GOAL-ORIENTED ERROR ESTIMATION FOR PARAMETER-DEPENDENT NONLINEAR PROBLEMS

ALEXANDRE JANON[*], MAËLLE NODET[†], CHRISTOPHE PRIEUR[‡], AND CLÉMENTINE PRIEUR[§]

Abstract. The main result of this paper gives a numerically efficient method to bound the error that is made when approximating the output of a nonlinear problem depending on an unknown parameter (described by a probability distribution). The class of nonlinear problems under consideration includes high-dimensional nonlinear problems with a nonlinear output function. A goal-oriented probabilistic bound is computed by considering two phases. An offline phase dedicated to the computation of a reduced model during which the full nonlinear problem needs to be solved only a small number of times. The second phase is an online phase which approximates the output. This approach is applied to a toy model and to a nonlinear partial differential equation, more precisely the Burgers equation with unknown initial condition given by two probabilistic parameters. The savings in computational cost are evaluated and presented.

Key words. goal-oriented; probabilistic error estimation; nonlinear problems; uncertainty quantification

AMS subject classifications. 49Q12; 62F12; 65C20; 82C80

1. Introduction.

Numerical simulation is a key component of numerous domains: industry, environment, engineering, physics for instance. In some cases time is the limiting factor, and the numerical simulation should be very fast and accurate. For example, the control of the trajectory of a space satellite may require efficient real-time computations. Another example would be the iterative optimization algorithm used in numerical weather prediction, which requires numerous calls to a numerical atmosphere model, to be performed in a limited time. In both examples, the computing time is a key factor: it must be very short, either because the computation is repeated many times in a relatively short interval (many-query context) or because the result cannot wait (real-time context).

In this context, it is crucial to provide fast numerical evaluations for linear or nonlinear problems. These procedures are generally called “metamodelling”, “model reduction”, “dimension reduction”. It consists in replacing the existing model, called the “full” model, by a fast approximation. There exist both stochastic and deterministic approaches to building such approximations. On the stochastic part we can mention polynomial chaos approximation [24, 6, 14], Gaussian processes (including Kriging and RKHS-based methods –reproducing kernel Hilbert spaces) [13, 20], low-rank tensor methods [19], etc. which all provide cheap and fast approximations of the full model. On the deterministic side we can cite the reduced basis method [15], POD (proper orthogonal decomposition) [26], balanced truncation [16], etc. All these methods have in common that they provide a way to build a numerical model which is faster than the full model.

Of course, decreasing the computational time required for solving the model is not the only aim these methods have. It is crucial that they also provide accurate ap-

[*] Laboratoire de Mathématiques d’Orsay, Université Paris-Sud, France alexandre.janon@math.u-psud.fr

[†] Univ. Grenoble Alpes, INRIA, France maelle.nodet@univ-grenoble-alpes.fr

[‡] Univ. Grenoble Alpes, CNRS, Gipsa-lab, F-38000 Grenoble, France Christophe.Prieur@gipsa-lab.fr

[§] Univ. Grenoble Alpes, INRIA, France clementine.prieur@univ-grenoble-alpes.fr

proximations of the full model. The approximation error, i.e. the comparison between the full model and the metamodel, should ideally be certified and known by the user of the metamodel. In practice, some metamodelling methods only provide limited validation and certification so that the user has to take a leap of faith because there is no quantified guarantee about the metamodel accuracy. However, it is possible in some cases to design metamodels which include a certified error bound. In this latter case, the user does not know exactly the approximation error, but the error is guaranteed to be lower than the provided bound. Moreover, the error bound computation is included in the metamodel, so that its computational burden stays small compared to the full model. For example, we can cite [18] where the authors provide such bounds in the framework of the reduced basis method (dimension reduction).

Furthermore, in many application cases, one is not interested in the solution by itself, but rather in a quantity of interest, or model output, which is a functional of this solution. Taking this functional into account when performing the model reduction leads to a so-called goal-oriented method. For instance, goal-oriented basis choice procedures have been successfully introduced in the context of dynamical systems in [27, 7], where the basis is chosen so as to contain the modes that are relevant to accurately represent the output of interest, and in a general context in [3], where the basis is chosen so as to minimize the overall output error. All those papers showed that using an adapted basis could lead to a great improvement of reduction error. In [18], the authors consider, in the context of reduced basis, goal-oriented error estimation, that is, the description of a rigorous and computable error bound between the model output and the reduced one. In [10], the authors outperform the accuracy of the bound in [18] by accepting a small risk $\alpha \in (0,1)$ of this bound to be violated. They provide a so-called probabilistic error bound. Providing such error bound for a large class of nonlinear problems and for any approximation procedure is the aim of this paper.

More precisely, we are considering, for a given parameter μ in a parameter space $\mathcal{P}$, the solution $u(\mu) \in X$ of an equation of the form $\mathcal{M}(\mu, u(\mu)) = 0$, with $\mathcal{M} : \mathcal{P} \times X \to Y$, and X, Y two finite dimensional vector spaces that are introduced in Section 2 below.

We extend in the present paper the results in [10] by providing a probabilistic goal-oriented error estimation procedure for a large class of nonlinear problems $\mathcal{M}(\mu, u(\mu)) = 0$, and for very general metamodelling methods. More specifically, let $s(\mu) = \langle \ell, u(\mu) \rangle_X$, for a given $\ell \in X$, and $\widetilde{s}(\mu) = \langle \ell, \widetilde{u}(\mu) \rangle_X$, with $\widetilde{u}$ a quite general metamodel for u. The problem under consideration in this paper is to derive a numerically efficient method to compute a probabilistic bound for the error $|s(\mu) - \widetilde{s}(\mu)|$. Our approach is based on a generalized notion of the finite difference adjoint of an operator in the nonlinear and probabilistic context. Two different phases are applied when estimating the error on the output of the nonlinear problem. First an offline phase, where the high-dimensional nonlinear problem is solved a small number of times, and an online phase where, using the new notion of finite difference adjoint, a low dimensional problem is solved. This paper also provides an illustration by means of numerical experiments.

The notion of adjoint we introduce here is related to the adjoint used in a posteriori error estimation works such as in [2]. We propose a simple and robust definition, nicely tailored to the online error bound setting. The idea of probabilistic error estimation also appeared in the ROMES method [5]. This last approach is of different nature, as it is based on a step of statistical modeling of the errors introduced by the reduced-order model.

The paper is organized as follows: in Section 2, we precise the objectives of our study, that is the derivation of an offline/online probabilistic goal-oriented error estimation procedure in a nonlinear context. In Section 3, we describe the different steps of the procedure. More precisely, we introduce in Section 3.1, the notion of finite difference adjoint of an operator, before extending in Section 3.2 the procedure in [10] to nonlinear models and linear outputs. In Section 3.3, we prove that the results in Section 3.2 can easily be extended to nonlinear models and nonlinear outputs. Section 3.5 provides the different steps for a practical efficient evaluation of the error bound. Some numerical experiments are presented in Section 4, first with a linear transport equation, then with the nonlinear Burgers partial differential equation. Section 5 contains some concluding remarks and Appendix A collects the proof of some intermediate results.

2. Problem statement.

Let $\mathcal{P} \subset \mathbb{R}^d$ denote a parameter space, and let P be a probability distribution on $\mathcal{P}$. Let X (*resp.* Y) be vector space of dimension $\mathcal{N}$ (resp. $\mathcal{S}$) endowed with a scalar product $\langle \cdot, \cdot \rangle_X$ (*resp.* $\langle \cdot, \cdot \rangle_Y$). In the following, when there is no ambiguity, the dependence in the vector space for the scalar product will be omitted in the notation $\langle \cdot, \cdot \rangle$. Let us consider a nonlinear function $\mathcal{M} : \mathcal{P} \times X \to Y$. Given a parameter $\mu \in \mathcal{P}$, we denote by $u(\mu) \in X$ a solution to the equation:

$$\mathcal{M}(\mu, u(\mu)) = 0, \tag{1}$$

and we define the *output* by

$$s(\mu) = \langle \ell, u(\mu) \rangle_X, \tag{2}$$

for a given $\ell \in X$.

We assume that for every $\mu \in \mathcal{P}$, Equation (1) admits a unique solution in X, so that the application $s : \mathcal{P} \to \mathbb{R}$ is well-defined.

In a *many-query context*, that is in a context requiring a potentially large number of evaluations of the output, it is common to call for model reduction. More precisely, let $\widetilde{X}$ be a subspace of X, of dimension N such that $N << \mathcal{N}$. We consider $\widetilde{u} : \mathcal{P} \to \widetilde{X}$ an approximation (in a very wide sense of the term) of $u : \mathcal{P} \to X$. Let us define the approximate output $\widetilde{s}(\mu)$ by

$$\widetilde{s}(\mu) = \langle \ell, \widetilde{u}(\mu) \rangle_X.$$

The objective is then to provide a probabilistic error bound between $s(\mu)$ and $\widetilde{s}(\mu)$. In other words, one accepts the risk of this bound $\epsilon(\mu; \alpha)$ being violated for a set of parameters having "small" probability measure $\alpha \in (0, 1)$:

$$P\big(|s(\mu) - \widetilde{s}(\mu)| \geq \epsilon(\mu; \alpha) \big) \leq \alpha.$$

This quantity $\epsilon(\mu; \alpha)$ is a so-called “goal-oriented probabilistic error bound”.

For sake of efficiency, the computation of the approximate output can be split into two phases:

- an *offline* phase, dedicated to the construction of the reduced model $\widetilde{u}$, during which one has to solve the full dimensional problem (1) only for a reasonably small number of parameters $\mu_1, \dots, \mu_\kappa$;
- an *online* phase, during which we evaluate the approximate output $\widetilde{s}(\cdot) = \langle \ell, \widetilde{u}(\cdot) \rangle_X$ for all queried μ.

In practice, for any $\mu \in \mathcal{P}$, the computational time of $\widetilde{u}(\mu)$ is much smaller than the one of $u(\mu)$, hence this splitting into offline and online phases can be interesting in terms of overall computing time: the offline phase can be computationally expensive, provided that the number of queries is large enough and/or the online phase per query is fast enough.

In this article, we will not focus on the ways of constructing efficient offline-online approximation procedures for $u(\mu)$, as in *e.g.*, [18], [8], [22], [12]. Assumptions on the approximation procedure in use are very mild (see Section 3.5 and more specifically Lemma 11). Under these mild assumptions, we propose hereafter a new procedure to compute efficiently, using an online / offline decomposition, a goal-oriented probabilistic error bound $\epsilon(\mu;\alpha)$ which generalizes the error bound described in [10] (see also [11] for further results in control theory).

