisfy the first-order (necessary) optimality conditions:*

$$\begin{cases} \sum_{q=1}^{Q_a} \Theta_q^a(\hat{\boldsymbol{\mu}}) a_q(u(\hat{\boldsymbol{\mu}}), v) = \sum_{q=1}^{Q_f} \Theta_q^f(\hat{\boldsymbol{\mu}}) f_q(v) & \forall v \in X, \\ \sum_{q=1}^{Q_a} \Theta_q^a(\hat{\boldsymbol{\mu}}) a_q^*(p(\hat{\boldsymbol{\mu}}), \psi) = \sum_{q=1}^{Q_g} \Theta_q^g(\hat{\boldsymbol{\mu}}) \{s_q(u(\boldsymbol{\mu}), \psi) + l_q(\psi)\} & \forall \psi \in X, \\ \nabla_{\boldsymbol{\mu}} J(\hat{\boldsymbol{\mu}}) \cdot (\hat{\boldsymbol{\mu}} - \mathbf{k}) \geq 0 & \forall \mathbf{k} \in \overline{\mathcal{P}_{ad}}. \end{cases} \quad (19)$$

3.4 Second-order optimality condition

Provided that the functions $\Theta_q^\lambda(\cdot)$, $\lambda = a, f, g$, are of class $C^2(\mathcal{P})$, a similar adjoint-based approach can be exploited to evaluate the Hessian of the cost functional $\mathbb{H}_J(\boldsymbol{\mu}) : \mathcal{H} \rightarrow \mathbb{R}^{d \times d}$. Indeed, this latter quantity is required to implement a Newton method for numerical optimization, as well as to characterize a posteriori error bounds for the solution of the optimization problem. Furthermore, a sufficient second-order condition ensuring the strict local convexity of J is the positive-definiteness of the Hessian matrix. We aim at evaluating the Hessian matrix $(\mathbb{H}_J(\boldsymbol{\mu}))_{ij} = \frac{\partial^2 J}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu})$, $1 \leq i, j \leq d$ of J without relying on the second-order parametric sensitivities of the state solution, defined as the components of the Hessian $(\mathbb{H}_u(\boldsymbol{\mu}))_{ij} = \frac{\partial^2 u}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu})$, $1 \leq i, j \leq d$, i.e. by solving less than $d^2 + d + 1$ PDE problems; note that each component of \mathbb{H}_u is an element of X . For the sake of simplicity, instead of the generic directional derivatives $\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu}) \cdot \mathbf{k}$, $\nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu}) \cdot \mathbf{k}$ here we directly consider the partial derivatives with respect to parameter components. By taking $\mathbf{k} = \mathbf{e}_i$ in (15) and deriving with respect to μ_j we get

$$(\mathbb{H}_J(\boldsymbol{\mu}))_{ij} = \frac{\partial^2 J}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu}) = \frac{\partial}{\partial \mu_j} \left(\frac{\partial \tilde{J}}{\partial u}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) \cdot \frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu}) \right) + \frac{\partial}{\partial \mu_j} \left(\frac{\partial \tilde{J}}{\partial \mu_i}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) \right)$$

where

$$\begin{aligned} \frac{\partial}{\partial \mu_j} \left(\frac{\partial \tilde{J}}{\partial u}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) \cdot \frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu}) \right) &= \sum_{q=1}^{Q_g} \frac{\partial \Theta_q^g}{\partial \mu_j}(\boldsymbol{\mu}) \left\{ s_q(u(\boldsymbol{\mu}), \frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu})) + l_q(\frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu})) \right\} \\ &+ \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \left\{ s_q(u(\boldsymbol{\mu}), \frac{\partial^2 u}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu})) + l_q(\frac{\partial^2 u}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu})) \right\} \\ &+ \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) s_q \left(\frac{\partial u}{\partial \mu_j}(\boldsymbol{\mu}), \frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu}) \right) \end{aligned} \quad (20)$$

and

$$\begin{aligned} \frac{\partial}{\partial \mu_j} \left(\frac{\partial \tilde{J}}{\partial \mu_i}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) \right) &= \sum_{q=1}^{Q_g} \frac{\partial^2 \Theta_q^g}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu}) \left\{ \frac{1}{2} s_q(u(\boldsymbol{\mu}), u(\boldsymbol{\mu})) + l_q(u(\boldsymbol{\mu})) \right\} \\ &+ \sum_{q=1}^{Q_g} \frac{\partial \Theta_q^g}{\partial \mu_i}(\boldsymbol{\mu}) \left\{ s_q(\frac{\partial u}{\partial \mu_j}(\boldsymbol{\mu}), u(\boldsymbol{\mu})) + l_q(\frac{\partial u}{\partial \mu_j}(\boldsymbol{\mu})) \right\}. \end{aligned}$$

The second term appearing at the right-hand side of (20), namely

$$P(\boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \left\{ s_q(u(\boldsymbol{\mu}), \frac{\partial^2 u}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu})) + l_q(\frac{\partial^2 u}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu})) \right\} = dg((\mathbb{H}_u(\boldsymbol{\mu}))_{ij}, u(\boldsymbol{\mu}); \boldsymbol{\mu}) \quad (21)$$

is nothing but the right-hand side of the adjoint problem (17) with $\psi = (\mathbb{H}_u(\boldsymbol{\mu}))_{ij}$, so that

$$P(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) a_q^*(p(\boldsymbol{\mu}), (\mathbb{H}_u(\boldsymbol{\mu}))_{ij}) = \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) a_q((\mathbb{H}_u(\boldsymbol{\mu}))_{ij}, p(\boldsymbol{\mu})).$$

To avoid the evaluation of $\mathbb{H}_u(\boldsymbol{\mu})$, we can derive the state equation with respect to parameter components (similarly to what we have done in (11)) to obtain

$$\begin{aligned} \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) a_q((\mathbb{H}_u(\boldsymbol{\mu}))_{ij}, v) &= \sum_{q=1}^{Q_f} \frac{\partial^2 \Theta_q^f}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu}) f_q(v) - \sum_{q=1}^{Q_a} \frac{\partial^2 \Theta_q^a}{\partial \mu_j \partial \mu_i}(\boldsymbol{\mu}) a_q(u(\boldsymbol{\mu}), v) \\ &\quad - \frac{\partial a}{\partial \mu_i}(\frac{\partial u}{\partial \mu_j}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) - \frac{\partial a}{\partial \mu_j}(\frac{\partial u}{\partial \mu_i}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) \quad \forall v \in X \end{aligned} \quad (22)$$

being

$$\frac{\partial a}{\partial \mu_i}(u, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \frac{\partial \Theta_q^a(\boldsymbol{\mu})}{\partial \mu_i} a_q(u, v), \quad i = 1, \dots, d. \quad (23)$$

By taking $v = p(\boldsymbol{\mu})$ in (22), we can thus evaluate (21) as $P(\boldsymbol{\mu}) = \tilde{P}(u(\boldsymbol{\mu}), \frac{\partial u}{\partial \mu_1}(\boldsymbol{\mu}), \dots, \frac{\partial u}{\partial \mu_d}(\boldsymbol{\mu}))$. Hence, computing the Hessian matrix $\mathbb{H}_J(\boldsymbol{\mu})$ requires to evaluate only the state $u(\boldsymbol{\mu})$, the adjoint state $p(\boldsymbol{\mu})$ and the sensitivities $\partial_{\mu_i} u(\boldsymbol{\mu})$, that is, to solve $d+2$ instead of d^2+d+1 PDE problems.

4 Reduced basis method for PDE-constrained optimization

In this section we illustrate a reduced basis (RB) method for the efficient solution of PDE-constrained parametric optimization problems under the form ($\boldsymbol{\mu}$ -OP). We exploit a descent method for numerical optimization (such as the the projected gradient, Newton or quasi-Newton methods; see e.g. [31, 15]). To speedup the evaluation of the cost functional (and its gradient), which requires the solution of the state and the adjoint problems, we rely on a very cheap RB approximation of both these problems instead than on a much more expensive high-fidelity approximation. Such a procedure, to which we can refer to as *optimize-then-reduce*, has already been exploited to tackle several optimal control and optimal design problems. Nevertheless, a complete framework for the error control on the optimal solution (and on other quantities required to set a *reduced* descent method) is still lacking, except for the results shown in [12]. However, in these works only a sensitivity-based approach has been presented; our goal is to develop instead an adjoint-based approach, and then compare these two approaches.

4.1 High-fidelity FE approximation

Our RB method relies upon a high-fidelity Finite Element (FE) approximation of the state and the adjoint problems – the *reduction* stage does not replace the *discretization* stage of the usual *optimize-then-discretize* approach, rather it is built upon. We introduce a FE subspace $X_h \subset X$ of dimension $N_h < +\infty$, and denote by $u_h(\boldsymbol{\mu}) \in X_h$, $p_h(\boldsymbol{\mu}) \in X_h$ the FE approximation for the state and the adjoint solution, respectively, which are obtained by solving

$$\begin{aligned} a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) &= f(v_h; \boldsymbol{\mu}) & \forall v_h \in X_h \\ a^*(p_h(\boldsymbol{\mu}), \psi_h; \boldsymbol{\mu}) &= dg(u_h(\boldsymbol{\mu}), \psi_h; \boldsymbol{\mu})_{\mathcal{Z}} & \forall \psi_h \in X_h; \end{aligned} \quad (24)$$

here we use a more compact notation, without expliciting the affine parametric dependence. The high-fidelity approximation of problem ($\boldsymbol{\mu}$ -OP) reads

$$\begin{aligned} \hat{\boldsymbol{\mu}}_h &= \arg \min_{\boldsymbol{\mu} \in \mathcal{P}_{ad}} J_h(\boldsymbol{\mu}) = \tilde{J}(u_h(\boldsymbol{\mu}), \boldsymbol{\mu}) \\ \text{s.t.} \quad a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) &= f(v_h; \boldsymbol{\mu}) & \forall v_h \in X_h. \end{aligned} \quad (\boldsymbol{\mu}\text{-OP}_h)$$

Hence, a descent method based on the high-fidelity approximation (24) yields a convergent sequence $\{\boldsymbol{\mu}_h^{(k)}\}_{k \geq 0} \in \mathcal{P}_{ad}$ such that

$$\boldsymbol{\mu}_h^{(k+1)} = \Pi_{\mathcal{P}_{ad}}(\boldsymbol{\mu}_h^{(k)} + \sigma_k \mathbf{d}_h^{(k)}), \quad k = 0, 1, \dots$$

where $\Pi_{\mathcal{P}_{ad}} : \mathbb{R}^d \rightarrow \mathcal{P}_{ad}$ denotes the projection operator onto \mathcal{P}_{ad} and $\mathbf{d}_h^{(k)}$ is a descent direction, such that $\langle \nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu}_h^{(k)}), \mathbf{d}_h^{(k)} \rangle < 0$. Here we denote by $\nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}} \tilde{J}(u_h(\boldsymbol{\mu}), \boldsymbol{\mu})$ and by $\mathbf{d}_h^{(k)}$ the descent direction, that is

$$\mathbf{d}_h^{(k)} = \begin{cases} -\nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}_h^{(k)}) & \text{Gradient method} \\ -\mathbb{H}_{\text{dfp}}^{(k)} \nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}_h^{(k)}) & \text{quasi-Newton method} \\ -(\mathbb{H}_{J,h}(\boldsymbol{\mu}_h^{(k)}))^{-1} \nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}_h^{(k)}) & \text{Newton method,} \end{cases} \quad (25)$$

where the step size $\sigma_k > 0$ such that $J(\boldsymbol{\mu}_h^{(k+1)}) < J(\boldsymbol{\mu}_h^{(k)})$ can be selected through an inexact line search or the so-called Armijo rule [31]. $\mathbb{H}_{\text{dfp}}^{(k)}$ is the approximation of the inverse $\mathbb{H}_{J,h}^{-1}(\boldsymbol{\mu})$ of the high-fidelity Hessian, obtained according to the iterative Davidon-Fletcher-Powell (DFP) formula:

$$\mathbb{H}_{\text{dfp}}^{(k+1)} = \mathbb{H}_{\text{dfp}}^{(k)} + \frac{(\delta \boldsymbol{\mu}_k \delta \boldsymbol{\mu}_k^T)}{\mathbf{y}_k^T \delta \boldsymbol{\mu}_k} - \frac{(\mathbb{H}_{\text{dfp}}^{(k)} \mathbf{y}_k \mathbf{y}_k^T \mathbb{H}_{\text{dfp}}^{(k)})}{\mathbf{y}_k^T \mathbb{H}_{\text{dfp}}^{(k)} \mathbf{y}_k}, \quad (26)$$

where $\delta \boldsymbol{\mu}_k = -\sigma_k \mathbb{H}_{\text{dfp}}^{(k)} \nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}^{(k)})$ and $\mathbf{y}_k = \nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}^{(k+1)}) - \nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}^{(k)})$. Note that $\mathbb{H}_{\text{dfp}}^{(k)}$ does not require to solve problem (22) to evaluate second-order parametric sensitivities; these latter are instead required in the case of a Newton method. Finally, given a prescribed tolerance $\varepsilon_{\text{opt}}^{\text{tol}} > 0$, we iterate until, e.g., $\|\nabla_{\boldsymbol{\mu}} J_h(\hat{\boldsymbol{\mu}}_h)\|_{\mathbb{R}^d} < \varepsilon_{\text{opt}}^{\text{tol}}$ or $\|\mathbf{d}_h\|_{\mathbb{R}^d} < \varepsilon_{\text{opt}}^{\text{tol}}$. Any nonlinear programming solver can be applied to find a local optimum, such as interior point methods, penalty methods, or sequential quadratic programming, see e.g. [31]. Here we rely on simpler methods by assuming to start the optimization process with an initial guess not too far from the optimal solution in order to keep under control (and estimate) the reduction error at each stage of the minimization procedure.

4.2 RB approximation

The RB approximation of a parametrized PDE is obtained by solving a reduced problem, resulting from the (Galerkin) projection of the original problem onto a low dimensional subspace, spanned by (possibly few) snapshots of the high-fidelity problem obtained for properly selected parameter values [35]. For the case at hand, the RB approximation of the state $u_n(\boldsymbol{\mu}) \in X_n^u \subset X_h$ and the adjoint $p_n(\boldsymbol{\mu}) \in X_n^p \subset X_h$ variables solve

$$\begin{aligned} a(u_n(\boldsymbol{\mu}), v_n; \boldsymbol{\mu}) &= f(v_n; \boldsymbol{\mu}) & \forall v_n \in X_n^u \\ a^*(p_n(\boldsymbol{\mu}), \psi_n; \boldsymbol{\mu}) &= dg(u_n(\boldsymbol{\mu}), \psi_n; \boldsymbol{\mu}) & \forall \psi_n \in X_n^p. \end{aligned} \quad (27)$$

The RB approximation of problem ($\boldsymbol{\mu}$ -OP_h) thus reads

$$\begin{aligned} \hat{\boldsymbol{\mu}}_n &= \arg \min_{\boldsymbol{\mu} \in \mathcal{P}_{ad}} J_n(\boldsymbol{\mu}) = \tilde{J}(u_n(\boldsymbol{\mu}), \boldsymbol{\mu}) \\ \text{s.t.} \quad a(u_n(\boldsymbol{\mu}), v_n; \boldsymbol{\mu}) &= f(v_n; \boldsymbol{\mu}) & \forall v_n \in X_n. \end{aligned} \quad (\boldsymbol{\mu}\text{-OP}_n)$$

Hence, a descent method which exploits at each step the RB approximation ($\boldsymbol{\mu}$ -OP_n) yields a convergent sequence $\{\boldsymbol{\mu}_n^{(k)}\}_{k \geq 0} \in \mathcal{P}_{ad}$ such that

$$\boldsymbol{\mu}_n^{(k+1)} = \Pi_{\mathcal{P}_{ad}}(\boldsymbol{\mu}_n^{(k)} + \mathbf{d}_n^{(k)}), \quad k = 0, 1, \dots$$

where $\mathbf{d}_n^{(k)}$ is a reduced descent direction defined as in (25)–(26) by replacing J_h with J_n ; here $\nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}} \tilde{J}(u_n(\boldsymbol{\mu}), \boldsymbol{\mu})$. Finally, given a prescribed tolerance $\varepsilon_{\text{opt}}^{\text{tol}} > 0$, we iterate until, e.g., $\|\nabla_{\boldsymbol{\mu}} J_n(\hat{\boldsymbol{\mu}}_n)\|_{\mathbb{R}^d} < \varepsilon_{\text{opt}}^{\text{tol}}$ or $\|\mathbf{d}_n\|_{\mathbb{R}^d} < \varepsilon_{\text{opt}}^{\text{tol}}$. The proposed framework can also be extended to other constrained optimization algorithms – such as trust region or active set methods: our choice to adopt a more straightforward optimization method aims at better highlighting the computational performances of the reduced-order model.

4.3 RB space construction

We now address the construction of the RB spaces X_n^u, X_n^p for the state and the adjoint problem, respectively. The two mostly exploited techniques are POD and greedy algorithms, see, e.g. [1, 9, 10, 38, 18, 39] for recent applications to PDE-constrained optimization problems. Here we exploit a greedy algorithm relying on *a posteriori* error bounds, comparing different strategies for the construction of state space and adjoint (or, alternatively, sensitivity) spaces.

First, we propose a simultaneous strategy for the state-adjoint formulation: since these two problems are not disjoint – the right-hand side of the adjoint problem depends on the state solution $u(\boldsymbol{\mu})$ – we can take advantage of the state basis when constructing the adjoint space in order to speedup the whole procedure. This yields to the *simultaneous state/adjoint* greedy algorithm 1. A second option consists in selecting the same parameter value for both state and adjoint problems at each iteration, see Algorithm 2. As we will see in the considered test cases, parameter sampling in this case is mainly driven by one of the two problems, with a consequent loss of information in the two RB spaces X_n^u, X_n^p . The greedy algorithm can be also performed when relying on a state-sensitivity approach, instead than on a state-adjoint approach. In this case, considering for instance a different sampling for each problem, we obtain Algorithm 3. A comparison among these three algorithms is presented in Sect. 6. We denote by *Gram – Schmidt*(X_n, u) the result of the orthonormalization of u with respect to the n elements of X_n , whereas $\Delta_n^u(\boldsymbol{\mu}), \Delta_n^p(\boldsymbol{\mu})$ are two error bounds for state and adjoint solutions, such that (see Sect. 5.1)

$$\|u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu})\|_X \leq \Delta_n^u(\boldsymbol{\mu}) \quad \|p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu})\|_X \leq \Delta_n^p(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{P}.$$

Algorithm 1 Offline stage / state/adjoint, two samplings

1: **procedure** STATE-ADJOINT GREEDY ALGORITHM, TWO SAMPLINGS
Require: $n_{\max}, \varepsilon_{RB}^{tol}, \Xi_{train} \subset \mathcal{P}, \boldsymbol{\mu}_u^1, \boldsymbol{\mu}_p^1 \in \mathcal{P}$
Ensure: RB state and adjoint spaces X_n^u, X_n^p
2: $S_n^u = \emptyset, X_n^u = \emptyset, S_n^p = \emptyset, X_n^p = \emptyset$
3: $n = 0, \delta_0 = \varepsilon_{RB}^{tol} + 1$
4: **while** $n < n_{\max}$ and $\delta_n > \varepsilon_{RB}^{tol}$ **do**
5: $n \leftarrow n + 1$
6: compute $u_h(\boldsymbol{\mu}_u^n)$
7: $\zeta_n^u = \text{Gram} - \text{Schmidt}(X_n^u, u_h(\boldsymbol{\mu}_u^n))$
8: $X_n^u \leftarrow X_n^u \cup \zeta_n^u, S_n^u \leftarrow S_n^u \cup \{\boldsymbol{\mu}_u^n\}$
9: $[\delta_n^u, \boldsymbol{\mu}_u^n] = \arg \max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_n^u(\boldsymbol{\mu})$
10: compute $p_h(\boldsymbol{\mu}_p^n)$
11: $\zeta_n^p = \text{Gram} - \text{Schmidt}(X_n^p, p_h(\boldsymbol{\mu}_p^n))$
12: $X_n^p \leftarrow X_n^p \cup \zeta_n^p, S_n^p \leftarrow S_n^p \cup \{\boldsymbol{\mu}_p^n\}$
13: $[\delta_n^p, \boldsymbol{\mu}_p^n] = \arg \max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_n^p(\boldsymbol{\mu})$
14: $\delta_n = \max(\delta_n^u, \delta_n^p)$

Algorithm 2 Offline stage / state/adjoint, one sampling

1: **procedure** STATE-ADJOINT GREEDY ALGORITHM, ONE SAMPLING
Require: $n_{\max}, \varepsilon_{RB}^{tol}, \Xi_{train} \subset \mathcal{P}, \boldsymbol{\mu}^1 \in \mathcal{P}$
Ensure: RB state and adjoint spaces X_n^u, X_n^p
2: $S_n = \emptyset, X_n^u = \emptyset,$
3: $n = 0, \delta_0 = \varepsilon_{RB}^{tol} + 1$
4: **while** $n < n_{\max}$ and $\delta_n > \varepsilon_{RB}^{tol}$ **do**
5: $n \leftarrow n + 1$
6: compute $u_h(\boldsymbol{\mu}^n), p_h(\boldsymbol{\mu}^n)$
7: $\zeta_n^u = \text{Gram} - \text{Schmidt}(X_n^u, u_h(\boldsymbol{\mu}^n)), X_n^u \leftarrow X_n^u \cup \zeta_n^u$
8: $\zeta_n^p = \text{Gram} - \text{Schmidt}(X_n^p, p_h(\boldsymbol{\mu}^n)), X_n^p \leftarrow X_n^p \cup \zeta_n^p$
9: $S_n \leftarrow S_n \cup \{\boldsymbol{\mu}^n\}$
10: $[\delta_n, \boldsymbol{\mu}^n] = \arg \max_{\boldsymbol{\mu} \in \Xi_{train}} (\Delta_n^u(\boldsymbol{\mu}), \Delta_n^p(\boldsymbol{\mu}))$

Algorithm 3 Offline stage / sensitivities

1: **procedure** STATE-SENSITIVITIES GREEDY ALGORITHM
Require: $n_{\max}, \varepsilon_{RB}^{tol}, \Xi_{train} \subset \mathcal{P}, \boldsymbol{\mu}_u^1, \boldsymbol{\mu}_1^1, \dots, \boldsymbol{\mu}_d^1 \in \mathcal{P}$
Ensure: RB state and sensitivity spaces $X_n^u, X_n^k, k = 1, \dots, d$

- 2: $S_n^u = \emptyset, X_n^u = \emptyset, S_n^k = \emptyset, X_n^k = \emptyset, k = 1, \dots, d$
- 3: $n = 0, \delta_0 = \varepsilon_{RB}^{tol} + 1$
- 4: **while** $n < n_{\max}$ and $\delta_n > \varepsilon_{RB}^{tol}$ **do**
- 5: $n \leftarrow n + 1$
- 6: compute $u_h(\boldsymbol{\mu}_u^n)$
- 7: $\zeta_n^u = \text{Gram} - \text{Schmidt}(X_n^u, u_h(\boldsymbol{\mu}_u^n))$
- 8: $X_n^u \leftarrow X_n^u \cup \zeta_n^u, S_n^u \leftarrow S_n^u \cup \{\boldsymbol{\mu}_u^n\}$
- 9: $[\delta_n^u, \boldsymbol{\mu}_u^n] = \arg \max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_n^u(\boldsymbol{\mu})$
- 10: **for** $k = 1 : d$ **do**
- 11: compute $\frac{\partial u_h}{\partial \mu_k}(\boldsymbol{\mu}_k^n)$
- 12: $\zeta_n^k = \text{Gram} - \text{Schmidt}(X_n^k, \frac{\partial u_h}{\partial \mu_k}(\boldsymbol{\mu}_k^n))$
- 13: $X_n^k \leftarrow X_n^k \cup \zeta_n^k, S_n^k \leftarrow S_n^k \cup \{\boldsymbol{\mu}_k^n\}$
- 14: $[\delta_n^k, \boldsymbol{\mu}_k^n] = \arg \max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_n^k(\boldsymbol{\mu})$
- 15: $\delta_n = \max(\delta_n^u, \delta_n^1, \dots, \delta_n^d)$

4.4 Reduced optimization

We now summarize the reduced computational procedure for the solution of a generic optimization problem through the RB method. This procedure is based on a suitable *offline-online splitting*, ensured by the affine parameter dependence assumption (4). RB spaces are thus built during a first expensive stage, performed *offline*, whereas the *online* stage consists in the optimization procedure, which requires a large number of evaluations of the reduced cost functional $\tilde{J}(u_n(\boldsymbol{\mu}), \boldsymbol{\mu})$ and the reduced descent direction $\mathbf{d}_n(\boldsymbol{\mu})$. Each evaluation entails the solution of the reduced state and adjoint problem; see Algorithm 4.