3. Probabilistic nonlinear error bound. In this section, we aim at providing a goal-oriented probabilistic error bound on the output. In [10], the authors propose such an error bound in the linear context, that is assuming that for any $\mu \in \mathcal{P}$, the operator $\mathcal{M}(\mu,\cdot) : X \to Y$ is affine (linear operator + a constant), and that the output is also linear. In the sequel we will call *linear* this case, as opposed to the nonlinear case where the model is not affine.

By accepting a small risk $\alpha \in (0,1)$ that this bound could be violated, the authors avoid the use of (often pessimistic) Lipschitz bounds. In this section, we extend the results in [10] to the cases where the operator $\mathcal{M}(\mu,\cdot) : X \to Y$ is nonlinear. In Section 3.2, the output is assumed to be linear, then in Section 3.3, the output may be nonlinear.

To derive an error bound, it seems natural to consider the so-called residual

$$r(\mu) = \mathcal{M}(\mu,\widetilde{u}(\mu)) - \mathcal{M}(\mu,u(\mu)),\ \mu \in \mathcal{P}. \tag{3}$$

In the sequel we explain why we need to define a new adjoint. To do so we recall the computations of the linear case, in order to draw the parallel with the nonlinear case and motivate the need for a new adjoint definition.
In the linear case, $\mathcal{M}(\mu,u) = A(\mu)u - b(\mu)$ where $b(\mu) \in Y$ is a given vector and $A(\mu) : X \to Y$ is a linear operator which is assumed to be invertible. In the following, $A(\mu)^\top$ denotes the transpose of $A(\mu)$. We can define $w(\mu) \in Y$ as the solution of the so-called dual problem:

$$\mathcal{M}^\star(\mu, w(\mu)) = A^\top(\mu)w(\mu) = \ell \tag{4}$$

where $\ell \in X$ is used in the definition of the linear output in (2), and with $\mathcal{M}^\star(\mu,\cdot)$ the linear adjoint of $\mathcal{M}(\mu,\cdot)$. Let $\mathbf{\Phi} = \{\phi_1,\dots,\phi_{\mathcal{S}}\}$ denote any orthonormal basis of Y. We then have

$$\begin{aligned}
\widetilde{s}(\mu) - s(\mu) &= \langle \ell, \widetilde{u}(\mu) - u(\mu)\rangle_X = \langle A^\top(\mu)w(\mu), \widetilde{u}(\mu) - u(\mu)\rangle_X \\
&= \langle w(\mu), A(\mu)\widetilde{u}(\mu) - A(\mu)u(\mu)\rangle_Y = \langle w(\mu), r(\mu)\rangle_Y \\
&= \sum_{i=1}^{\mathcal{S}} \langle w(\mu), \phi_i\rangle_Y \langle r(\mu), \phi_i\rangle_Y. \qquad (5)
\end{aligned}$$

In order to adapt this procedure to the nonlinear context, we need to define a generalization of the adjoint $\mathcal{M}^\star$ for nonlinear operators, that still allows (5). In Section 3.1 below, we will define $\mathcal{M}^\star : \mathcal{P} \times X \times X \times Y \to X$, and the associated adjoint problem for $w(\mu)$:

$$\mathcal{M}^\star(\mu, \widetilde{u}(\mu), u(\mu), w(\mu)) = \ell \tag{6}$$

which generalizes (4).

3.1. Finite difference adjoint of an operator.

The definition of the adjoint problem for nonlinear operators is not unique. In order to generalize (5) for nonlinear problems, we require a specific property, which leads us to choose the following definition:

Definition 1. *[Finite difference adjoint] We call* finite difference adjoint *any operator*

$$\mathcal{M}^\star : \mathcal{P} \times X \times X \times Y \to X,$$

linear in the last variable, such that the following identity holds:

$$\begin{gathered}\forall \mu \in \mathcal{P}, \forall x_1, x_2 \in X, \forall y \in Y\,, \\ \langle x_1 - x_2, \mathcal{M}^\star(\mu, x_1, x_2, y)\rangle_X = \langle \mathcal{M}(\mu, x_1) - \mathcal{M}(\mu, x_2), y\rangle_Y.\end{gathered} \tag{7}$$

Let us underline that previous definitions of nonlinear adjoint do not readily allow for this property, such as, e.g., the one offered by Definition 2.1 in [21]:

$$\forall \mu \in \mathcal{P}, \forall x \in X, \forall y \in Y,\ \langle x, \mathcal{M}^\star(\mu, x, y)\rangle_X = \langle \mathcal{M}(\mu, x), y\rangle_Y.$$

In our case the dependence in both x_1, x_2 is crucial, and missing in previous definitions. Indeed, as we will see later on in Equation (12) we do need a dependence in u and $\widetilde{u}$, and not only in $u - \widetilde{u}$ as in the linear case. A similar trick can be found in [2] (page 12), formalized below in Proposition 2, in which we propose a formula for finite difference adjoint candidate.

Proposition 2. *Assume that the operator $\mathcal{M} : \mathcal{P} \times X \to Y$ is continuously Fréchet-differentiable with respect to the second variable. Let $d\mathcal{M}(\mu, x) : X \to Y$ denote the Fréchet-derivative of $\mathcal{M}$ with respect to $x \in X$ at (μ, x). Let $d\mathcal{M}^\star(\mu, x) : Y \to X$ denote the (linear) adjoint of $d\mathcal{M}(\mu, x)$. We now denote*

$$\mathcal{M}^\star(\mu, x_1, x_2, y) = \int_0^1 d\mathcal{M}^\star(\mu, x_2 + s(x_1 - x_2))(y)ds \tag{8}$$

for all $(\mu, x_1, x_2, y) \in \mathcal{P} \times X \times X \times Y$. Then $\mathcal{M}^\star$ is a finite difference adjoint.

Proof of Proposition 2. The proof is postponed to Appendix A.1. □

From now on, we will choose formula (8) for the adjoint of $\mathcal{M}$. The following properties are immediate:

1. Assume that $\mathcal{M}(\mu, \cdot)$ is affine, with $\mathcal{M}(\mu, x) = A(\mu)x - b(\mu)$ where $A(\mu) : X \to Y$ is a linear operator and $b(\mu) \in Y$. Then

$$\forall \mu \in \mathcal{P}, \forall x_1, x_2 \in X, \forall y \in Y,\ \ \mathcal{M}^\star(\mu, x_1, x_2, y) = A(\mu)^\top y.$$

2. For all $\mu \in \mathcal{P}$, and for all $x_1, x_2 \in X$, $\mathcal{M}^\star(\mu, x_1, x_2, \cdot)$ is linear.

Let us now consider the adjoint problem described by (6):

$$\text{Find } w(\mu) \text{ solution of } \mathcal{M}^\star(\mu, \widetilde{u}(\mu), u(\mu), w(\mu)) = \ell. \tag{9}$$

This problem is linear. Let us assume that, for all $\mu \in \mathcal{P}$, it admits a solution. We have then the following lemma:

Lemma 3. *Assume that the operator $\mathcal{M} : \mathcal{P} \times X \to Y$ is continuously Fréchet-differentiable with respect to the second variable, and let $\mathcal{M}^*$ be defined by (8). Let*

$s(\mu) = \langle \ell, u(\mu)\rangle$ $\widetilde{s}(\mu) = \langle \ell, \widetilde{u}(\mu)\rangle$, $r(\mu)$ be defined in (3), let $w(\mu)$ be a solution of (9) and let $\{\phi_1, \dots, \phi_{\mathcal{S}}\}$ denote any orthonormal basis of Y. Then it holds

$$\widetilde{s}(\mu) - s(\mu) = \sum_{i=1}^{\mathcal{S}} \langle w(\mu), \phi_i\rangle_Y \langle r(\mu), \phi_i\rangle_Y. \tag{10}$$

Proof of Lemma 3. Proposition 2 applies and Item *2* after Proposition 2 claims that $\mathcal{M}^\star$ is linear in its fourth argument, thus the adjoint problem described in (6) is linear. We assume that for all $\mu \in \mathcal{P}$ it admits a solution $w(\mu)$.

Following the beginning of the proof of Theorem 1.1 in [10], we expand the residual in the basis $\mathbf{\Phi}$:

$$r(\mu) = \sum_{i=1}^{\mathcal{S}} \langle r(\mu), \phi_i\rangle_Y \phi_i. \tag{11}$$

Then:

$$\widetilde{s}(\mu) - s(\mu) = \langle l, \widetilde{u}(\mu) - u(\mu)\rangle_X.$$

As $w(\mu)$ is solution of (6), we get:

$$\widetilde{s}(\mu) - s(\mu) = \langle \mathcal{M}^\star(\mu, \widetilde{u}(\mu), u(\mu), w(\mu)), \widetilde{u}(\mu) - u(\mu)\rangle_X.$$

Then, applying Identity (7) we obtain:

$$\widetilde{s}(\mu) - s(\mu) = \langle \mathcal{M}(\mu, \widetilde{u}(\mu))) - \mathcal{M}(\mu, u(\mu)), w(\mu)\rangle_Y = \langle r(\mu), w(\mu)\rangle_Y. \tag{12}$$

Finally, recalling (11), and as the basis $\mathbf{\Phi}$ is orthonormal, we get (10). □

3.2. Probabilistic error bound for a nonlinear model with linear output.

This section is devoted to the statement of our probabilistic error bound, in the context where the model is nonlinear and where the output is linear.

We now introduce some notation necessary to the statement of our bound. Recall that $\mathbf{\Phi} = \{\phi_1, \dots, \phi_{\mathcal{S}}\}$ denotes any orthonormal basis of Y. Let $K \leq \mathcal{S}$ be a "truncation index". For any $i \in \{1, \dots, K\}$, we define:

$$D_i(\mu, \mathbf{\Phi}) = \langle w(\mu), \phi_i\rangle_Y, \quad \beta_i^{min}(\mathbf{\Phi}) = \min_{\mu\in\mathcal{P}} D_i(\mu, \mathbf{\Phi}), \quad \beta_i^{max}(\mathbf{\Phi}) = \max_{\mu\in\mathcal{P}} D_i(\mu, \mathbf{\Phi}).$$

The probabilistic error bound depends on the residual defined by (3):

$$r(\mu) = \mathcal{M}(\mu, \widetilde{u}(\mu)) - \mathcal{M}(\mu, u(\mu)) = \mathcal{M}(\mu, \widetilde{u}(\mu)).$$

Our aim is to propose a probabilistic upper bound for $|s(\widetilde{u}(\mu)) - s(u(\mu))|$. For this, let us consider the right-hand term in (10): $\sum_{i=1}^{\mathcal{S}} \langle r(\mu), \phi_i\rangle_Y \langle w(\mu), \phi_i\rangle_Y$. In order to bound this term, it seems natural to define, for any $\mu \in \mathcal{P}$, and for any $1 \leq i \leq \mathcal{S}$:

$$\beta_i^{up}(\mu, \mathbf{\Phi}) = \begin{cases} \beta_i^{max}(\mathbf{\Phi}) \text{ if } \langle r(\mu), \phi_i\rangle_Y > 0 \\ \beta_i^{min}(\mathbf{\Phi}) \text{ else} \end{cases}$$

$$\beta_i^{low}(\mu, \mathbf{\Phi}) = \begin{cases} \beta_i^{min}(\mathbf{\Phi}) \text{ if } \langle r(\mu), \phi_i\rangle_Y > 0 \\ \beta_i^{max}(\mathbf{\Phi}) \text{ else} \end{cases}$$

We then introduce a truncation argument, K. Our aim is then to bound the truncated sum $\left|\sum_{i=1}^{K}\langle r(\mu),\phi_i\rangle_Y\langle w(\mu),\phi_i\rangle_Y\right|$. We thus define:

$$T_1^{up}(\mu,K,\mathbf{\Phi})=\sum_{i=1}^{K}\langle r(\mu),\phi_i\rangle_Y\beta_i^{up}(\mu,\mathbf{\Phi}),\;\; T_1^{low}(\mu,K,\mathbf{\Phi})=\sum_{i=1}^{K}\langle r(\mu),\phi_i\rangle_Y\beta_i^{low}(\mu,\mathbf{\Phi}),$$

and

$$T_1(\mu,K,\mathbf{\Phi})=\max\left(\left|T_1^{up}(\mu,K,\mathbf{\Phi})\right|,\left|T_1^{low}(\mu,K,\mathbf{\Phi})\right|\right).$$

We refer to Section 3.5 for the computation of $T_1(\mu,K,\phi)$. Let us just underline that it will require, during the *online phase*, the computation of the K scalar products $\langle r(\mu),\phi_i\rangle_Y$, $i=1,\dots,K$. Under additional assumptions (see Lemma 11), it is possible to compute each of these scalar products with a cost independent from the full dimension $\mathcal{S}$. However, we still need to choose a truncation argument K independent from $\mathcal{S}$ to keep the cost of the *online* phase independent from $\mathcal{S}$. Note also that during the *offline* phase, $2K$ optimization problems have to be solved for the computation of β_i^{min}, β_i^{max}, $i=1,\dots,K$. Thus if we increase K, we also increase the cost of the *offline* phase. To deal with the remainder of the sum, we define:

$$T_2(K,\mathbf{\Phi})=\mathbf{E}_\mu\left(\left|\sum_{i=K+1}^{\mathcal{S}}\langle r(\mu),\phi_i\rangle_Y\langle w(\mu),\phi_i\rangle_Y\right|\right),$$

with $\mathbf{E}_\mu$ denoting the expectation with respect to the probability distribution of μ.

Our main result is then:

THEOREM 4. *Let $\alpha\in(0,1)$. We have*

$$P\left(|s(\mu)-\widetilde{s}(\mu)|\geq\epsilon(\mu;\alpha)\right)\leq\alpha$$

where the error bound $\epsilon(\mu;\alpha)$ is defined by

$$\epsilon(\mu;\alpha)=T_1(\mu,K,\mathbf{\Phi})+\frac{T_2(K,\mathbf{\Phi})}{\alpha}.$$

REMARK 5. *The result of Theorem 4 is a generalization of Theorem 1.1 in [10] to nonlinear operators $\mathcal{M}$.*

The result of Theorem 4 is true for any orthonormal basis $\mathbf{\Phi}$ of Y. For efficiency reasons, we would like to choose $\mathbf{\Phi}$ so that the parameter-independent part $T_2(K,\mathbf{\Phi})$ is the smallest possible, for a fixed truncation index $K\in\mathbb{N}^*$.

To our knowledge, minimizing $T_2(K,\mathbf{\Phi})$ over orthonormal bases of Y is an optimization problem for which no efficient algorithm exists. However, we can minimize an upper bound of $T_2(K,\mathbf{\Phi})$.

We define a self-adjoint, positive semi-definite operator $G:Y\to Y$ by:

$$\forall\varphi\in Y,\;\; G\varphi=\frac{1}{2}\mathbf{E}_\mu\left(\langle r(\mu),\varphi\rangle_Y r(\mu)+\langle w(\mu),\varphi\rangle_Y w(\mu)\right). \tag{13}$$

Let $\lambda_1\geq\lambda_2\geq\dots\lambda_{\mathcal{S}}\geq 0$ be the eigenvalues of G. Let, for $i\in\{1,2\dots,\mathcal{S}\}$, ϕ_i^G be an unit eigenvector of G associated with the i^{th} eigenvalue, and

$$\mathbf{\Phi}^G=\{\phi_1^G,\dots,\phi_{\mathcal{S}}^G\}.$$

We recall the following result that will be useful to prove our main result.

Lemma 6 (Theorem 1.2. in [10]). *It holds*

$$T_2(K, \mathbf{\Phi}^G) \leq \sum_{K+1}^{\mathcal{S}} \lambda_i^2.$$

Lemma 6 suggests a heuristic choice of $\mathbf{\Phi} = \mathbf{\Phi}^G$. Indeed, we see that the study of the eigenvalues of G allows to bound $T_2(K, \mathbf{\Phi}^G)$, e.g., if there exist $0 < \rho < 1$ and $C > 0$ such that $\lambda_i \leq C\rho^{\frac{i}{2}}$, then $T_2(K, \mathbf{\Phi}^G) \leq C\rho^{K+1}\dfrac{1-\rho^{\mathcal{S}-K}}{1-\rho}$. In many applications, the eigenvalues of G are observed to fast decay to zero.

We are now in position to prove our main result.

Proof of Theorem 4. We start from the result of Lemma 3:

$$\widetilde{s}(\mu) - s(\mu) = \sum_{i=1}^{\mathcal{S}} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y.$$

Then, we can argue as in the proof of Theorem 1.1 in [10]. By construction of $T_1(\mu, K, \mathbf{\Phi})$ one gets:

$$\left| \sum_{i=1}^{K} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y \right| \leq T_1(\mu, K, \mathbf{\Phi}).$$

Thus, for any $\alpha \in (0,1)$,

$$P\left(|\widetilde{s}(\mu) - s(\mu)| \geq T_1(\mu, K, \mathbf{\Phi}) + \frac{T_2(K, \mathbf{\Phi})}{\alpha} \right)$$

$$\leq P\left(|\widetilde{s}(\mu) - s(\mu)| \geq \left| \sum_{i=1}^{K} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y \right| + \frac{T_2(K, \mathbf{\Phi})}{\alpha} \right)$$

$$\leq P\left(\left| \sum_{i=K+1}^{\mathcal{S}} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y \right| \geq \frac{T_2(K, \mathbf{\Phi})}{\alpha} \right),$$

where in the last inequality, Lemma 3 has been used. Then, by Markov Inequality, using $\alpha \in (0,1)$, and by definition of $T_2(\mu, K, \mathbf{\Phi})$ we get:

$$P\left(\left| \sum_{i=K+1}^{\mathcal{S}} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y \right| \geq \frac{T_2(\mathbf{\Phi})}{\alpha} \right)$$

$$\leq \frac{\mathbf{E}_\mu \left(\left| \sum_{i=K+1}^{\mathcal{S}} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y \right| \right)}{\frac{T_2(K, \mathbf{\Phi})}{\alpha}} = \alpha.$$

This concludes the proof of Theorem 4. □

3.3. Corollary: error bound for a nonlinear output.

In this section we provide an extension of Theorem 4 to the context of a nonlinear output $S(\mu)$. To do so we consider the following problem:

Problem 7.

$$\textit{Find } v(\mu) \textit{ such that } \mathcal{H}(\mu, v(\mu)) = 0$$

where $\mathcal{H} : \mathcal{P} \times X \to Y$ is a (not necessarily linear with respect to the second argument) function, and consider the following output:

$$S(\mu) = f(v(\mu))$$

where f is a (not necessarily linear) function from Y to $\mathbb{R}$.

In the context of this section, our main result is based on

Lemma 8. *Problem 7 can be written in the framework of a non necessarily linear model $\mathcal{M} : \mathcal{P} \times (X \times \mathbb{R}) \to Y$ and of a linear output $s(\mu) = \langle \ell, u(\mu) \rangle_X$ with $\ell \in X \times \mathbb{R}$.*

Proof of Lemma 8. It relies on a classical idea in the field of Data Assimilation (see, e.g., [30]), which consists in augmenting the state vector $v(\mu)$ with the output $S(\mu)$:

$$u(\mu) = \begin{pmatrix} v(\mu) \\ S(\mu) \end{pmatrix} = \begin{pmatrix} \overline{u}(\mu) \\ \underline{u}(\mu) \end{pmatrix} \in X \times \mathbb{R}$$

where $\overline{u}(\mu) \in X$ denotes the first component of $u(\mu)$ (corresponding to $v(\mu)$) and $\underline{u}(\mu) \in \mathbb{R}$ its last component (corresponding to $S(\mu)$). We then define $\mathcal{M} : \mathcal{P} \times (X \times \mathbb{R}) \to Y$ by:

$$\mathcal{M}(\mu, u(\mu)) = \begin{pmatrix} \mathcal{H}(\mu, \overline{u}(\mu)) \\ f(\overline{u}(\mu)) - \underline{u}(\mu) \end{pmatrix},$$

and consider the following linear output:

$$s(\mu) = S(\mu) = \underline{u}(\mu) = \langle \ell, u(\mu) \rangle \text{ with } \ell = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in X \times \mathbb{R}.$$

Problem 7 is then equivalent to:

$$\text{find } u(\mu) \text{ such that } \mathcal{M}(\mu, u(\mu)) = 0 \text{ with the output } s(\mu) = \ell$$

This concludes the proof of Lemma 8. □

By combining Lemma 8 with Theorem 4, we get an error bound in the context of a nonlinear output $S(\mu)$. This gives a solution to Problem 7.