Algorithm 4 Online procedure

1: **procedure** REDUCED DESCENT METHOD
Require: $n_{\max}, \varepsilon_{opt}^{tol}, \boldsymbol{\mu}^{(0)} \in \mathcal{P}, \mathbf{d}_n^{(0)} \in \mathcal{P}$
Ensure: RB optimal solution $\hat{\boldsymbol{\mu}}_n$

- 2: $n = 0$
- 3: **while** $n < n_{\max}$ and $\|\nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu}_n)\| > \varepsilon_{opt}^{tol}$ **do**
- 4: $n \leftarrow n + 1$
- 5: compute $u_n(\boldsymbol{\mu}^{(k)}), p_n(\boldsymbol{\mu}^{(k)})$; evaluate $J_n(\boldsymbol{\mu}^{(k)}), \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu}^{(k)})$
- 6: **if** Gradient method **then**
- 7: evaluate σ_k (Armijio rule / line-search)
- 8: $\mathbf{d}_n^{(k)} = -\sigma_k \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu}^{(k)})$
- 9: **if** quasi-Newton method **then**
- 10: update $H_{dfp}^{(k)}$ (DFP formula)
- 11: $\mathbf{d}_n^{(k)} = -H_{dfp}^{(k)}(\boldsymbol{\mu}^{(k)}) \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu}^{(k)})$
- 12: **if** Newton method **then**
- 13: compute $\frac{\partial u_n}{\partial \mu_i}(\boldsymbol{\mu}^{(k)}), i = 1, \dots, d$ and evaluate RB Hessian $H_{J,n}(\boldsymbol{\mu}^{(k)})$
- 14: $\mathbf{d}_n^{(k)} = -(H_{J,n}(\boldsymbol{\mu}^{(k)}))^{-1} \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu}^{(k)})$
- 15: $\boldsymbol{\mu}^{(k+1)} = P_{\mathcal{H}_{ad}}(\boldsymbol{\mu}^{(k)} + \mathbf{d}_n^{(k)})$
- 16: $k \leftarrow k + 1$
- 17: $\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}^{(k+1)}$

The construction of the reduced-order algebraic structures required to assemble the reduced state and adjoint problems (27) can be performed very efficiently, relying on the $\boldsymbol{\mu}$ -independent high-fidelity structures. Being able to express high-fidelity structures decoupling $\boldsymbol{\mu}$ -dependent functions and $\boldsymbol{\mu}$ -independent structures hinges upon the affine parameter dependence and is a standard procedure in the context of RB approximation, see, e.g. [35, 37].

In this work, we want to show that a reduced order optimization framework based on a state-adjoint approach ensures the best trade-off between accuracy and efficiency. By relying on a RB approximation for both the state and the adjoint problems, we are able to perform efficient evaluations of the cost functional and its gradient, thus enabling to implement a reduced order gradient or quasi-Newton method. Instead, if willing at implementing a reduced order Newton method, the RB approximation of sensitivity equations is required in order to evaluate the Hessian of the cost functional efficiently. This has a remarkable computational impact, as it implies increasing costs associated with the *offline* procedure. Moreover, we compare different descent methods, assessing their computational speedup and the break-even between offline and online costs.

5 A posteriori error estimation

A key ingredient of the proposed framework is the capability to provide rigorous, sharp and inexpensive error bounds for any quantity involved in the optimization process. We thus manage to keep the error between high-fidelity and reduced-order quantities under control at each optimization step, preventing the reduced optimization framework to provide inaccurate results. We highlight that evaluating efficiently a tight lower-bound $\alpha_h(\boldsymbol{\mu}) \geq \alpha_h^{LB}(\boldsymbol{\mu}) > 0 \forall \boldsymbol{\mu} \in \mathcal{P}$ of the $\boldsymbol{\mu}$ -dependent, high-fidelity stability factor $\alpha_h(\boldsymbol{\mu})$ plays a key role in the a posteriori error bounds; to do this, we can rely on either the successive constraint method [16] or a suitable interpolation procedure [25, 30].

5.1 State, adjoint and sensitivities approximation

For the sake of completeness, we first report some well-known results related with the error bound on the state and adjoint solutions, see, e.g. [35, 37]. Let us denote by

$$\begin{aligned} r_u(v; \boldsymbol{\mu}) &= f(v; \boldsymbol{\mu}) - a(u_n(\boldsymbol{\mu}), v; \boldsymbol{\mu}) & \forall v \in X_h \\ r_p^n(v; \boldsymbol{\mu}) &= dg(v, u_n(\boldsymbol{\mu}); \boldsymbol{\mu}) - a(v, p_n(\boldsymbol{\mu}); \boldsymbol{\mu}) & \forall v \in X_h \end{aligned} \quad (28)$$

the high-fidelity residuals for the state and the adjoint problem, evaluated over the RB solution.

Proposition 5.1. *For any $\boldsymbol{\mu} \in \mathcal{P}$, the error between the high-fidelity and the RB approximation of the state (resp. adjoint) solution is bounded by*

$$\|u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu})\|_X \leq \Delta_n^u(\boldsymbol{\mu}) := \frac{\|r_u(\cdot; \boldsymbol{\mu})\|_{X'}}{\alpha_h^{LB}(\boldsymbol{\mu})}, \quad (29)$$

$$\|p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu})\|_X \leq \Delta_n^p(\boldsymbol{\mu}) := \frac{\|r_p^n(\cdot; \boldsymbol{\mu})\|_{X'}}{\alpha_h^{LB}(\boldsymbol{\mu})}. \quad (30)$$

□

Note that the lower bound $\alpha_h^{LB}(\boldsymbol{\mu})$ is the same in both estimates, and that the adjoint residual depend on the RB approximation of both state and the adjoint problems. Moreover, we can write

$$r_p(v; \boldsymbol{\mu}) = dg(v, u_h(\boldsymbol{\mu}); \boldsymbol{\mu}) - a(v, p_n(\boldsymbol{\mu}); \boldsymbol{\mu}) = r_p^n(v; \boldsymbol{\mu}) + dg(v, u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu}); \boldsymbol{\mu})$$

where the second term at the right-hand side can be easily bounded by using (29). It is indeed straightforward to obtain an error estimate for the (first-order) sensitivity equations. Subtracting the correspondent high-fidelity and reduced sensitivities (11), we obtain, for any $i = 1, \dots, d$:

$$a\left(\frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i} - \frac{\partial u_n(\boldsymbol{\mu})}{\partial \mu_i}, v_h; \boldsymbol{\mu}\right) = \frac{\partial a}{\partial \mu_i}(u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) \quad \forall v_h \in X_h.$$

Then, by using (29) and the stability estimate provided by the Lax-Milgram lemma, we obtain

$$\left\| \frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i} - \frac{\partial u_n(\boldsymbol{\mu})}{\partial \mu_i} \right\|_V \leq \Delta_n^i(\boldsymbol{\mu}) := \frac{M_{a, \mu_i}(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} \Delta_n^u(\boldsymbol{\mu}),$$

where $M_{a, \mu_i}(\boldsymbol{\mu})$ is the continuity constants associated to the form $\frac{\partial a}{\partial \mu_i}(\cdot, \cdot; \boldsymbol{\mu})$ defined by (23).

5.2 Cost functional

We now derive an error bound for a generic quadratic functional of $u(\boldsymbol{\mu})$ – such as the one appearing in (3) – based on a “primal-dual” approach. In a “primal only” approach such a bound would only depend on $\Delta_n^u(\boldsymbol{\mu})$ and would lose the so-called “quadratic effect” – see, e.g., [37, 33]).

Even if we do not need to introduce the Lagrangian formalism for deriving the optimality conditions (17)–(18), we find it useful for deriving the a posteriori error estimation on the cost functional provided by the following proposition, inspired by the *goal-oriented* analysis of [4]:

Proposition 5.2. *Let $J(\boldsymbol{\mu}) = g(u(\boldsymbol{\mu}), u(\boldsymbol{\mu}); \boldsymbol{\mu})$ be a quadratic functional, where $g : X \times X \times \mathcal{P} \rightarrow \mathbb{R}$ is a symmetric coercive bilinear form, for any $\boldsymbol{\mu} \in \mathcal{P}$. Let us denote by $J_h(\boldsymbol{\mu}) = g(u_h(\boldsymbol{\mu}), u_h(\boldsymbol{\mu}); \boldsymbol{\mu})$ and $J_n(\boldsymbol{\mu}) = g(u_n(\boldsymbol{\mu}), u_n(\boldsymbol{\mu}); \boldsymbol{\mu})$, respectively. Then, for any $\boldsymbol{\mu} \in \mathcal{P}$,*

$$|J_h(\boldsymbol{\mu}) - J_n(\boldsymbol{\mu})| \leq \Delta_n^J(\boldsymbol{\mu}) := \alpha_h^{LB}(\boldsymbol{\mu}) \Delta_n^u(\boldsymbol{\mu}) \Delta_n^p(\boldsymbol{\mu}). \quad (31)$$

Proof. For the sake of proving this result, we rely on the introduction of the Lagrangian functional associated to the constrained optimization problem (6). We rewrite $|J_h(\boldsymbol{\mu}) - J_n(\boldsymbol{\mu})|$ by combining in an appropriate manner the state and adjoint problems in order to identify the residuals. This yields

$$\begin{aligned} |J_h(\boldsymbol{\mu}) - J_n(\boldsymbol{\mu})| &= |J_h(\boldsymbol{\mu}) - a(u_h, p_h; \boldsymbol{\mu}) + f(p_h; \boldsymbol{\mu}) - J_n(\boldsymbol{\mu}) + a(u_n, p_n; \boldsymbol{\mu}) - f(p_n; \boldsymbol{\mu})| \\ &= |\mathcal{L}(u_h, p_h; \boldsymbol{\mu}) - \mathcal{L}(u_n, p_n; \boldsymbol{\mu})| \\ &= \frac{1}{2} \left| \mathcal{L}'((u_h, p_h); \mathbf{e}_h; \boldsymbol{\mu}) + \mathcal{L}'((u_n, p_n); \mathbf{e}_h; \boldsymbol{\mu}) + \int_0^1 \mathcal{L}'''((u_n, p_n) + t\mathbf{e}_h; \mathbf{e}_h; \boldsymbol{\mu}) dt \right|. \end{aligned}$$

The first and the third terms are null respectively for the assumption that the *high-fidelity* solution is the optimal one ($\mathcal{L}' = 0$) and for the linearity of $a(\cdot, \cdot; \boldsymbol{\mu})$ and of $F(\cdot; \boldsymbol{\mu})$. Using the definition (28) of the residuals we obtain, for any $\boldsymbol{\mu} \in \mathcal{P}$,

$$\begin{aligned} |J_h(\boldsymbol{\mu}) - J_n(\boldsymbol{\mu})| &\leq \frac{1}{2} |r_u(u_n(\boldsymbol{\mu}); p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu}))| + \frac{1}{2} |r_p(p_n(\boldsymbol{\mu}); u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu}))| \\ &\leq \frac{1}{2} \|r_u\|_{X'} \|p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu})\|_X + \frac{1}{2} \|r_p\|_{X'} \|u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu})\|_X. \end{aligned}$$

thanks to Cauchy-Schwarz inequality; (31) follows by (29)–(30). \square

5.3 Gradient of the cost functional

Being able to estimate the error on the gradient of the cost functional ensures that the descent directions $\mathbf{d}_n^{(k)}$ of the reduced minimization algorithm are indeed very close to the directions $\mathbf{d}_h^{(k)}$ we would have built by considering the high-fidelity approximation. Estimating the error $\|\nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu})\|_{\mathbb{R}^d}$ is usually quite involved and results in not so tight error bounds.

In this work we take advantage of the primal-dual approach in order to improve this estimate. For any $i = 1, \dots, d$, let us denote by

$$\frac{\partial f}{\partial \mu_i}(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \frac{\partial \Theta_q^f(\boldsymbol{\mu})}{\partial \mu_i} f_q(v), \quad (32)$$

$$\frac{\partial s}{\partial \mu_i}(u, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \frac{\partial \Theta_q^g(\boldsymbol{\mu})}{\partial \mu_i} s_q(u, v), \quad \frac{\partial l}{\partial \mu_i}(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \frac{\partial \Theta_q^g(\boldsymbol{\mu})}{\partial \mu_i} l_q(v), \quad (33)$$

and by $M_{f, \mu_i}(\boldsymbol{\mu})$, $M_{s, \mu_i}(\boldsymbol{\mu})$, $M_{l, \mu_i}(\boldsymbol{\mu})$ the continuity constants of the forms defined in (32)–(33).