3.4. Computation of the finite difference adjoint of $\mathcal{M}$.

Except in some particular cases there exists no explicit formulation of the adjoint of $\mathcal{M}$ in the context of Proposition 2. To illustrate this purpose, let us consider the case where $\mathcal{H}$ is affine (with respect to the second argument). For the sake of simplicity, let us fix in this section $X = \mathbb{R}^{\mathcal{N}}$. Let $B(\mu)$ denote the matrix representation of the linear part of $\mathcal{H}(\mu, \cdot)$ with respect to the canonical basis of X. Even in that case, as the output is nonlinear, the operator $\mathcal{M}$ is also nonlinear. Recall that we assume that $\mathcal{M}$ is continuously-Fréchet differentiable with respect to its second variable. We want to provide an explicit formulation for the adjoint of the operator $\mathcal{M}$, starting from (8). We first consider $d\mathcal{M}(\mu, \cdot)$. For $v \in \mathbb{R}^{\mathcal{N}+1}$, recall that:

$$d\mathcal{M}(\mu,u)(v) = \lim_{\alpha\to 0} \frac{\mathcal{M}(\mu,u+\alpha v) - \mathcal{M}(\mu,u)}{\alpha}$$

which leads immediately to:

$$d\mathcal{M}(\mu,u)(v) = \begin{pmatrix} B(\mu)\overline{v} \\ df(\overline{u})(\overline{v}) - \underline{v} \end{pmatrix} = \begin{pmatrix} B(\mu) & (0) \\ df(\overline{u}) & -1 \end{pmatrix} \begin{pmatrix} \overline{v} \\ \underline{v} \end{pmatrix}$$

so that $d\mathcal{M}(\mu,u)$ is the following matrix, defined by blocks:

$$d\mathcal{M}(\mu,u) = \begin{pmatrix} B(\mu) & (0) \\ df(\overline{u}) & -1 \end{pmatrix}$$

where the top left block has size $\mathcal{S}\times\mathcal{N}$, the top right block $\mathcal{S}\times 1$, the bottom left $1\times\mathcal{N}$ (as $f:\mathbb{R}^{\mathcal{N}}\to\mathbb{R}$) and the bottom right lives in $\mathbb{R}$. Then we have, for $x, x' \in \mathbb{R}^{\mathcal{N}}$:

$$\begin{aligned} \mathcal{M}^\star(\mu,x,x',y) &= \int_0^1 \begin{pmatrix} B(\mu)^\top & df(x'+\alpha(x-x'))^\top \\ (0) & -1 \end{pmatrix} (y)\, d\alpha \\ &= \begin{pmatrix} B(\mu)^\top & (\int_0^1 df(x'+\alpha(x-x'))\, d\alpha)^\top \\ (0) & -1 \end{pmatrix} (y) \end{aligned}$$

The above formula cannot be simplified, in general. Except in special cases, the integral over $(0,1)$ therefore must be numerically computed. In Section 4 we will consider both cases, analytical (Section 4.1) or numerical computation (Section 4.2).

Below we provide examples for which an explicit formulation for the integral $\displaystyle\int_0^1 df(x'+\alpha(x-x'))\, d\alpha)$ is available.

Example 9 (Special case $\mathcal{N}=1$). *In the special case where $\mathcal{N}=1$ we can change variable in the integral:*

$$\int_0^1 df(x'+\alpha(x-x'))\, d\alpha = \frac{f(x)-f(x')}{x-x'}$$

Although this case is exceedingly simple (because for any numerical problem $\mathcal{N}>1$), this kind of simplification can happen in other cases, as we will see below.

Example 10 (Special cases $\displaystyle\int f$ explicit). *In some cases the above integral can also be explicitly computed. We give a few nonlinear examples below.*

1. *f additive: $f:\mathbb{R}^{\mathcal{N}}\to\mathbb{R}, x\mapsto f(x) = \displaystyle\sum_{i=1}^{\mathcal{N}} f_i(x_i)$ where f_i are $\mathbb{R}\to\mathbb{R}$ differentiable functions. In that case, the previous change of variable still applies, and we get:*

$$\begin{aligned} \int_0^1 df^T(x'+\alpha(x-x'))\, d\alpha &= \int_0^1 (f_1'(x_1'+\alpha(x_1-x_1')),\ldots)\, d\alpha \\ &= \left(\frac{f_1(x_1)-f_1(x_1')}{x_1-x_1'},\ldots,\frac{f_{\mathcal{N}}(x_{\mathcal{N}})-f_{\mathcal{N}}(x_{\mathcal{N}}')}{x_{\mathcal{N}}-x_{\mathcal{N}}'}\right) \end{aligned}$$

For example:

(a) $f : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}, x \mapsto f(x) = \sum_{i=1}^{\mathcal{N}} x_i^2$, *then we get*

$$\int_0^1 df^T(x' + \alpha(x - x'))\, d\alpha = (x_1 + x_1', x_2 + x_2', \dots, x_{\mathcal{N}} + x_{\mathcal{N}}')$$

(b) $f : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}, x \mapsto f(x) = \sum_{i=1}^{\mathcal{N}} e^{x_i}$, *and it holds*

$$\int_0^1 df^T(x' + \alpha(x - x'))\, d\alpha = \left(\frac{e^{x_1} - e^{x_1'}}{x_1 - x_1'}, \dots, \frac{e^{x_{\mathcal{N}}} - e^{x_{\mathcal{N}}'}}{x_{\mathcal{N}} - x_{\mathcal{N}}'} \right)$$

2. $f : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}, x \mapsto f(x) = \left(\sum_{i=1}^{\mathcal{N}} x_i^2 \right)^{1/2}$. *Then it holds*

$$\begin{aligned} & \int_0^1 df^T(x' + \alpha(x - x'))\, d\alpha \\ = & \int_0^1 \frac{1}{\left(\sum_{i=1}^{\mathcal{N}} (x_i' + \alpha(x_i - x_i'))^2 \right)^{1/2}} (x_1' + \alpha(x_1 - x_1'), \dots)\, d\alpha \end{aligned}$$

which can therefore be explicitly computed as a function of x and x' coordinates:

$$\Big(x_i'\gamma a^{-1/2} - x\sqrt{c}a^{-1} + \frac{1}{2}\gamma a^{-3/2} + x_i'\sqrt{c}a^{-1} - \frac{1}{2} x_i' b\gamma a^{-3/2} + x_i'\delta a^{-1/2}$$
$$+ x_i\sqrt{a+b+c}\,a^{-1} - \frac{1}{2} x_i b\delta a^{-3/2} - x_i'\sqrt{a+b+c}\,a^{-1} + \frac{1}{2} x_i' b\delta a^{-3/2} \Big)_{i=1,\dots,\mathcal{N}}$$

where:

$$a = \sum_{i=1}^{\mathcal{N}} (x_i - x_i')^2, \quad b = 2\sum_{i=1}^{\mathcal{N}} x_i(x_i - x_i'), \quad c = \sum_{i=1}^{\mathcal{N}} {x_i'}^2,$$

$$\gamma = \ln \frac{b + 2\sqrt{ac}}{\sqrt{a}}, \delta = \ln \frac{b + 2a + 2\sqrt{a+b+c}\sqrt{a}}{\sqrt{a}}.$$

Dual error bound in the context of a nonlinear output. Let us come back to our initial purpose, that is the extension of our procedure to the context of a nonlinear output. The adjoint problem writes:

$$\mathcal{M}^\star(\mu, \widetilde{u}(\mu), u(\mu), w(\mu)) = \ell = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in X \times \mathbb{R}.$$

In a general context, the existence of a solution to this problem is not trivial, and may fail. However, if the operator $\mathcal{H}$ is linear, even if the output is nonlinear, as the adjoint problem writes equivalently:

$$\left\{ \begin{array}{rcll} B(\mu)^\top \overline{w} + \displaystyle\int_0^1 df(u + s(\widetilde{u} - u))^\top \underline{w}\, ds & = & 0 & \} \quad \mathcal{N} \text{ equations} \\ -\underline{w} & = & 1 & \} \quad 1 \text{ equation} \end{array} \right. ,$$

the unicity of the solution is provided as soon as $B(\mu)$ is invertible. In other words, $\overline{w}$ is equal to:

$$\overline{w} = B(\mu)^{-\top} \int_0^1 df(\widetilde{u} + s(u - \widetilde{u}))^\top\, ds.$$

3.5. Efficient bound evaluation in a many-query or real-time context. In practice, the error bound $\epsilon(\mu;\alpha)$ used in Theorem 4 can not be directly evaluated, and one has to define a computable approximation $\widehat{\epsilon}(\mu;\alpha)$. Our approximation is justified and commented in [10] Section 1.3, and we recall it here for sake of self-containedness. We end this section with Lemma 11, which gives sufficient conditions to ensure efficient computation of our online error bound.

Estimation of $\mathbf{\Phi}^G$. We consider a finite subset of parameters $\Xi \subset \mathcal{P}$, randomly sampled from the probability distribution P, and we estimate the linear operator $G : Y \to Y$ by a linear operator $\hat{G} : Y \to Y$ defined as:

$$\forall \varphi \in Y, \hat{G}\varphi = \frac{1}{2\#\Xi} \sum_{\mu\in\Xi} \left(\langle r(\mu), \varphi\rangle_Y r(\mu) + \langle w(\mu), \varphi\rangle_Y w(\mu) \right)$$

and we take as $\{\phi_i\}_{i=1,\dots,K}$ the unit eigenvectors of $\hat{G}$ associated with its K largest eigenvalues. The computation of these eigenvectors can be entirely processed during the offline phase (see [10, Section 1.3] for more details).

Computation of $T_1(\mu, K, \mathbf{\Phi})$. Recall that

$$T_1(\mu, K, \mathbf{\Phi}) = \max \left(\left|T_1^{up}(\mu, K, \mathbf{\Phi})\right|, \left|T_1^{low}(\mu, K, \mathbf{\Phi})\right| \right)$$

with

$$\begin{cases} T_1^{up}(\mu, K, \mathbf{\Phi}) = \displaystyle\sum_{i=1}^{K} \langle r(\mu), \phi_i\rangle_Y \beta_i^{up}(\mu, \mathbf{\Phi})\,, \\ T_1^{low}(\mu, K, \mathbf{\Phi}) = \displaystyle\sum_{i=1}^{K} \langle r(\mu), \phi_i\rangle_Y \beta_i^{low}(\mu, \mathbf{\Phi})\,. \end{cases}$$

The $\beta(\mu, \mathbf{\Phi})$ values can be approximated using a simple discrete minimization (ie., replacing $\mathcal{P}$ by a discrete sample Ξ in the minimum/maximum defining $\beta^{max}(\mathbf{\Phi})$ and $\beta^{min}(\mathbf{\Phi})$). In some cases, one can use a continuous optimization method to solve these minimum/maximum problems. It is clear that all these computations can be done during the offline phase.

We now discuss the computation of the K scalar products $\langle r(\mu), \phi_i\rangle_Y$ for $i = 1, \dots, K$ with an offline/online procedure. Recall that $\widetilde{X}$ is a subspace of X, of dimension N such that $N << \mathcal{N}$.

LEMMA 11. *Let $\{y_1, \dots, y_{\mathcal{S}}\}$ denote an orthonormal basis of Y and $\{x_1, \dots, x_{\mathcal{N}}\}$ denote an orthonormal basis of X. Assume that $\mathcal{M} : \mathcal{P} \times X \to Y$ is defined by:*

$$\mathcal{M}\left(\mu, \sum_{i=1}^{\mathcal{N}} v_i x_i\right) = \sum_{j=1}^{\mathcal{S}} m_j(\mu, v_1, \dots, v_{\mathcal{N}}) y_j\,. \tag{14}$$

where for all $j = 1, ..., \mathcal{S}$, m_j is a function from $\mathcal{P} \times \mathbb{R}^{\mathcal{N}}$ to $\mathbb{R}$.

Assume moreover that: $\forall\, j = 1, \dots, \mathcal{S}$, $\forall\, \mu \in \mathcal{P}$, $\forall (v_1, \dots, v_{\mathcal{N}}) \in \mathbb{R}^{\mathcal{N}}$,

$$m_j(\mu, v_1, \dots, v_{\mathcal{N}}) = \sum_{k=0}^{T_j} Q_{k,j}(v_1, \dots, v_{\mathcal{N}}) h_k(\mu) \tag{15}$$

with

$$h_k : \mathcal{P} \to \mathbb{R}, \quad \forall k \in \{0, \dots, T_j\}$$

and

$$Q_{k,j}(v_1,\dots,v_{\mathcal{N}}) = \sum_{\alpha=(\alpha_1,\dots,\alpha_{\mathcal{N}})\in I_{j,k}} q_{j,k,\alpha} \prod_{l\in V_\alpha} \varphi_{\alpha_l}(v_l) \tag{16}$$

where:

$$\begin{aligned} &I_{j,k}\subset \mathbb{N}^{\mathcal{N}}, \quad I=\bigcup_{j=1}^{\mathcal{S}}\bigcup_{k=1}^{T_j} I_{j,k}, \quad \#I=M \\ &V_\alpha \subset \{1,\dots,\mathcal{N}\}, \quad \max_{\alpha\in I}\#\{V_\alpha\}=L \end{aligned} \tag{17}$$

and $\{\varphi_l\}_{l\in\{1,\dots,\mathcal{N}\}}$ a set of functions from $\mathbb{R}$ to $\mathbb{R}$. We set $T=\max_{j=1,\dots,\mathcal{S}} T_j$. Assume that, for $\alpha\in I$, $l\in V_\alpha$, the cost of computing $\varphi_{\alpha_l}(v_l)$ is bounded by some positive constant R. Then, it is possible to compute each of the scalar products $\langle r(\mu),\phi_i\rangle_Y$ for $i=1,\dots,K$, with an offline/online procedure whose online phase has the cost $T\times M\times L\times R$.

Proof of Lemma 11. The proof is postponed to Appendix A.2. □

Let us now comment the result and the assumptions in Lemma 11 with a few remarks.

REMARK 12. *The decomposition (15) plays an analogous role to the "affine parameter dependence" that is commonly assumed in the literature (see, e.g., [18], page 1526) on linear problems.*

In case Decomposition (15) does not hold true anymore but one still wants a complexity independent from the full dimension $\mathcal{N}$, we may employ EIM (Empirical Interpolation Methods) introduced in [1] to recover the required affine decomposition in the parameter μ.

REMARK 13. *Equations (16) and (17) imply that the functions $\{Q_{k,j}(v_1,\dots,v_{\mathcal{N}})\}_{1\le j\le\mathcal{N},1\le k\le T_j}$ admit a sparse representation on $\{\varphi_{\alpha_l}(v_l)\}_{\alpha\in I,l\in V_\alpha}$ and that the interaction order is bounded by L. Let us emphasize that the result in Lemma 11 implies that the cost does not depend on the high dimension $\mathcal{N}$. Therefore if we assume that T, M, L and $R << \mathcal{S}$, then it is possible to compute the K scalar products, with an offline/online procedure with a small cost (with respect to $\mathcal{S}$). A bound for the cost in the particular case where $\varphi_{\alpha_l}(v_l)=v_l^{\alpha_l}$ is provided in Remark 15 in Appendix A.2.*

REMARK 14. *Obtaining such a sparse representation for the $\{Q_{k,j}(v_1,\dots,v_{\mathcal{N}})\}_{1\le j\le\mathcal{N},1\le k\le T_j}$ is not an obvious task and requires the use or development of a high-dimensional approximation tool. Assuming that the interaction order is bounded by L may also be questionable for some applications. Note that, in case these assumptions are not satisfied, it is still possible to work with the K scalar products themselves, without any approximation. In that case, the cost of the online phase is $\mathcal{O}(\mathcal{S})$, which is still better than the full problem, whose complexity is $\mathcal{O}(\mathcal{S}^\alpha)$ with $\alpha\ge 2$ in most cases.*

In the context of nonlinear partial differential equations, DEIM, a discrete variant of EIM, was suggested and analyzed in [4], allowing to reduce the computational complexity of the reduced order model due to its dependence on the nonlinear full dimension model $\mathcal{N}$. Contrarily to DEIM, our approach is not specific to projection-based metamodels.

Approximation of $T_2(K, \mathbf{\Phi})$. A Monte-Carlo estimator of $T_2(K, \mathbf{\Phi})$ is used:

$$\hat{T}_2(K, \mathbf{\Phi}) = \frac{1}{2\#\Xi} \sum_{\mu \in \Xi} \left| \widetilde{s}(\mu) - s(\mu) - \sum_{i=1}^{K} \langle r(\mu), \phi_i \rangle_Y \langle w(\mu), \phi_i \rangle_Y \right|,$$

where Ξ is a sample of $\mathcal{P}$.

As this quantity is μ-independent, it can be computed for once during the offline phase. Of course this computation is a numerical estimation, as such it can induce approximation errors. Such error analysis, which is related to the central limit theorem, is discussed in [10, Section A].

Computable error bound We now rely on Theorem 4 and set:

$$\hat{\epsilon}(\mu; \alpha) = T_1(\mu, K, \mathbf{\Phi}) + \frac{\hat{T}_2(K, \mathbf{\Phi})}{\alpha}.$$

It is an estimator for the error bound $\epsilon(\mu; \alpha)$ in Theorem 4. As before, the error associated to the approximation is analyzed and discussed in [10, Section A].

4. Numerical experiments.

4.1. First experiments with a toy model.

We now apply our error bound on a non-homogeneous linear transport equation with a nonlinear output. We use the results of Section 3.3.

4.1.1. Toy model. Let $u_e = u_e(t, x)$ be the solution of the linear transport equation:

$$\frac{\partial u_e}{\partial t}(t, x) + \mu \frac{\partial u_e}{\partial x}(t, x) = \sin(x) \exp(-x)$$

for all $(t, x) \in (0, 1) \times (0, 1)$, satisfying the initial condition:

$$u_e(t = 0, x) = x(1 - x) \quad \forall x \in [0, 1],$$

and boundary condition:

$$u_e(t, x = 0) = 0 \quad \forall t \in [0, 1].$$

The parameter μ is chosen in $\mathcal{P} = [0.5, 1]$ and $\mathcal{P}$ is endowed with the uniform measure.

We choose a number of timesteps N_t and a number of space points N_x, we set $\Delta_t = 1/N_t$ and $\Delta_x = 1/N_x$ and we introduce the discrete unknown vector $u = (u_i^n)_{i=0,\ldots,N_x; n=0,\ldots,N_t}$. We note here that the considered PDE is an hyperbolic evolution equation , and that we perform the reduction on the space-time unknown u, of dimension $\mathcal{N} = (N_x + 1) \cdot (N_t + 1)$. This is different from reducing the space-discretized equation at each time step.

The u vector satisfies the discretized initial-boundary conditions:

$$(18) \qquad \forall i, \;\; u_i^0 = i\Delta_x(1 - i\Delta_x)$$

$$(19) \qquad \forall n, \;\; u_0^n = 0$$

and the first-order upwind scheme implicit relation:

$$(20) \qquad \forall i, n \;\; \frac{u_{i+1}^{n+1} - u_{i+1}^n}{\Delta_t} + \mu \frac{u_{i+1}^{n+1} - u_i^{n+1}}{\Delta_x} = \sin(i\Delta_x) \exp(-i\Delta_x).$$

Let $B(\mu)$ (resp. ϕ) be the matrix (resp. the vector) so that (18), (19) and (20) are equivalent to:

$$B(\mu)u = \phi \in \mathbb{R}^N \text{ with } N = N_x \times N_t. \tag{21}$$

We consider the different outputs of interest of Example 2 in Section 3.3:

- Square output: $s(\mu) = \left(u_{N_x}^{N_t}\right)^2$
- Exponential output: $s(\mu) = \exp\left(u_{N_x}^{N_t}\right)$
- Triple exponential output: $s(\mu) = \exp\left(3u_{N_x}^{N_t}\right)$

In the following, we take $\Delta_t = 0.02$ and $\Delta_x = 0.05$.

4.1.2. Reduction. The approximation $\widetilde{u}$ of u is computed by using a "reduced basis" approach [18]. To be more specific, $\widetilde{u}$ is the solution of:

$$Z^t B(\mu) Z \widetilde{u} = Z^t \phi,$$

where Z is an appropriate matrix found by Proper Orthogonal Decomposition (POD) (see [23] for instance). The Z matrix is the matrix of an orthogonal set of N vectors in $X = \mathbb{R}^{\mathcal{N}}$, endowed with the Euclidian scalar product. The N number is called the *reduced basis size*.

The Z matrix is computed using a POD snapshot of size 70, and $K = 20$ retained $\widehat{\phi}_i^G$ vectors. We took a very low risk level $\alpha = 0.0001$.