Proposition 5.3. *Under the assumptions of Proposition 5.2, for any $\boldsymbol{\mu} \in \mathcal{P}$*

$$\|\nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu})\|_{\mathbb{R}^d} \leq \Delta_n^{\nabla J}(\boldsymbol{\mu}) := \left(\sum_{i=1}^d (C_i^u(\boldsymbol{\mu}) \Delta_n^u(\boldsymbol{\mu}) + C_i^p(\boldsymbol{\mu}) \Delta_n^p(\boldsymbol{\mu}))^d \right)^{1/d}, \quad (34)$$

where

$$\begin{aligned} C_i^u(\boldsymbol{\mu}) &= M_a(\boldsymbol{\mu}) \|p_n(\boldsymbol{\mu})\|_X \frac{M_{a,\mu_i}(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} + (2M_{s,\mu_i} \frac{M_f(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} + M_{l,\mu_i}(\boldsymbol{\mu})), \\ C_i^p(\boldsymbol{\mu}) &= M_{f,\mu_i}(\boldsymbol{\mu}) + M_{a,\mu_i}(\boldsymbol{\mu}) \frac{M_f(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})}. \end{aligned}$$

Proof. For each component μ_i , $i = 1, \dots, d$, using the chain rule (8) and the adjoint problem (17) we obtain

$$\begin{aligned} \left| \frac{DJ_h}{D\mu_i} - \frac{DJ_n}{D\mu_i} \right| &\leq \left| \left(\frac{\partial J_h}{\partial u_h}, \frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i} \right) - \left(\frac{\partial J_n}{\partial u_n}, \frac{\partial u_n(\boldsymbol{\mu})}{\partial \mu_i} \right) \right| + \left| \frac{\partial J_h}{\partial \mu_i} - \frac{\partial J_n}{\partial \mu_i} \right| \\ &= \left| a^* \left(p_h(\boldsymbol{\mu}), \frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i}; \boldsymbol{\mu} \right) - a^* \left(p_n(\boldsymbol{\mu}), \frac{\partial u_n(\boldsymbol{\mu})}{\partial \mu_i}; \boldsymbol{\mu} \right) \right| + \left| \frac{\partial J_h}{\partial \mu_i} - \frac{\partial J_n}{\partial \mu_i} \right| = I + II \end{aligned}$$

Henceforth, we proceed separately for the two terms, given by the error weighted by the bilinear form and by the term concerning the derivative with respect to μ_i , respectively. For the former term, we add and subtract $a(p_n(\boldsymbol{\mu}), \partial u_h(\boldsymbol{\mu})/\partial \mu_i; \boldsymbol{\mu})$ and apply Cauchy-Schwarz inequality, obtaining

$$I \leq \left| a^* \left(p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu}), \frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i}; \boldsymbol{\mu} \right) \right| + \left| a^* \left(p_n(\boldsymbol{\mu}), \frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i} - \frac{\partial u_n(\boldsymbol{\mu})}{\partial \mu_i}; \boldsymbol{\mu} \right) \right| = I_a + I_b.$$

We estimate I_b using the continuity of $a^*(\cdot, \cdot; \boldsymbol{\mu})$, that is,

$$I_b \leq M_a(\boldsymbol{\mu}) \|p_n(\boldsymbol{\mu})\|_X \left\| \frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i} - \frac{\partial u_n(\boldsymbol{\mu})}{\partial \mu_i} \right\|_X \leq M_a(\boldsymbol{\mu}) \|p_n(\boldsymbol{\mu})\|_X \frac{M_{a,\mu_i}(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} \Delta_n^u(\boldsymbol{\mu});$$

while, using the problem for the (first-order) sensitivity, we obtain

$$\begin{aligned} I_a &= \left| a \left(\frac{\partial u_h(\boldsymbol{\mu})}{\partial \mu_i}, p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu}); \boldsymbol{\mu} \right) \right| = \left| \frac{\partial f}{\partial \mu_i}(p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu})) - \frac{\partial a}{\partial \mu_i}(u_h(\boldsymbol{\mu}), p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu}); \boldsymbol{\mu}) \right| \\ &\leq M_{f,\mu_i}(\boldsymbol{\mu}) \|p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu})\|_X + M_{a,\mu_i}(\boldsymbol{\mu}) \frac{M_f(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} \|p_h(\boldsymbol{\mu}) - p_n(\boldsymbol{\mu})\|_X \\ &\leq \left(M_{f,\mu_i}(\boldsymbol{\mu}) + M_{a,\mu_i}(\boldsymbol{\mu}) \frac{M_f(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} \right) \Delta_n^p(\boldsymbol{\mu}). \end{aligned}$$

Using the continuity of the operators in the cost functional and its quadratic form, we obtain

$$\begin{aligned} II &= \left| \frac{\partial s}{\partial \mu_i}(u_h(\boldsymbol{\mu}), u_h(\boldsymbol{\mu}); \boldsymbol{\mu}) + \frac{\partial l}{\partial \mu_i}(u_h(\boldsymbol{\mu})) - \frac{\partial s}{\partial \mu_i}(u_n(\boldsymbol{\mu}), u_n(\boldsymbol{\mu})) - \frac{\partial l}{\partial \mu_i}(u_n(\boldsymbol{\mu})) \right| \\ &\leq M_{s,\mu_i} (\|u_h(\boldsymbol{\mu})\|_X + \|u_n(\boldsymbol{\mu})\|_X) \|u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu})\|_X + M_{l,\mu_i} \|u_h(\boldsymbol{\mu}) - u_n(\boldsymbol{\mu})\| \\ &\leq \left(2M_{s,\mu_i}(\boldsymbol{\mu}) \frac{M_f(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} + M_{l,\mu_i}(\boldsymbol{\mu}) \right) \Delta_n^u(\boldsymbol{\mu}). \end{aligned}$$

By combining the previous inequalities, (34) follows. \square

5.4 Optimal solution

Finally, we can provide an estimate for the error $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|_{\mathbb{R}^d}$ between the optimal parameters obtained with the high-fidelity and the reduced optimization framework, respectively. By combining the previous results, we ensure that the reduced optimization method converges to an optimal solution $\hat{\boldsymbol{\mu}}_n$ which is close to the high-fidelity solution $\hat{\boldsymbol{\mu}}_h$ up to a prescribed threshold.

Let us denote by $\boldsymbol{\mu}_h \in \mathcal{P}$ the solution to problem $(\boldsymbol{\mu}\text{-OP}_h)$, such that $\nabla_{\boldsymbol{\mu}} J_h(\hat{\boldsymbol{\mu}}_h) = \mathbf{0}$, by $\mathbb{H}_{J,h}(\boldsymbol{\mu})$ the *high-fidelity* Hessian of the cost functional (see Sect. 3.4) and by $B_r(\boldsymbol{\mu})$ a closed ball with center $\boldsymbol{\mu} \in \mathcal{P}$ and radius $r > 0$. Moreover, let $\hat{\boldsymbol{\mu}}_n \in \mathcal{P}$ be the approximate solution to problem $(\boldsymbol{\mu}\text{-OP}_n)$ fulfilling, for some $\epsilon_J > 0$,

$$\|\nabla_{\boldsymbol{\mu}} J_n(\hat{\boldsymbol{\mu}}_n)\|_{\mathbb{R}^d} \leq \epsilon_J.$$

Theorem 5.4. Denote by

$$\lambda = \|\mathbb{H}_{J,h}(\hat{\boldsymbol{\mu}}_n)^{-1}\|_{\mathbb{R}^{d \times d}}, \quad \epsilon = \|\nabla_{\boldsymbol{\mu}} J_h(\hat{\boldsymbol{\mu}}_n) - \nabla_{\boldsymbol{\mu}} J_n(\hat{\boldsymbol{\mu}}_n)\|_{\mathbb{R}^d} + \epsilon_J.$$

Moreover, define by $L(r) = \sup_{\boldsymbol{\mu} \in \bar{B}_r(\hat{\boldsymbol{\mu}}_n)} \|\mathbb{H}_{J,h}(\hat{\boldsymbol{\mu}}_n) - \mathbb{H}_{J,h}(\boldsymbol{\mu})\|_{\mathbb{R}^{d \times d}}$. Provided that

$$2\lambda L(2\lambda\epsilon) \leq 1 \tag{35}$$

there exists a unique solution $\hat{\boldsymbol{\mu}}_h \in \bar{B}_{2\lambda\epsilon}(\hat{\boldsymbol{\mu}}_n)$ of the optimization problem $(\boldsymbol{\mu}\text{-OP}_h)$, such that

$$\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\| \leq 2\lambda\epsilon. \tag{36}$$

Proof. We do not report the full proof for the sake of space; the interested reader can refer, e.g., to [7, 8]. It is straightforward to see that the solution $\hat{\boldsymbol{\mu}}_h$ to problem $(\boldsymbol{\mu}\text{-OP}_h)$ is a fixed point of the map $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ defined by $\Phi(\boldsymbol{\mu}) = \boldsymbol{\mu} - \mathbb{H}_{J,h}(\hat{\boldsymbol{\mu}}_n)^{-1} \nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu})$, since $\nabla_{\boldsymbol{\mu}} J_h(\hat{\boldsymbol{\mu}}_h) = \mathbf{0}$. The general Brezzi-Rappaz-Raviart theory (see, e.g. [8, Section 2]) provides existence and uniqueness of the fixed point, as well as the error bound (36). \square

Similarly to previous bounds, the error bound (36) combines two terms: λ , which plays the role of a stability factor, and ϵ , which can be considered as the dual norm of the residual of the minimization problem $\nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}) = \mathbf{0}$. Evaluating the (inverse norm of the) high-fidelity Hessian $\mathbb{H}_{J,h}(\cdot)$ entails the solution of the high-fidelity state (and adjoint) problem for $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}_n$; however, this computation has to be performed only when reaching the stopping criterion on the reduced optimization, if we want to certify the optimal solution through a bound of the error $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|_{\mathbb{R}^d}$. A possible alternative in the case when multiple evaluations of the error bound (36) are required would be to rely on interpolation procedures (in the parameter space), similarly to what has been recently proposed in [25] for the fast and reliable approximation of stability factors related to parametrized PDE operators. In the following section we show the effectivity of the proposed error estimates, and we assess the computational performances of our reduced optimization framework.

6 Numerical results

In this section we present some numerical results dealing with quadratic cost functional and elliptic scalar stationary PDE, characterized by physical and/or geometric parameters playing the role of control variables². After performing the offline stage, the inexpensive RB online evaluation takes place at each iteration of the optimization scheme. Although in our results the target z_d is fixed, we highlight that the proposed framework can also be used to solve optimization problems for different target functions z_d (possibly depending on $\boldsymbol{\mu} \in \mathcal{P}$) without requiring a new offline stage.

6.1 Graetz conduction-convection problem

The Graetz problem concerns the forced convection of a heat flow in a pipeline with walls at variable temperature [3]. We deal with a parametrized version of this problem, where $d = 4$ parameters are:

- $\mu_1 \in [1, 5]$ the length of the pipe section;
- $\mu_2 \in [1, 7]$, being μ_2^2 the Péclet number;
- $\mu_3 \in [0, \frac{\pi}{4}]$ the angle of incidence of the advection field;
- $\mu_4 \in [0, \frac{\pi}{2}]$ the amplitude of the inlet Dirichlet boundary condition.

We consider the domain $\tilde{\Omega}(\boldsymbol{\mu}) = (0, 3 + \mu_1) \times (0, 1)$; in the first subregion $(0, 1 + \mu_1) \times (0, 1)$ the flow comes in contact with the hot wall Γ_{in} , whereas in the second (observation) region $(1 + \mu_1, 3 + \mu_1) \times (0, 1)$ the fluid flows through cold walls.

²Computations have been run on a laptop with a 2,2 GHz Intel Core i7 processor and 8 GB of RAM.

The state problem (for the temperature variable) reads as follows: find $u(\boldsymbol{\mu})$ such that

$$\begin{cases} -\frac{1}{\mu_2^2}\Delta u(\boldsymbol{\mu}) + \mathbf{b}(\mu_3) \cdot \nabla u(\boldsymbol{\mu}) = 0 & \text{in } \tilde{\Omega}(\mu_1) \\ u(\boldsymbol{\mu}) = 0 & \text{on } \tilde{\Gamma}_D \\ u(\boldsymbol{\mu}) = \cos(\mu_4)(1-x_1)(x_1-1-\mu_1) & \text{on } \tilde{\Gamma}_{in}(\mu_1) \\ u(\boldsymbol{\mu}) = \sin(\mu_4)(1-x_1)(x_1-1-\mu_1) & \text{on } \tilde{\Gamma}_{in}(\mu_1) \\ \frac{\partial u(\boldsymbol{\mu})}{\partial \mathbf{n}} = 0 & \text{on } \tilde{\Gamma}_{out}, \end{cases} \quad (37)$$

where $\mathbf{b}(\mu_3) = [\cos(\mu_3)10x_2(1-x_2), -\sin(\mu_3)]$. Given a target z_d , the goal is to approximate, through the reduced framework, the solution

$$\hat{\boldsymbol{\mu}}_h = \arg \min_{\boldsymbol{\mu} \in \mathcal{P}} J_h(\boldsymbol{\mu}), \quad J_h(\boldsymbol{\mu}) = \frac{1}{2} \|u_h(\boldsymbol{\mu}) - z_d\|_{L^2(\tilde{\Omega}_{obs}(\boldsymbol{\mu}))}^2, \quad (38)$$

the minimum of J_h being reached for $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_{target}$, i.e. $J_h(\hat{\boldsymbol{\mu}}) = 0$. For the sake of numerical test, we set $\boldsymbol{\mu}_{target} = [3, 4, \frac{\pi}{10}, 1]$ and the corresponding high-fidelity solution $u_h(\boldsymbol{\mu}_{target})$ as target z_d .

6.1.1 Offline stage

After mapping the problem (37)-(38) onto a fixed reference domain Ω , we construct a high-fidelity approximation through the Galerkin FE method, using piecewise linear elements; the dimension of the corresponding space is $N_h = 4033$. The affine parametric dependence (4) is recovered through $Q_a = 7$ and $Q_f = 15$ terms, respectively. Then, we turn to the construction of a RB approximation for both the state and the adjoint problem.

For the sake of comparison, we consider the three greedy algorithms introduced in Sect. 4.3. Numerical results presented in Fig. 1 outline two main evidences: the adjoint-based approach enables to obtain a RB space of lower dimension keeping the same accuracy level with respect to the sensitivity-based approach (see Fig. 1, left). Moreover (see Fig. 1, right), the adjoint-based approach performs better when the snapshots for the state and the adjoint problems are computed with respect to parameter values independently sampled (Algorithm 1, two samplings) – that is, the retained parameter sets S_n^u and S_n^p are different (see Fig. 2). For this reason, we consider only Algorithms 1 and 3 for the sake of comparison.

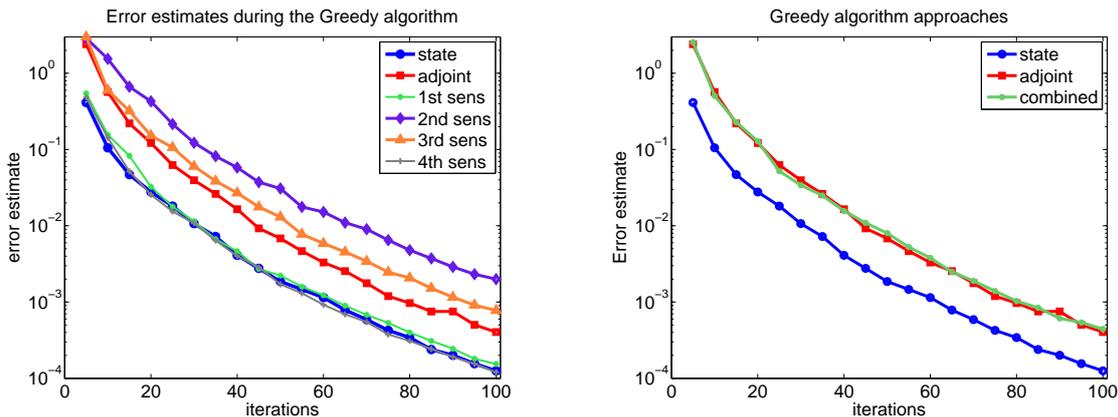


Figure 1: Convergence of the three greedy algorithms presented in Section 4.3.

The three proposed strategies show different *offline* computational costs: indeed, the need of constructing a RB space for each of the $d > 1$ sensitivity equations has a considerable impact, which becomes larger and larger as the desired dimension N_h of the RB spaces increases; see Table 1.

The adjoint-based strategy relying on two independent sampling for the construction of state and adjoint subspaces (Algorithm 1) is the best approach in terms of both efficiency and accuracy.

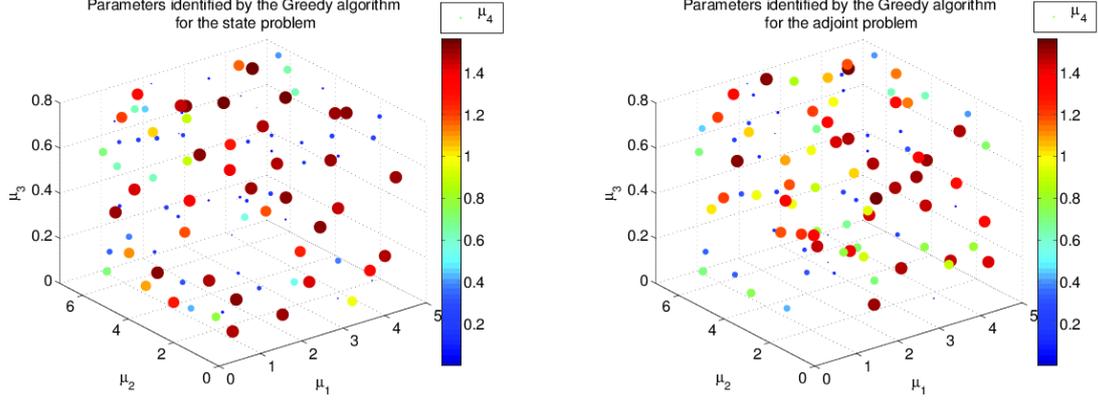


Figure 2: Sampled parameters during the greedy algorithm for state (left) and adjoint (right) problems.

	$N = 30$	$N = 50$	$N = 70$	$N = 90$	$N = 110$
state+adjoint (Alg. 1)	6 m	9.5 m	13.5 m	17.5 m	22 m
state+sensitivities (Alg. 3)	8 m	13 m	19 m	25 m	32 m

Table 1: CPU time (minutes) for the *offline* stage, for different greedy algorithms.

6.1.2 Online stage

We now compare the (reduced) descent methods introduced in Sect. 4.2 – namely, projected gradient, quasi-Newton and Newton methods – prescribing a stopping criterium based on the gradient of the cost functional $\|\nabla \boldsymbol{\mu} J_n\|_{\mathbb{R}^d}$. In a first test case we start from $\boldsymbol{\mu}^{(0)} = (1, 1, 0.01, \frac{\pi}{2})$ and we apply the gradient algorithm by considering either the Armijo rule or the line-search method for choosing the step size. The method converges to the optimal value $\boldsymbol{\mu}_{target}$, although it requires a large number of iterations (and subiterations for the step size criterium); the corresponding state and adjoint solutions are reported in Fig. 3.

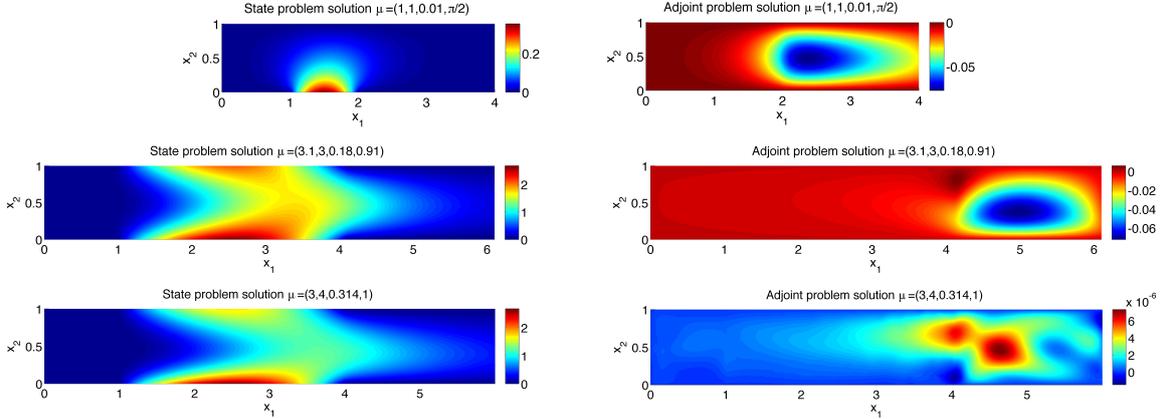


Figure 3: RB solutions of state and adjoint problems for $\boldsymbol{\mu} = \boldsymbol{\mu}^{(0)}$ (top), $\boldsymbol{\mu} = \boldsymbol{\mu}^{(3)} = (3.1, 3, 0.18, 0.91)$ (middle), $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}_n$ (bottom).

Provided that the cost functional is nonconvex with respect to $\boldsymbol{\mu}$, the reduced quasi-Newton method converges only if the starting value $\boldsymbol{\mu}^{(0)}$ is sufficiently close to the optimal value. As a consequence, we exploit the gradient algorithm (globally convergent) to generate a suitable starting point for the quasi-Newton method. In this case, we run the gradient method until it satisfies the stopping criterium with tolerance $tol = 10^{-2}$, then we turn to the quasi-Newton method. This combination allows us to achieve very high accuracy with a limited number of iterations (e.g. 18 iterations of gradient method with line-search rule plus 7 of Newton method in order to reach a tolerance of 10^{-9} in the stopping criterium). A stagnation effect on the cost functional, yielding

a convergence of the reduced descent method to an optimal value $\hat{\boldsymbol{\mu}}_n \rightarrow \boldsymbol{\mu}_{target}$, is observed in the case of a poor RB approximation, see Fig. 4, for both Newton and quasi-Newton methods; a similar conclusion is found for the gradient method.

Hence, already for RB subspaces of relatively small dimension ($N = 75, 100$), a remarkable accuracy is reached. Concerning the CPU time required to perform optimization, we pursue a computational speedup ranging from 17 in the Newton case to 93 in the gradient case. The smallest number of iterations required to reach convergence is obtained in the Newton case (see Table 2).

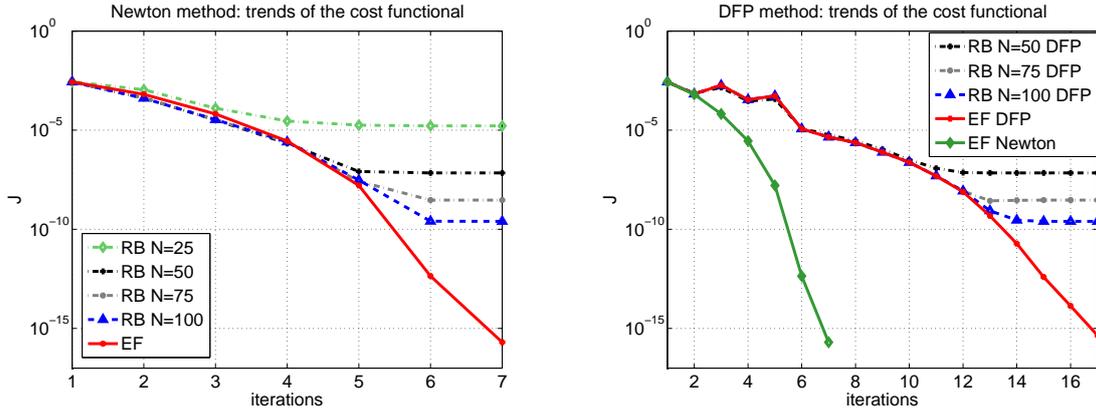


Figure 4: Cost functional values at each iteration of Newton (left) and quasi-Newton (right) methods for different RB dimensions N and comparison with the high-fidelity FE approximation.

	$\#it_{RB}$	$CPUtime/it_{RB}$	$CPUtime/it_{FE}$
Gradient (Armijio)	2502	0.0104 s	0.9327 s
Gradient (line search)	173	1.926 s	59.027 s
Gradient (line search) + quasi-Newton	18 + 17	0.0088 s	0.1249 s
Gradient (line search) + Newton	18 + 7	0.0235 s	0.4047 s

Table 2: Number of iterations and computational times of the Gradient method with Armijio rule and inexact line-search in $(10^{-2}, 50)$ with 500 samples, of quasi-Newton and Newton methods ($N = 100$ RB functions).