4.1.3. Results. In the following, the true error for a given parameter μ is defined as $|s(\mu) - \widetilde{s}(\mu)|$, and the error bound as $\hat{\epsilon}(\mu; \alpha)$.

Let us underline that in the present study the error is given *with respect to the full discretized solution* (and not to the theoretical solution of the partial differential equation).

In Figure 1, we plotted, as functions of the reduced basis size, the averaged effectivity, that is defined by

$$\overline{\text{Eff}}_{N_{\text{test}}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \frac{\hat{\epsilon}(\mu_i; \alpha)}{|s(\mu_i) - \widetilde{s}(\mu_i)|}$$

as far as the standard deviation of the effectivity ($S_{\text{Eff},N_{\text{test}}}$), that is defined by

$$S^2_{\text{Eff},N_{\text{test}}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left(\frac{\hat{\epsilon}(\mu_i; \alpha)}{|s(\mu_i) - \widetilde{s}(\mu_i)|} - \overline{\text{Eff}}_{N_{\text{test}}} \right)^2$$

with $N_{\text{test}} = 200$ the size of the random sample of parameter values $\mu_1, \ldots, \mu_{N_{\text{test}}}$, for three different output cases (square, exponential and triple exponential).

The graphs show that the averaged effectivity decreases at the logarithmic scale almost linearly as the size of the reduced basis increases. We see that our error bound becomes sharp (close to one) as the size of the reduced basis increases, despite the highly-nonlinear output functions that have been chosen (yet, it seems almost unaffected by the degree of nonlinearity in the output). The standard deviation also decreases almost linearly at the logarithmic scale.

4.2. Nonlinear models. In this section, we illustrate the results of Section 3.2 on the discretized non-viscous Burgers equation, as an example of nonlinear model.

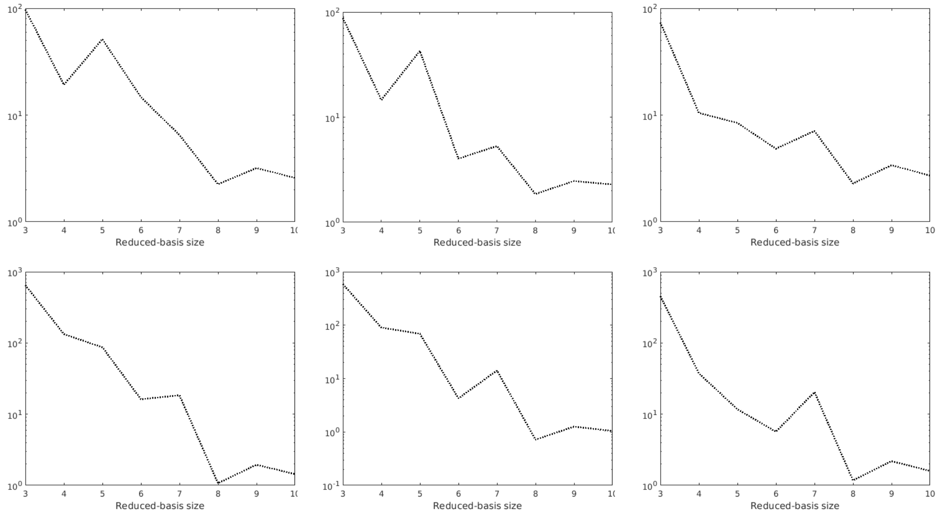

Fig. 1. *We plot (on a logarithmic scale) (top) the averaged effectivity* $\overline{\mathrm{Eff}}_{N_{\mathrm{test}}}$ *with* $N_{\mathrm{test}} = 200$ *in the square (top left), the exponential (top middle) and the triple exponential (top right) output case, as functions of the reduced basis size; (down) the standard deviation of the effectivity* $S_{\mathrm{Eff},N_{\mathrm{test}}}$ *with* $N_{\mathrm{test}} = 200$ *in the square (down left), the exponential (down middle) and the triple exponential (down right) output case, as functions of the reduced basis size.*

There exist various results for the application of the certified reduced basis method to nonlinear problems (see, e.g., [25, 17, 9, 28, 29]). In these papers, the authors study error bounds specifically designed for (linearized or not) (viscous) Burgers equation, and linear or quadratic outputs. We apply here a space-time reduction procedure. We then show numerical results obtained for our probabilistic error bound, with a linear output.

4.2.1. Description of the model and output of interest. We are looking for $u = u(t, x)$ satisfying:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{1}{2}\dfrac{\partial (u^2)}{\partial x} = 0, \quad \forall (t,x) \in (0,1)\times(0,1) \\ u(t, x=0) = 1, \quad \forall t \in [0,1] \\ u(t=0, x) = \cos^2(\alpha x) + \beta x, \quad \forall x \in [0,1] \end{cases}$$

where the parameter vector $\mu = (\alpha, \beta)$ belongs to $[0,1]\times[0,1]$ and follows an uniform law.

We discretize the above equation by using an upwind scheme. We choose a number of timesteps N_t and a number of space points N_x, and we set $\Delta_t = 1/N_t$ and $\Delta_x = 1/N_x$, and we look for $(u_i^n)_{i,n}$, where $i = 0, \dots, N_x - 1$ and $n = 0, \dots, N_t - 1$ so that:

$$\begin{cases} \dfrac{u_i^{n+1} - u_i^n}{\Delta_t} + \dfrac{1}{2}\dfrac{\left(u_i^{n+1}\right)^2 - \left(u_{i-1}^{n+1}\right)^2}{\Delta_x} = 0 \;\; \forall (i,n) \in \{1, ..., N_x - 1\} \times \{0, ..., N_t - 2\} \\ u_0^n = 1 \;\; \forall n \in \{0, ..., N_t - 1\} \\ u_i^0 = \cos^2(\alpha i \Delta_x) + \beta i \Delta_x \;\; \forall i \in \{0, ..., N_x - 1\} \end{cases}$$

The output functional of interest is given by the ℓ vector defined by:

$$\ell_i^n = \begin{cases} \dfrac{1}{N_x} \text{ if } n = N_t - 1, \forall i \in \{0, ..., N_x - 1\} \\ 0 \text{ else.} \end{cases}$$

In other words, the output of interest is the spatial average at final time.

4.2.2. Reduction. As for the toy model, the reduction is performed on the full space-time state vector $(u_i^n)_{i,n}$. We also choose a Z matrix by a POD procedure, then define the reduced state vector $(\widetilde{u}_i^n)_{i,n}$ as:

$$(\widetilde{u}_i^n)_{i,n}(\mu) = \text{argmin}_{v \in \text{Range}(Z)} ||\mathcal{M}(\mu, v)||^2$$

where Range(Z) is the column space of Z, and $|| \cdot ||^2$ denotes the Euclidean norm.

We note here that we do not claim that this reduction is the state-of-the-art for Burgers equation, as model reduction by itself is not the goal of this paper.

4.2.3. Numerical experiments. Table 1 gives the name and description of the various parameters used in the numerical code. Table 2 describes the various experiments that have been performed and provides references to the associated figures.

Parameter	Description	Usual range
N_x	Number of space discretization points	$40 - 80$
N_t	Number of time steps	$10 - 20$
N_{test}	Monte-Carlo sample size	100
N_{snap}	Size of the POD training sample set	70
N_ϕ	Index K for the estimation of T_1 using basis ϕ_G	8
N_{basis}	Size of the POD basis	$3 - 10$
Δ_t	Time step	$\Delta_t = 1/N_t$
Δ_x	Space step	$\Delta_x = 1/N_x$

Table 1
Descriptions of the numerical parameters.

Experiment label	N_t	N_x	N_{test}	N_{snap}	N_ϕ	Figure
(a) $t10 \times x40$	10	40	100	70	8	2
(b) $t20 \times x40$	20	40	100	70	8	2
(c) $t10 \times x80$	10	80	100	70	8	3
(d) $t20 \times x80$	20	80	100	70	8	3

Table 2
Numerical setup of the different experiments.

Let us define the averaged true error and the averaged error bound as

$$\overline{e}_{N_{\text{test}}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} |s(\mu_i) - \widetilde{s}(\mu_i)| \text{ and } \overline{\hat{\epsilon}}_{N_{\text{test}}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \hat{\epsilon}(\mu_i; \alpha),$$

as far as the standard deviation of the true error $(S_{e,N_{\text{test}}})$ and the standard deviation of the error bound $(S_{\hat{\epsilon},N_{\text{test}}})$, that means

$$S_{e,N_{\text{test}}}^2 = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left(|s(\mu_i) - \widetilde{s}(\mu_i)| - \overline{e}_{N_{\text{test}}}\right)^2$$

$$\text{and } S^2_{\hat{\epsilon},N_{\text{test}}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left(\hat{\epsilon}(\mu_i;\alpha) - \bar{\hat{\epsilon}}_{N_{\text{test}}}\right)^2,$$

with $N_{\text{test}} = 100$ the size of the random sample of parameter values $\mu_1, \dots, \mu_{N_{\text{test}}}$.

These quantities are plotted in Figure 2 (*resp.* 3), for a size of the POD truncated basis varying from 3 to 10, with $N_t = 10, 20$, $N_x = 40$ (*resp.* $N_x = 80$) and other parameters described in Table 2 (as before, the error is given with respect to the full discretized solution). Note that the averaged true error and the averaged error bound (at the logarithmic scale) decrease with the same slope as the size of the POD basis increases. Also the standard deviation of the true error and the one of the error bound (at the logarithmic scale) decrease with the same slope as the size of the POD increases. Moreover, the standard deviation has the same order of magnitude as the mean, both decreasing linearly with the size of the POD basis.