The best option concerning computational efficiency is the quasi-Newton method. For the case at hand, using the gradient method to generate a suitable starting point $\tilde{\boldsymbol{\mu}}_0$, the quasi-Newton (DFP) method converges in 17 iterations to the sub-optimal value $\hat{\boldsymbol{\mu}}_n = (2.99996, 4.00002, 0.31416, 0.99996)$, with cost functional $J_n(\hat{\boldsymbol{\mu}}_n) = 2.51 \cdot 10^{-10}$. In particular, we initialize the surrogate of the Hessian matrix $\mathbb{H}_J(\boldsymbol{\mu})$ with a diagonal matrix of elements equal to the step size identified by the inexact line search (as in a step of the gradient method). Starting from the same initial guess $\tilde{\boldsymbol{\mu}}_0$, the Newton method converges in 7 iterations to $\hat{\boldsymbol{\mu}}_n = (2.99996, 4.00003, 0.314164, 0.99996)$, with cost functional $J_n(\hat{\boldsymbol{\mu}}_n) = 2.51 \cdot 10^{-10}$. We remark that the accuracy of the reduced-order model (in terms of the number n of basis functions) has a relevant role in achieving the minimum value of the cost functional: the larger n , the smaller $J_n(\hat{\boldsymbol{\mu}}_n)$, see Fig. 4 – recall that $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_{target}$. In addition to its better efficiency with respect to the sensitivity-based approach, the adjoint-based method yields inexpensive a posteriori error bounds for the cost functional and its gradient. In this case, the effectivities

$$\eta_J(\boldsymbol{\mu}) = \frac{\Delta_n^J(\boldsymbol{\mu})}{|J_h(\boldsymbol{\mu}) - J_n(\boldsymbol{\mu})|}, \quad \eta_{\nabla J}(\boldsymbol{\mu}) = \frac{\Delta_n^{\nabla J}(\boldsymbol{\mu})}{\|\nabla_{\boldsymbol{\mu}} J_h(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}} J_n(\boldsymbol{\mu})\|_{\mathbb{R}^d}}$$

for J and ∇J , respectively, are indeed very close to 1 – that is, the proposed error bounds show to be very tight for the case at hand, see Fig. 5. This is in our opinion a remarkable result, since error bounds usually show large effectivities in the case of optimal control problems, see e.g. [9, 10].

	η_J	$\eta_{\nabla J}$
$N = 30$	2.0716	4.3478
$N = 65$	1.2286	4.5187
$N = 90$	1.1121	4.1614

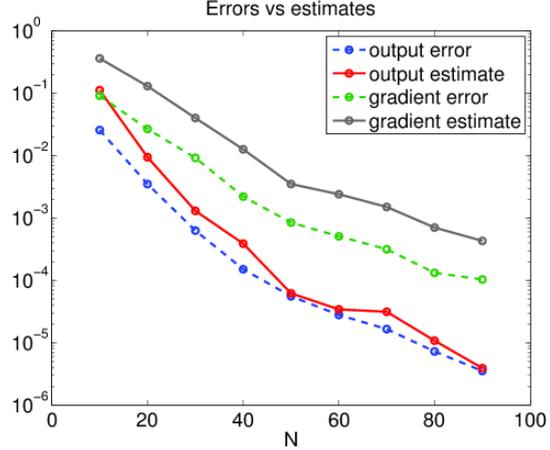


Figure 5: A posteriori error bounds for the cost functional and its gradient: effectivities (left) and behavior with respect to RB dimension N , average on a sample $\Xi_{test} \subset \mathcal{P}$ of dimension 100 (right).

Finally, to assess the accuracy of the proposed framework, we evaluate the error $\|\hat{\mu}_j - \hat{\mu}_n\|$ between the high-fidelity and the reduced-order solution of the optimization problems. We also verify from a computational standpoint the result of Theorem 5.4, focusing on the dependence of the error bound (36) on the RB dimension n and the tolerance ε_{opt}^{tol} of the stopping criterium. These results confirm that it is fundamental to achieve a very good approximation of the optimality conditions in a neighborhood of the optimal solution in order to guarantee a good approximation of the optimal parameters. In fact, the estimate is only certified for a number of basis greater than 90, which decreases noticeably the error $\|\nabla_{\mu} J_n(\mu) - \nabla_{\mu} J_h(\mu)\|_{\mathbb{R}^d}$ (and so ϵ). On the other hand for tolerances smaller than 10^{-5} and 100 basis the estimate is always verified.

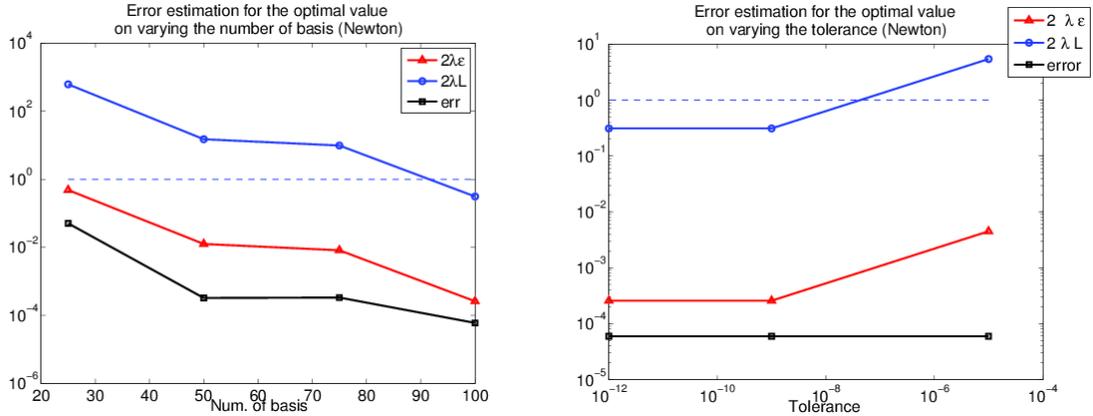


Figure 6: A posteriori error estimates for the optimal parameters (Newton method started with line search gradient iterations) varying the number of basis N at fixed tolerance $\epsilon_J = 10^{-6}$ of the stopping criterium for the optimization (left) and varying ϵ_J at fixed $N = 100$ (right).

6.2 NACA airfoil shape design

We now turn to the optimal design of NACA airfoils. Parametrizing the airfoil shape enables to cast the problem into the PDE-constrained parametric optimization framework presented in this paper. In our case we deal with $d = 3$ parameters, namely

- $\mu_1 \in [0.04, 0.24]$ the maximum thickness of the airfoil;
- $\mu_2 \in [0, 0.012]$ the curvature of the wing profile;
- $\mu_3 \in [-0.45\pi, 0.45\pi]$ the angle of incidence of the flow w.r.t. the chord of the airfoil.

We consider a set of affine maps (naturally induced by the selected parameters) defined over a domain decomposition in order to recover the affine parametric dependence property (4), as shown e.g. in [37]. This yields $Q_a = 184$ and $Q_f = 188$ terms in expression (4), respectively. Other options to deal with more general nonaffine maps – such as (discrete) empirical interpolation – can also be used, see e.g. [24]. We describe the flow around the airfoil through a potential model, leading to the following state problem for the velocity potential: find $\phi \in X$ such that

$$\begin{cases} -\Delta\phi = 0 & \text{in } \tilde{\Omega}(\mu_1, \mu_2) \\ \phi = \phi_{ref}(\mu_3) & \text{on } \tilde{\Gamma}_{out} \\ \frac{\partial\phi}{\partial n} = \mathbf{u}_{in}(\mu_3) \cdot \mathbf{n} & \text{on } \tilde{\Gamma}_{in} \\ \frac{\partial\phi}{\partial n} = 0 & \text{on } \tilde{\Gamma}_{wing}(\mu_1, \mu_2). \end{cases} \quad (39)$$

The velocity-pressure formulation follows by the identity $\mathbf{u} = -\nabla\phi$ and the Bernoulli equation. Following [28, Chapter 2], we aim at reconstructing the airfoil shape corresponding to a given target velocity field in the rear $\tilde{\Omega}_{obs} \subset \Omega$ of the airfoil, that is, to approximate

$$\hat{\boldsymbol{\mu}}_h = \arg \min_{\boldsymbol{\mu} \in \mathcal{P}} J_h(\boldsymbol{\mu}), \quad J_h(\boldsymbol{\mu}) = \frac{1}{2} \int_{\tilde{\Omega}_{obs}} |\nabla(\phi_h(\boldsymbol{\mu}) - z_d)|^2 d\Omega. \quad (40)$$

Here the target z_d is the high-fidelity solution $\phi_h(\boldsymbol{\mu}_{target})$ with $\boldsymbol{\mu}_{target} = [0.15, 0.006, \frac{\pi}{6}]$.

6.2.1 Offline stage

The approximation of the problem (39)–(40), mapped onto the fixed reference domain Ω , is firstly obtained through the Galerkin FE method with linear finite elements, resulting in a high-fidelity approximation of dimension $N_h = 10886$. Then, we exploit the greedy Algorithms 1 and 3 for constructing the RB spaces for state and adjoint (resp., sensitivity) problems. Also in this case, the adjoint-based method seems to be more effective in terms of accuracy and efficiency, see Fig. 7 and Table 3; the retained snapshots are reported in Fig. 8.

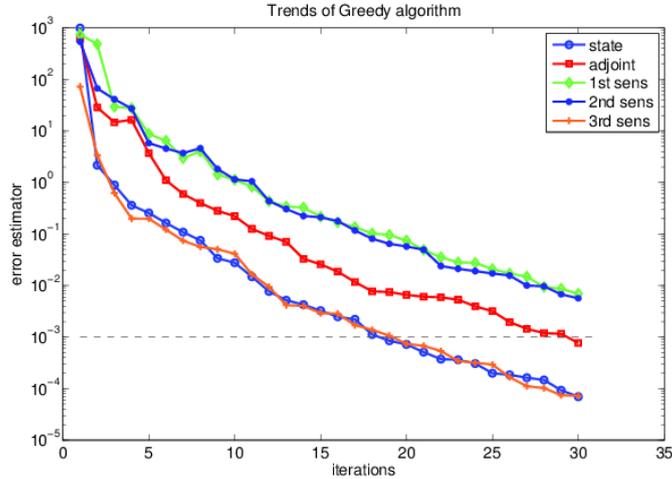


Figure 7: Convergence of greedy Algorithm 1 (resp. 3) for state and adjoint (resp. sensitivity) problems.

Basis computation	$N = 10$	$N = 20$	$N = 30$
state+adjoint (Alg. 1)	10 m	20 m	31 m
state+sensitivities (Alg. 3)	24 m	47 m	72 m

Table 3: CPU time for the *offline* stage, for different greedy algorithms.

As expected, the offline computational cost are higher with respect to the previous case, because of the larger dimension N_h of the FE space and the larger number Q_a, Q_f of terms in the affine expansion. Indeed, this yields a higher complexity in the $\boldsymbol{\mu}$ -dependence of the PDE solutions and additional costs in the assembling of $\boldsymbol{\mu}$ -independent arrays.

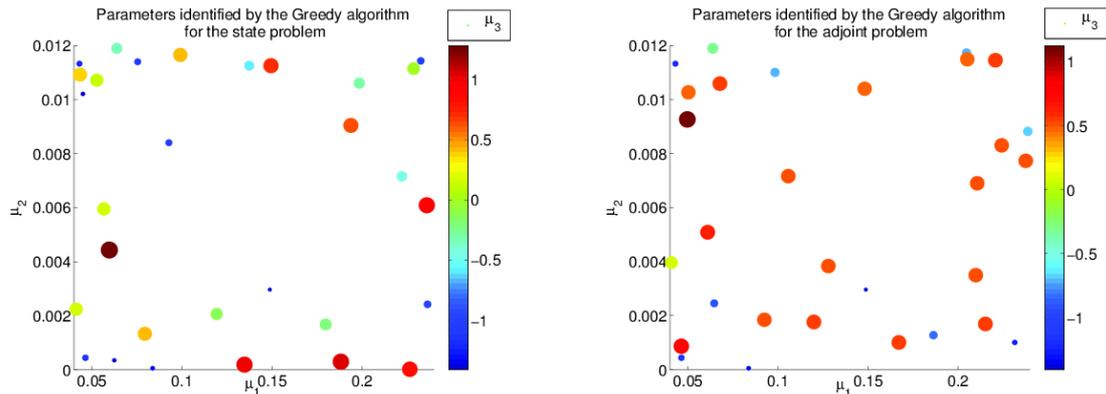


Figure 8: Sampled parameters during the greedy algorithm for state (left) and adjoint (right) problems.

6.2.2 Online stage

Starting from the initial guess $\boldsymbol{\mu}^{(0)} = (0.08, 0.012, -\frac{\pi}{4})$, we compare the results obtained with the reduced and the high-fidelity optimization framework (see Algorithm 4), testing different options concerning the descent directions. For the case at hand, the Newton method converges considering $\boldsymbol{\mu}_n^{(0)} = (0.08, 0.012, -\frac{\pi}{4})$ as initial guess. The iterations generated by the high-fidelity and the reduced Newton method (varying the number of basis functions) are displayed in Fig. 9, top and Fig. 10, left. Clearly, the results obtained with the reduced optimization algorithm get closer and closer to the high-fidelity solution as the number of RB functions increases. The RB approximation of both state and adjoint problems are reported in Fig. 11 for different parameter values (initial guess, second iteration of the Newton method and optimal parameter value); the corresponding velocity and pressure fields are reported in Fig. 12.

On the other hand, the quasi-Newton method (with $H^{(0)}$ equal to the identity matrix) requires about the same number of iterations to converge as the Newton method, yet entailing a remarkably lower computational effort (both during the *offline* and *online* phase). The iterations generated by the high-fidelity and the reduced quasi-Newton method are displayed in Fig. 9, bottom and Fig. 10, right.

In order to control the reduction error at each step of the minimization procedure, we can rely on the proposed error estimates. In particular, we compute the effectivities of the error estimator for the cost functional and its gradient (see Fig. 13). As in the previous example, the goal-oriented analysis provides a tight error bound for the cost functional J , and a reliable (but less tight) error bound in the case of the gradient ∇J . Moreover, we can assess the influence of reduction errors on the optimal solution thanks to the result of Theorem 5.4. We report in Fig. 14 the error bound (36) for different RB dimensions n and tolerances ε_{opt}^{tol} of the stopping criterium.

By considering a RB dimension greater than 9 and a tolerance $\varepsilon_{opt}^{tol} < 5 \cdot 10^{-3}$, the hypothesis of Theorem 5.4 are verified. Hence, the error bound (36) provide a reliable estimate of the error $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|_{\mathbb{R}^p}$ on the optimal parameter values.

Finally, we report some details concerning the computational costs, by considering as starting point $\boldsymbol{\mu}_n^{(0)} = (0.22, 0.01, 0.4\pi)$ with a tolerance $\varepsilon_{opt}^{tol} = 10^{-10}$ for the reduced optimization procedure. As expected, the gradient method requires several iterations to satisfy the stopping criterium – this is due to the elongated shape of the level sets of the cost functional, see Fig. 9. The smallest number of iterations required to reach convergence is obtained in the Newton case (see Table 4).

For the sake of comparison, we report in Table 5 the total computational cost entailed by the reduced optimization, including also the offline construction of the RB spaces, see Table 3 for the CPU times concerning this latter operation. We consider $N = 30$ RB functions and (i) the greedy Algorithm 1 for the gradient and the quasi-Newton methods, (ii) the greedy Algorithm 3 including the reduction of the sensitivity equations for the Newton method, respectively.

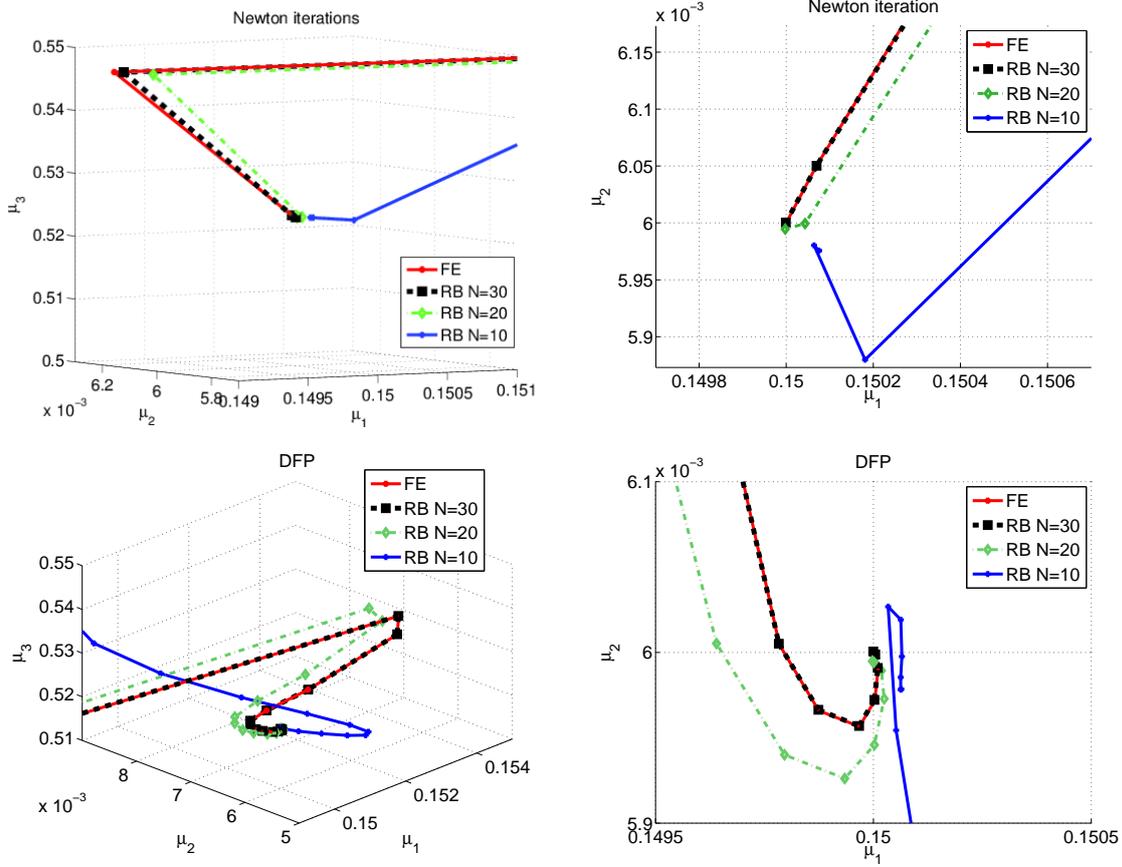


Figure 9: Iterations of Newton (top) and quasi-Newton methods (bottom) in the parameter space.

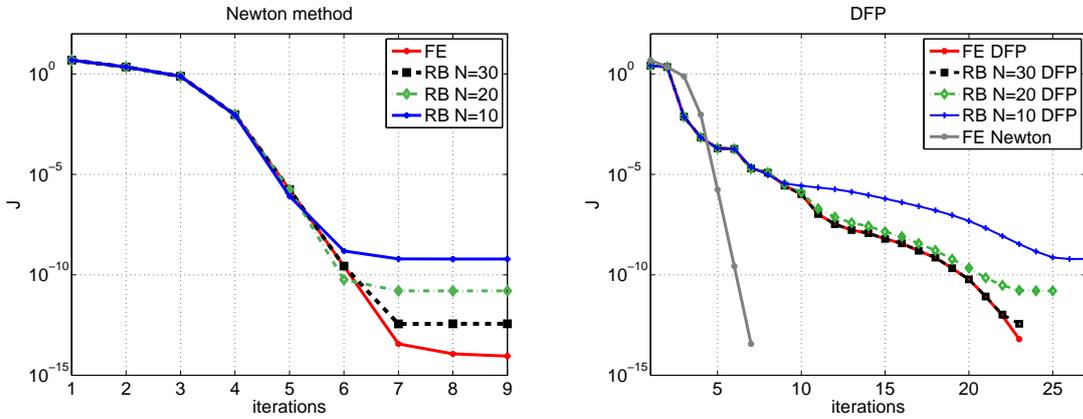


Figure 10: Trend of the cost functional for Newton (left) and quasi-Newton method (right).

	$\#it_{RB}$	$CPUtime/it_{RB}$	$CPUtime/it_{FE}$
Gradient (Armijio)	1699	0.0788 s	1.5001 s
Gradient (l-s)	626	10.58 s	191.5 s
Gradient (l-s) + quasi-Newton	12 + 37	0.0385 s	0.7951 s
Gradient (l-s) + Newton	12 + 8	0.0917 s	1.4944 s

Table 4: Number of iterations and CPU times of the Gradient method with Armijio rule and inexact line-search in $(10^{-2}, 50)$ with 500 samples, of quasi-Newton and Newton methods ($N = 30$ RB functions).

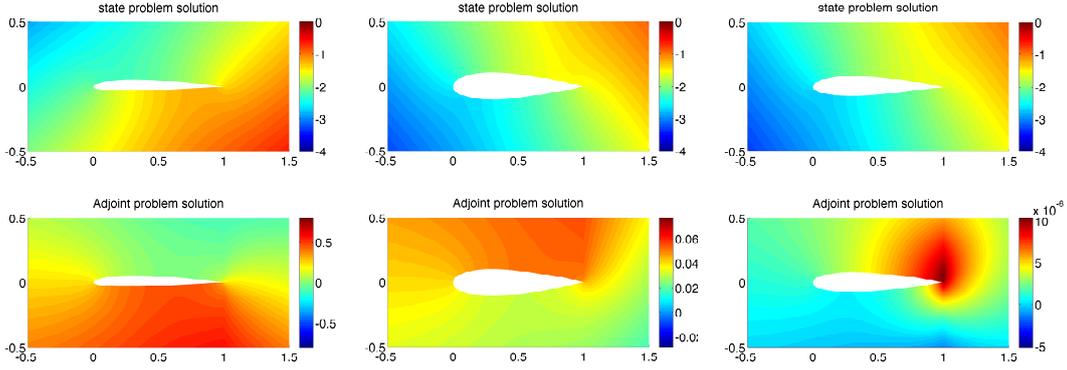


Figure 11: Solutions of the state and adjoint problems for $\boldsymbol{\mu}_n^{(0)} = (0.08, 0.012, -\frac{\pi}{4})$ (left), $\boldsymbol{\mu}_n^{(2)} = (0.2029, 0, 0.5759)$ (center) and $\hat{\boldsymbol{\mu}}_n = (0.1501, 0.006, 0.5236)$ (right).

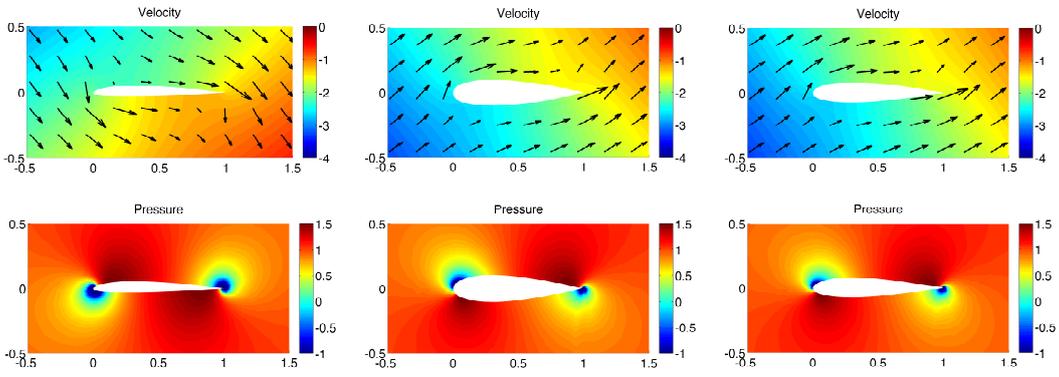


Figure 12: Velocity and pressure fields for $\boldsymbol{\mu}_n^{(0)} = (0.08, 0.012, -\frac{\pi}{4})$ (left), $\boldsymbol{\mu}_n^{(2)} = (0.2029, 0, 0.5759)$ (center) and $\hat{\boldsymbol{\mu}}_n = (0.1501, 0.006, 0.5236)$ (right).

	RB offline	RB online	FE
Gradient (Armijio)	31 <i>m</i>	2.21 <i>m</i>	42.47 <i>m</i>
Gradient (l-s)	31 <i>m</i>	110.38 <i>m</i>	1998 <i>m</i>
Gradient (l-s) + quasi-Newton	31 <i>m</i>	2.14 <i>m</i>	38.69 <i>m</i>
Gradient (l-s) + Newton	72 <i>m</i>	2.13 <i>m</i>	38.39 <i>m</i>

Table 5: Overall computational times for the high-fidelity and the reduced (offline+online) optimization.