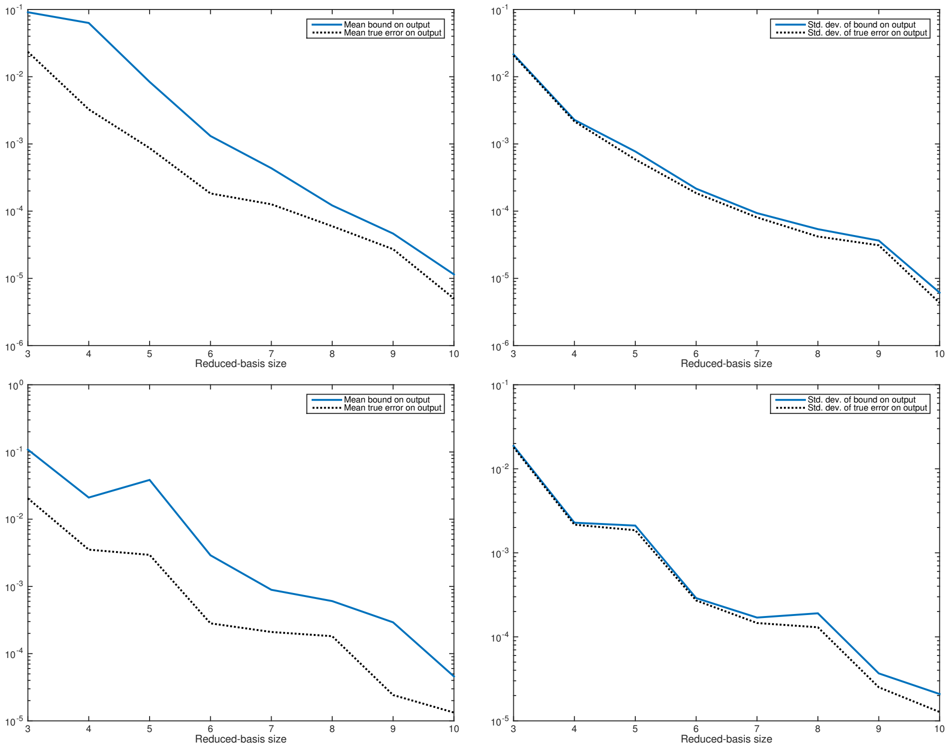

FIG. 2. *We plot (on a logarithmic scale) (left) the averaged true error $\bar{e}_{N_{\text{test}}}$ (dashed line) and the error bound $\bar{\hat{\epsilon}}_{N_{\text{test}}}$ (plain line) with $N_{\text{test}} = 100$ for experiments (a) $t10 \times x40$ (top) and (b) $t20 \times x40$ (bottom), as functions of the size of the POD basis; (right) the standard deviation of the true error $S_{e,N_{\text{test}}}$ (dashed line) and of the error bound $S_{\hat{\epsilon},N_{\text{test}}}$ (plain line) with $N_{\text{test}} = 100$ for experiments (a) $t10 \times x40$ (top) and (b) $t20 \times x40$ (bottom), as functions of the size of the POD basis.*

To quantify the computing gain we define and compute the following speed-up ratios. The first ratio r_1 is suitable to study real-time problems computing gain:

$$r_1 = \frac{\text{full pb computing time}}{\text{online computing time}}$$

Indeed for real-time problem the offline cost is not an issue, and one is really interested in the online acceleration.

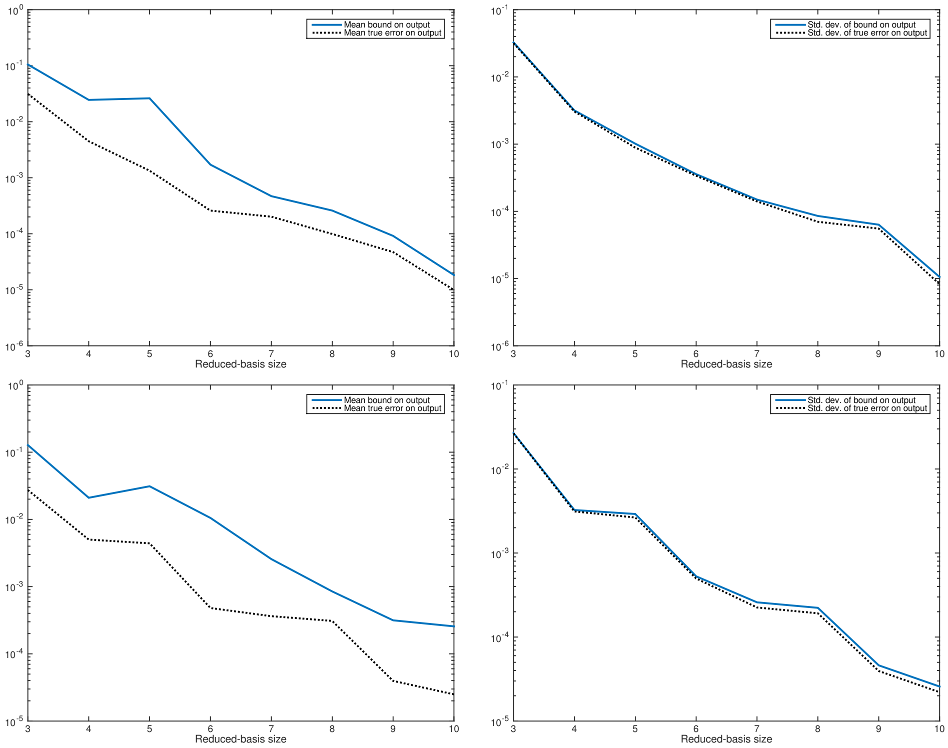

FIG. 3. *We plot (on a logarithmic scale) (left) the averaged true error $\overline{e}_{N_{\text{test}}}$ (dashed line) and the error bound $\overline{\hat{\epsilon}}_{N_{\text{test}}}$ (plain line) with $N_{\text{test}} = 100$ for experiments (c) $t10 \times x80$ (top left) and (d) $t20 \times x80$ (bottom left), as functions of the size of the POD basis; (right) standard deviation of the true error $S_{e,N_{\text{test}}}$ (dashed line) and of the error bound $S_{\hat{\epsilon},N_{\text{test}}}$ (plain line) with $N_{\text{test}} = 100$ for experiments (c) $t10 \times x80$ (top right) and (d) $t20 \times x80$ (bottom right), as functions of the size of the POD basis.*

On the contrary, for many-query problems, the total computing time is the quantity of interest, and we shall therefore define and compute the second speed-up ratio r_2:

$$r_2 = \frac{K \times \text{full pb computing time}}{\text{offline} + K \times \text{online computing time}}$$

with $K = 1000$.

The larger the speed-up ratios, the more efficient the use of a reduction procedure is. In our experiments, the computing time were computed using Matlab cputime function. We summarize in Table 3 the full, online and offline costs, as well as the speed-up ratios, for the various experiments described in Table 2. Note that the full problem is solved by an implicit nonlinear scheme, which probably explains that the time required to solve the full problem depends not only on N_x and N_t but on the interaction between both. Contrary to the full problem computation time, the online computation time remains low as the number of time and space discretization points increase, ranging from 14 to 19. Thus the speed-up ratio r_1 increases with N_x and N_t, starting around 12 and up to 850. The offline computation time seems to be around five times larger than the full problem computation time. Although it becomes non negligible as N_x and N_t increase, even higher than the full problem computation time, the increase of the speed-up ratio r_2 still proves the efficiency of the reduction for solving many-query problems.

Experiment name	(a)	(b)	(c)	(d)
	$t10 \times x40$	$t20 \times x40$	$t10 \times x80$	$t20 \times x80$
full pb computing time	168.2	863.0	976.5	16430.0
online computing time	14.1	16.1	15.5	19.3
offline computing time	931.9	4314.0	4713.4	77817.0
speed-up ratio r_1	11.9	53.5	63.1	849.1
speed-up ratio r_2	11.2	42.2	48.4	169.1
Figure	2	2	3	3

Table 3

Table of costs using Matlab cputime function, for a size of the truncated POD equal to 8.

5. Conclusion. A class of nonlinear problems depending on a probabilistic vector has been considered, and a numerically efficient method has been designed to compute the error estimation, when approximating the output error. This method is based on two phases. The offline phase requires to compute the solution of a high-dimensional problem, and the online phase is based on the computation of the solution of a reduced-order problem. This approach has been applied to a toy model and to a nonlinear partial differential equation, namely the Burgers equation parametrized by two probabilistic coefficients. An application of this numerical method to other mathematical problems is under investigation, more precisely, it could be fruitful to investigate the impact of this new result in control theory (as done in [11] for a linear problem). Perspectives in environmental modeling, among other domains where the sensitivity analysis is crucial, are also worth considering. Also other infinite dimensional nonlinear problems could be considered, as those described by a nonlinear partial differentiable equation, where shocks may appear (it was not the case for the Burgers equation studied here.)

Acknowledgments. We would like to thank Frédéric Lagoutière for his suggestion on the choice of the numerical scheme for the Burgers equation. We also thank the anonymous referees for their valuable suggestions.

REFERENCES

[1] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An empirical interpolation-method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339 (2004), pp. 667–672.

[2] R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica 2001, 10 (2001), pp. 1–102.

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

[4] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.

[5] M. Drohmann and K. Carlberg, *The romes method for statistical modeling of reduced-order-model error*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 116–145.

[6] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*, Springer-Verlag, New York, 1991, doi:10.1007/978-1-4612-3094-6, http://dx.doi.org/10.1007/978-1-4612-3094-6.

[7] M. Ilak and C. Rowley, *Modeling of transitional channel flow using balanced proper orthogonal decomposition*, Physics of Fluids, 20 (2008), p. 034103.

[8] A. Janon, M. Nodet, and C. Prieur, *Certified reduced-basis solutions of viscous burgers equation parametrized by initial and boundary values*, ESAIM: Mathematical Modelling and Numerical Analysis, 47 (2013), pp. 317–348.

[9] A. Janon, M. Nodet, and C. Prieur, *Certified reduced-basis solutions of viscous Burgers equations parametrized by initial and boundary values*, ESAIM Math. Model. Numer. Anal., 47 (2013), pp. 317–348.

[10] A. Janon, M. Nodet, and C. Prieur, *Goal-oriented error estimation for the reduced basis method, with application to sensitivity analysis*, Journal of Scientific Computing, 68 (2016), pp. 21–41.

[11] A. Janon, M. Nodet, C. Prieur, and C. Prieur, *Global sensitivity analysis for the boundary control of an open channel*, Mathematics of Control, Signals, and Systems, 28 (2016), pp. 1–27.

[12] J. Kleijnen, *Design and analysis of simulation experiments*, Springer Publishing Company, Incorporated, 2007.

[13] J. Kleijnen, *Simulation experiments in practice: statistical design and regression analysis*, Journal of Simulation, 2 (2008), pp. 19–27.

[14] O. P. Le Maître, O. M. Knio, B. J. Debusschere, H. N. Najm, and R. G. Ghanem, *A multigrid solver for two-dimensional stochastic diffusion equations*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 4723–4744, doi:10.1016/S0045-7825(03)00457-2, http://dx.doi.org/10.1016/S0045-7825(03)00457-2.

[15] Y. Maday, A. Patera, and D. Rovas, *A blackbox reduced-basis output bound method for noncoercive linear problems*, in Nonlinear Partial Differential Equations and their Applications Collège de France Seminar Volume XIV, D. Cioranescu and J.-L. Lions, eds., vol. 31 of Studies in Mathematics and Its Applications, Elsevier, 2002, pp. 533 – 569, doi:10.1016/S0168-2024(02)80025-X, http://www.sciencedirect.com/science/article/pii/S016820240280025X.

[16] B. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Transactions on Automatic Control, 26 (1981), pp. 17–31.

[17] N. Nguyen, G. Rozza, and A. Patera, *Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation*, Calcolo, 46 (2009), pp. 157–185.

[18] N. Nguyen, K. Veroy, and A. Patera, *Certified real-time solution of parametrized partial differential equations*, in Handbook of Materials Modeling, Springer, 2005, pp. 1523–1558.

[19] A. Nouy, *Low-rank tensor methods for model order reduction*, arXiv preprint arXiv:1511.01555, (2015).

[20] T. Santner, B. Williams, and W. Notz, *The design and analysis of computer experiments springer-verlag*, New York. 283pp, (2003).

[21] J. M. A. Scherpen and W. S. Gray, *Nonlinear Hilbert adjoints: properties and applications to Hankel singular value analysis*, Nonlinear Anal., 51 (2002), pp. 883–901.

[22] M. Scheuerer, R. Schaback, and M. Schlather, *Interpolation of spatial data – a stochastic or a deterministic problem?*, European Journal of Applied Mathematics, 24 (2013), pp. 601–629.

[23] L. Sirovich, *Turbulence and the dynamics of coherent structures. part i-ii*, Quarterly of applied mathematics, 45 (1987), pp. 561–590.

[24] C. Soize and R. Ghanem, *Physical systems with random uncertainties: chaos representations with arbitrary probability measure*, SIAM J. Sci. Comput., 26 (2004), pp. 395–410, doi:10.1137/S1064827503424505, http://dx.doi.org/10.1137/S1064827503424505.

[25] K. Veroy, C. Prud'homme, and A. Patera, *Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds*, Comptes Rendus Mathematique, 337 (2003), pp. 619–624.

[26] S. Volkwein, *Proper orthogonal decomposition and singular value decomposition.* Spezialforschungsbereich F003 Optimierung und Kontrolle, Projektbereich Kontinuierliche Optimierung und Kontrolle, Bericht. Nr. 153, Graz, 1999.

[27] K. Willcox and J. Peraire, *Balanced model reduction via the proper orthogonal decomposition*, AIAA journal, 40 (2002), pp. 2323–2330.

[28] M. Yano and A. T. Patera, *A space–time variational approach to hydrodynamic stability theory*, in Proc. R. Soc. A, vol. 469, The Royal Society, 2013, p. 20130036.

[29] M. Yano, A. T. Patera, and K. Urban, *A space-time hp-interpolation-based certified reduced basis method for burgers' equation*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 1903–1935.

[30] D. Zupanski and M. Zupanski, *Model error estimation employing an ensemble data assimilation approach*, Monthly Weather Review, 134 (2006), pp. 1337–1354, doi:10.1175/

MWR3125.1.

Appendix A. Postponed proofs.

A.1. Proof of Proposition 2. For all $\mu \in \mathcal{P}, x, y \in X, z \in Y$ we have:

$$\begin{aligned}
\langle x-y, \mathcal{M}^\star(\mu,x,y,z)\rangle_X &= \langle x-y, \int_0^1 d\mathcal{M}^\star(y+s(x-y))(z)\,ds\rangle_X \\
&= \int_0^1 \langle d\mathcal{M}(y+s(x-y))(x-y), z\rangle_Y\,ds \\
&= \langle \int_0^1 d\mathcal{M}(y+s(x-y))(x-y)\,ds, z\rangle_Y \\
&= \langle \mathcal{M}(x)-\mathcal{M}(y), z\rangle_Y
\end{aligned}$$

□

A.2. Proof of Lemma 11. Let us recall the formula for the residual:

$$r(\mu) = \mathcal{M}(\mu, \widetilde{u}(\mu)) \tag{22}$$

so that the scalar products we need to compute are, for all i:

$$\langle r(\mu), \phi_i\rangle = \langle \mathcal{M}(\mu, \widetilde{u}(\mu)), \phi_i\rangle_Y \tag{23}$$

Here we describe the online/offline procedure to compute

$$\langle \mathcal{M}(\mu, v), \phi_i\rangle_Y \tag{24}$$

where $v \in \widetilde{X}$ and $\mu \in \mathcal{P}$ are given.
We also make all the asumptions of Lemma 11 regarding the decomposition of $\mathcal{M}$ and m_j. Using the decomposition (14) we have

$$\langle \mathcal{M}(\mu, v), \phi_i\rangle_Y = \sum_{j=1}^{\mathcal{S}} m_j(\mu, v)\langle y_j, \phi_i\rangle_Y \tag{25}$$

We replace (15) and (16) in (25):

$$\begin{aligned}
\langle \mathcal{M}(\mu, v), \phi_i\rangle_Y &= \sum_{j=1}^{\mathcal{S}}\sum_{k=0}^{T_j} Q_{k,j}(v)h_k(\mu)\langle y_j, \phi_i\rangle_Y \\
&= \sum_{j=1}^{\mathcal{S}}\sum_{k=0}^{T_j}\sum_{\alpha\in I_{j,k}} q_{j,k,\alpha}\left(\prod_{l\in V_\alpha}\varphi_{\alpha_l}(v_l)\right)h_k(\mu)\langle y_j, \phi_i\rangle_Y
\end{aligned} \tag{26}$$

Now we set:

$$q_{j,k,\alpha} = 0 \text{ if } \alpha \in I \setminus I_{j,k} \text{ or if } k > T_j$$

to get

$$\begin{aligned}
\langle \mathcal{M}(\mu, v), \phi_i\rangle_Y &= \sum_{j=1}^{\mathcal{S}}\sum_{k=0}^{T}\sum_{\alpha\in I} q_{j,k,\alpha}\left(\prod_{l\in V_\alpha}\varphi_{\alpha_l}(v_l)\right)h_k(\mu)\langle y_j, \phi_i\rangle_Y \\
&= \sum_{k=0}^{T} h_k(\mu)\sum_{\alpha\in I}\left(\prod_{l\in V_\alpha}\varphi_{\alpha_l}(v_l)\right)\sum_{j=1}^{\mathcal{S}} q_{j,k,\alpha}\langle y_j, \phi_i\rangle_Y
\end{aligned} \tag{27}$$

During the online phase we are given μ and v. The following quantities are independent of μ and v, therefore can be computed during the offline phase:

$$G_{\alpha,k,i} = \sum_{j=1}^{\mathcal{S}} q_{j,k,\alpha} \langle y_j, \phi_i \rangle_Y \text{ for all } k \in \{0, \dots, T\}, i \in \{1, \dots, \mathcal{S}\}, \alpha \in V_\alpha \tag{28}$$

and the online computation then writes:

$$\langle \mathcal{M}(\mu, v), \phi_i \rangle_Y = \sum_{k=0}^{T} h_k(\mu) \sum_{\alpha \in I} \left(\prod_{l \in V_\alpha} \varphi_{\alpha_l}(v_l) \right) G_{\alpha,k,i} \tag{29}$$

Looking back to (29) and using notations (17), the total operation count for the *online phase* is given by:

$$T \times M \times L \times R\,.$$

This concludes the proof of Lemma 11. □

Remark 15. *Let us consider the particular case (polynomial case) where for any $\alpha \in I$ and any $l \in V_\alpha$, $\varphi_{\alpha_l}(v_l) = v_l^{\alpha_l}$. Let us decompose v onto a basis $\{f_1, \dots, f_N\}$ of $\widetilde{X} \subset X$. First we write each f_k in the basis $\{x_1, \dots, x_{\mathcal{N}}\}$ of X:*

$$f_k = \sum_{i=1}^{\mathcal{N}} f_{k,i} x_i$$

Then we write v:

$$v = \sum_{k=1}^{N} v'_k f_k = \sum_{k=1}^{N} \sum_{i=1}^{\mathcal{N}} f_{k,i} v'_k x_i \tag{30}$$

so that we can write:

$$v = \sum_{i=1}^{\mathcal{N}} v_i x_i, \qquad \textit{with } v_i = \sum_{k=1}^{N} f_{k,i} v'_k \tag{31}$$

Formula (16) requires v_l to the power α_l, so we use the multinomial formula to get:

$$v_l^{\alpha_l} = \left(\sum_{k=1}^{N} f_{k,\alpha_l} v'_k \right)^{\alpha_l} = \sum_{\beta \in B(N,\alpha_l)} \binom{\alpha_l}{\beta} \prod_{k=1}^{N} (v'_k f_{k,l})^{\beta_k} \tag{32}$$

using the multinomial indices and coefficients:

$$\begin{aligned} &B(N, \alpha_l) = \{\beta = (\beta_1, \dots, \beta_N) \in \mathbb{N}^N, \sum_{k=1}^{N} \beta_k = \alpha_l\} \\ &\binom{\alpha_l}{\beta} = \frac{\alpha_l!}{\beta_1! \dots \beta_N!} \end{aligned} \tag{33}$$

For the study of the complexity of the online computation (29)*, we first consider the cost for the computation of $v_l^{\alpha_l}$. Using equation (32), we get:*

$$v_l^{\alpha_l} = \sum_{\beta \in B(N,\alpha_l)} \binom{\alpha_l}{\beta} \prod_{k=1}^{N} (v'_k f_{k,l})^{\beta_k}$$

The product $\prod_{k=1}^{N}(v'_k f_{k,l})^{\beta_k}$ *costs (up to a multiplicative constant)* $\beta_1 + \ldots + \beta_N = \alpha_l$ *multiplications, so that the computation of* $v_l^{\alpha_l}$ *costs (up to a multiplicative constant)* $\#B(N,\alpha_l) \times \alpha_l$ *operations. We know that*

$$\#B(N,\alpha_l) = \left(\begin{array}{c} \alpha_l + N - 1 \\ N - 1 \end{array} \right)$$

so if we set

$$R' = \max_{\alpha \in I} \max_{l \in V_\alpha} \alpha_l \left(\begin{array}{c} \alpha_l + N - 1 \\ N - 1 \end{array} \right)$$

then the cost of computating $v_l^{\alpha_l}$ *is (up to a multiplicative constant) bounded by* R'*. Looking back to (29) and using notations (17), the total operation count for the* online phase *is bounded by:*

$$const. \times T \times M \times L \times R' .$$