Performing the reduced optimization during the online stage provides a speedup of about 20 for any chosen method. We get a computational speedup also including both offline and online costs, except for the Newton method, which requires however the construction of $d + 1 > 2$ RB spaces (state + d sensitivity equations). In the case of a single solution of the optimization problem, we save about 20%, 90% and 15% of the CPU time in the case of the gradient method with the Armijio rule, with line search and of the quasi-Newton method, respectively. The gain is even larger if more than one optimization problem has to be solved online, for instance by varying the target function z_d : in this case the offline stage must not be run again, and the optimization just requires a very fast online stage (order of 2 minutes). Last but not least, a possible restarting of the optimization algorithm, whenever required, can be performed inexpensively.

	η_J	$\eta_{\nabla J}$
$N = 10$	28.4091	1000
$N = 20$	2.2999	454.5455
$N = 30$	1.9201	416.6667

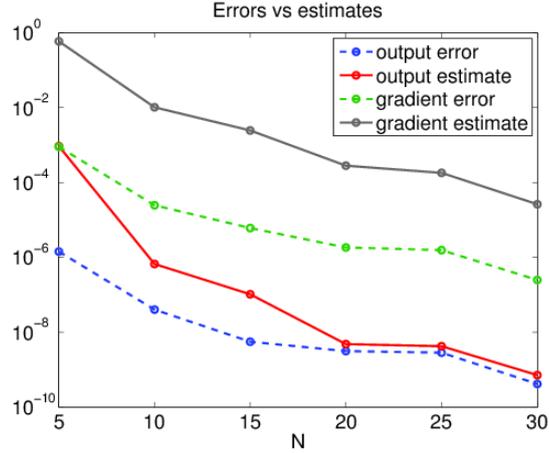


Figure 13: A posteriori error bounds for the cost functional and its gradient: effectivities (left) and behavior with respect to the RB dimension N , average on a sample $\Xi_{test} \subset \mathcal{P}$ of dimension 100 (right).

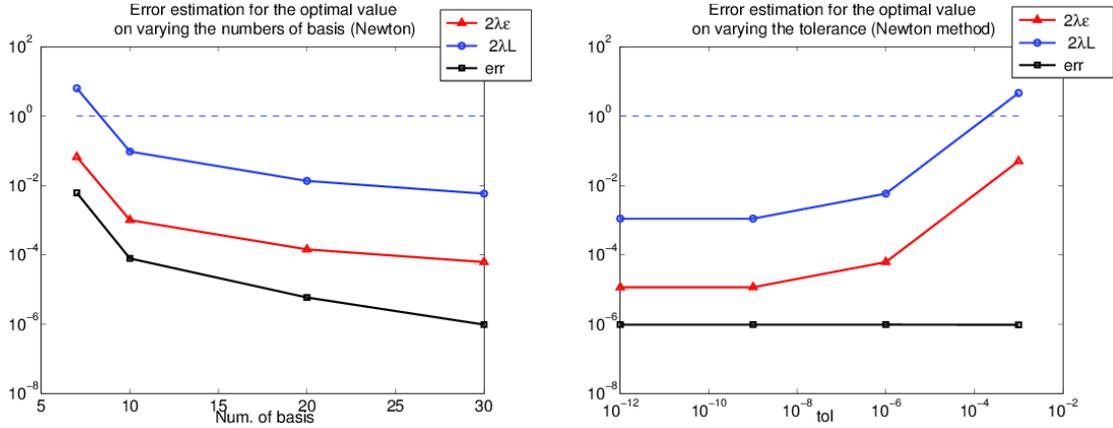


Figure 14: A posteriori error estimates for the optimal parameters (Newton method started with line search gradient iterations) varying the number of basis (left) and the tolerance of the stopping criterium (right).

7 Conclusions

In this work we have proposed a certified reduced basis method for the efficient solution of PDE-constrained parametric optimization problems. We have compared *(i)* a sensitivity-based and *(ii)* an adjoint-based approach, relying in both cases on the reduced basis method to speedup the PDE solution at each step of a descent optimization algorithm. We have also provided a posteriori error bounds for the cost functional, its gradient and the optimal solution in order to control the accuracy of both descent directions evaluated through the RB framework and optimal solutions resulting from the application of the proposed reduced framework. Numerical test cases dealing with linear-quadratic optimal control problems assess the validity of the theoretical results shown in this paper and the efficiency of the proposed reduced-order methodology. Two cases have been considered, dealing with *(i)* the optimal control of thermal flows and *(ii)* the optimal design of airfoil profiles in the case of a potential flow model. In both cases the adjoint-based approach, together with a quasi-Newton method for numerical optimization, provide the best tradeoff in terms of accuracy and computational efficiency.

References

- [1] D. Amsallem, M. J. Zahr, Y. Choi, and C. Farhat. Design optimization using hyper-reduced order models. *Struct. Multidisc. Optim.*, 2014.
- [2] M. Bambach, M. Heinkenschloss, and M. Herty. A method for model identification and parameter estimation. *Inverse Problems*, 29(2), 2013.
- [3] R. F. Barron, X. Wang, T. A. Ameen, and R. O. Warrington. The graetz problem extended to slip-flow. *International Journal of Heat and Mass Transfer*, 40(8):1817 – 1823, 1997.
- [4] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica 2001*, 10:1–102, 2001.
- [5] P. Benner, E. Sachs, and S. Volkwein. Model order reduction for PDE constrained optimization. In G. Leugering, P. Benner, S. Engell, A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, editors, *Trends in PDE Constrained Optimization*, volume 165 of *International Series of Numerical Mathematics*, pages 303–326. Springer International Publishing, 2014.
- [6] A. Borzi and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2011.
- [7] F. Brezzi, J. Rappaz, and P. A. Raviart. Finite dimensional approximation of nonlinear problems. *Num. Math.*, 38(1):1–30, 1982.
- [8] G. Caloz and J. Rappaz. Numerical analysis for nonlinear and bifurcation problems. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis, Vol. V, Techniques of Scientific Computing (Part 2)*, pages 487–637. Elsevier Science B.V., 1997.
- [9] L. Dedè. Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems. *SIAM J. Sci. Comput.*, 32(2):997–1019, 2010.
- [10] L. Dedè. Reduced basis method and error estimation for parametrized optimal control problems with control constraints. *J. Sci. Comput.*, 50(2):287–305, 2012.
- [11] M. Dihlmann and B. Haasdonk. Certified nonlinear parameter optimization with reduced basis surrogate models. In *Proc. Appl. Math. Mech.*, volume 13, pages 3–6. WILEY-VCH Verlag, 2013.
- [12] M. A. Dihlmann and B. Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. *Computational Optimization and Applications*, 60(3):753–787, 2015.
- [13] M. Grepl, N.C. Nguyen, K. Veroy, A.T. Patera, and G.R. Liu. Certified rapid solution of partial differential equations for real-time parameter estimation and optimization. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. Van Bloemen Waanders, editors, *Real-time PDE-Constrained Optimization*, pages 197–215. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007.
- [14] M.A. Grepl and M. Kärcher. Reduced basis a posteriori error bounds for parametrized linear-quadratic elliptic optimal control problems. *C. R. Math. Acad. Sci. Paris*, 349(15-16):873 – 877, 2011.
- [15] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, 2009.
- [16] 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. *C. R. Acad. Sci. Paris. Sér. I Math.*, 345:473–478, 2007.

- [17] M. Kahlbacher and S. Volkwein. Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Disc. Math.: Diff. Incl., Control and Optim.*, 27:95–117, 2007.
- [18] M. Kahlbacher and S. Volkwein. POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems. *ESAIM: Math. Model. Numer. Anal.*, 46(02):491–511, 2012.
- [19] M. Kärcher and M.A. Grepl. A certified reduced basis method for parametrized elliptic optimal control problems. *ESAIM Control Optim. Calc. Var.*, 20(2):416–441, 2014.
- [20] K. Kunisch and S. Volkwein. Proper orthogonal decomposition for optimality systems. *ESAIM Math. Modelling Numer. Anal.*, 42(1):1–23, 2008.
- [21] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. A reduced computational and geometrical framework for inverse problems in haemodynamics. *Int. J. Numer. Methods Biomed. Engng.*, 29(7):741–776, 2013.
- [22] T. Lassila and G. Rozza. Parametric free-form shape design with {PDE} models and reduced basis method. *Comput. Methods Appl. Mech. Engng.*, 199(23–24):1583 – 1592, 2010.
- [23] 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.
- [24] A. Manzoni. *Reduced Models for Optimal Control, Shape Optimization and Inverse Problems in Haemodynamics*. PhD thesis, Lausanne, 2012.
- [25] A. Manzoni and F. Negri. Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs. *Adv. Comput. Math.*, 2015. In press.
- [26] A. Manzoni, S. Pagani, and T. Lassila. Accurate solution of bayesian inverse uncertainty quantification problems using model and error reduction methods. Mathicse Report Nr. 47.2014, 2014.
- [27] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *Int. J. Numer. Methods Fluids*, 70(5):646–670, 2012.
- [28] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids*. Oxford University Press, 2001.
- [29] F. Negri, A. Manzoni, and G. Rozza. Reduced basis approximation of parametrized optimal flow control problems for the stokes equations. *Comput. & Math. with Appl.*, 69(4):319–336, 2015.
- [30] F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni. Reduced basis method for parametrized elliptic optimal control problems. *SIAM J. Sci. Comput.*, 35(5):A2316–A2340, 2013.
- [31] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer New York, 2006.
- [32] I. B. Oliveira and A. T. Patera. Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations. *Optim. & Engng.*, 8(1):43–65, 2007.
- [33] N. A. Pierce and M. B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM review*, 42(2):247–264, 2000.
- [34] A. Quarteroni. *Numerical Models for Differential Problems*, volume 8. Springer, MS & A Series, 2014.
- [35] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *J. Math. Ind.*, 1(1:3), 2011.

- [36] A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis methods for optimal control of advection-diffusion problem. In *Advances in Numerical Mathematics*, W. Fitzgibbon, R. Hoppe, J. Periaux, O. Pironneau, and Y. Vassilevski, Editors, pages 193–216, 2007.
- [37] 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.
- [38] T. Tonn, K. Urban, and S. Volkwein. Comparison of the reduced basis and pod a posteriori error estimators for an elliptic linear-quadratic optimal control problem. *Math. Comput. Model. Dynam. Syst.*, 17(4):355–369, 2011.
- [39] M. J. Zahr and C. Farhat. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *Int. J. Numer. Methods Engrg.*, 102(5):1111–1135, 2015.
- [40] E. Zeidler. *Nonlinear Functional Analysis and its Applications*, volume III: Variational Methods. Springer-Verlag, New York, 1985.
- [41] E. Zeidler. *Nonlinear Functional Analysis and its Applications*, volume I: Fixed-Point Theorems. Springer-Verlag, New York, 1985.
- [42] Y. Zhang, L. Feng, S. Li, and Benner P. Accelerating pde constrained optimization by the reduced basis method. Preprint MPIMD/14-09, Max Planck Institute Magdeburg, 2014.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 31/2015** Pini, A.; Vantini, S.; Colombo, D.; Colosimo, B. M.; Previtali, B.
Domain-selective functional ANOVA for process analysis via signal data: the remote monitoring in laser welding
- 32/2015** Agasisti, T.; Ieva, F.; Masci, C.; Paganoni, A.M.
Does class matter more than school? Evidence from a multilevel statistical analysis on Italian junior secondary school students
- 33/2015** Fumagalli, A.; Pasquale, L.; Zonca, S.; Micheletti, S.
An upscaling procedure for fractured reservoirs with non-matching grids
- 34/2015** Bernardi, M.S.; Mazza, G.; Ramsay, J.O.; Sangalli, L.M.
A separable model for spatial functional data with application to the analysis of the production of waste in Venice province
- 30/2015** Pini, A.; Vantini, S.
Interval-wise testing for functional data
- 29/2015** Antonietti, P.F.; Cangiani, A.; Collis, J.; Dong, Z.; Georgoulis, E.H.; Giani, S.; Houston, P.
Review of Discontinuous Galerkin Finite Element Methods for Partial Differential Equations on Complicated Domains
- 28/2015** Taffetani, M.; Ciarletta, P.
Beading instability in soft cylindrical gels with capillary energy: weakly non-linear analysis and numerical simulations
- 25/2015** Del Pra, M.; Fumagalli, A.; Scotti, A.
Well posedness of fully coupled fracture/bulk Darcy flow with XFEM
- 27/2015** Marron, J.S.; Ramsay, J.O.; Sangalli, L.M.; Srivastava, A.
Functional Data Analysis of Amplitude and Phase Variation
- 26/2015** Tagliabue, A.; Dede', L.; Quarteroni, A.
Nitsche's Method for Parabolic Partial Differential Equations with Mixed Time Varying Boundary Conditions