



HAL
open science

Goal-oriented error estimation for parameter-dependent nonlinear problems

Alexandre Janon, Maëlle Nodet, Christophe Prieur, Clémentine Prieur

► To cite this version:

Alexandre Janon, Maëlle Nodet, Christophe Prieur, Clémentine Prieur. Goal-oriented error estimation for parameter-dependent nonlinear problems . ESAIM: Mathematical Modelling and Numerical Analysis, 2018, 52 (2), pp.705-728. 10.1051/m2an/2018003 . hal-01290887

HAL Id: hal-01290887

<https://hal.science/hal-01290887v1>

Submitted on 18 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 **GOAL-ORIENTED ERROR ESTIMATION FOR FAST**
2 **APPROXIMATIONS OF NONLINEAR PROBLEMS**

3 ALEXANDRE JANON*, MAËLLE NODET†, CHRISTOPHE PRIEUR‡, AND CLÉMENTINE
4 PRIEUR§

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

15 **Key words.** Nonlinear problem; sensitivity analysis; numerical computation; many-query con-
16 text; probabilistic estimator

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

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

28 In this paper we work in this context, namely providing fast numerical solutions
29 to given problems. We are not focused on HPC (high performance computing), we are
30 rather interested in accelerating existing numerical methods for nonlinear problems.

31
32 We focus on the procedures of accelerating existing numerical models. These
33 procedures are generally called “metamodelling”, “model reduction”, “dimension re-
34 duction”. It consists in replacing the existing model, called the “full” model, by a fast
35 approximation. There exist both stochastic and deterministic approaches to building
36 such approximations. On the stochastic part we can mention polynomial chaos ap-
37 proximation [18, 2, 9], Gaussian processes (including Kriging and RKHS –reproducing
38 kernel Hilbert spaces) [8, 14], low-rank tensor methods [13], etc. which all provide
39 cheap and fast approximations of the full model. On the deterministic side we can
40 cite the reduced basis method [10], POD (proper orthogonal decomposition) [19], bal-
41 anced truncation [11], etc. All these methods have in common that they provide a
42 way to build a numerical model which is faster than the full model.

*Laboratoire de Mathématiques d’Orsay (CNRS), Université Paris-Sud, Universit Paris Saclay,
France alexandre.janon@math.u-psud.fr

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

‡Gipsa-lab, CNRS, Grenoble, France Christophe.Prieur@gipsa-lab.fr

§Univ. Grenoble Alpes, INRIA, France Clementine.Prieur@imag.fr

43 Of course, accelerating the model is not the only aim these methods have. It
 44 is crucial that they also provide accurate approximations of the full model. The
 45 approximation error, i.e. the comparison between the full model and the metamodel,
 46 should ideally be certified and known by the user of the metamodel. In practice, some
 47 metamodeling methods only provide limited validation and certification so that the
 48 user has to take a leap of faith because there is no quantified guarantee about the
 49 metamodel accuracy. However, it is possible in some cases to design metamodels which
 50 include a certified error bound. In this latter case, the user does not know exactly
 51 the approximation error, but the error is guaranteed to be lower than the provided
 52 bound. Moreover, the error bound computation is included in the metamodel, so that
 53 its computational burden stays small compared to the full model. For example, we can
 54 cite [12] where the authors provide such bounds in the framework of the reduced basis
 55 method (dimension reduction). Providing such error bound for nonlinear problems is
 56 the aim of this paper. We will clarify below precisely how we aim to do this and what
 57 differentiates us from current approaches.

58 In the following, we are considering, for a given parameter μ in a parameter
 59 space \mathcal{P} , the solution $u(\mu) \in X$ of an equation of the form $\mathcal{M}(\mu, u(\mu)) = 0$, with
 60 $\mathcal{M} : \mathcal{P} \times X \rightarrow Y$, and X, Y two finite dimensional vector spaces to be specified
 61 further in Section 2. In many application cases, however, one is not interested in the
 62 solution $u(\mu)$ by itself, but rather in a quantity of interest, or model output, which is
 63 a functional of this solution. Taking this functional into account when performing the
 64 model reduction leads to a so-called goal-oriented method. For instance, goal-oriented
 65 basis choice procedures have been successfully introduced in the context of dynamical
 66 systems in [20, 3], where the basis is chosen so as to contain the modes that are
 67 relevant to accurately represent the output of interest, and in a general context in [1],
 68 where the basis is chosen so as to minimize the overall output error. All those papers
 69 showed that using an adapted basis could lead to a great improvement of reduction
 70 error. In [12], the authors consider, in the context of reduced basis, goal-oriented
 71 error estimation, that is, the description of a rigorous and computable error bound
 72 between the model output and the reduced one. In [5], the authors outperform the
 73 accuracy of the bound in [12] by accepting a small risk $\alpha \in (0, 1)$ of this bound to be
 74 violated. They provide a so-called probabilistic error bound.

75 In the present paper, we extend the results in [5] by providing a probabilistic
 76 goal-oriented error estimation procedure for nonlinear problems $\mathcal{M}(\mu, u(\mu)) = 0$, and
 77 for very general metamodeling approaches. The main point for this generalization is
 78 the notion of finite difference adjoint of an operator introduced in Proposition 1 of
 79 Section 3.1.

80 The paper is organized as follows: in Section 2, we precise the objectives of our
 81 study, that is the derivation of an offline/online probabilistic goal-oriented error esti-
 82 mation procedure in a nonlinear context. In Section 3, we describe the different steps
 83 of the procedure. More precisely, we introduce in Section 3.1, the notion of finite
 84 difference adjoint of an operator, before extending in Section 3.2 the procedure in [5]
 85 to nonlinear models and linear outputs. In Section 3.3, we prove that the results in
 86 Section 3.2 can be extended to nonlinear models and nonlinear outputs. Section 3.4
 87 provides the different steps for a practical efficient evaluation of the error bound. Some
 88 numerical experiments are given in Section 4 where first a linear transport is consid-
 89 ered and then the nonlinear Burgers partial differential equation. Section 5 contains
 90 some concluding remarks and Appendix A collects the proof of some intermediate
 91 results.

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

$$98 \quad (1) \quad \mathcal{M}(\mu, u(\mu)) = 0,$$

99 and we define the *output* by

$$100 \quad (2) \quad s(\mu) = \langle \ell, u(\mu) \rangle_X,$$

101 for a given $\ell \in X$.

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

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

$$109 \quad \tilde{s}(\mu) = \langle \ell, \tilde{u}(\mu) \rangle_X.$$

110 The objective is then to provide some probabilistic error bound between $s(\mu)$ and
 111 $\tilde{s}(\mu)$. In other words, one accepts the risk of this bound $\epsilon(\mu; \alpha)$ being violated for a
 112 set of parameters having "small" probability measure $\alpha \in (0, 1)$:

$$113 \quad P(|s(\mu) - \tilde{s}(\mu)| \geq \epsilon(\mu; \alpha)) \leq \alpha.$$

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

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

- 117 • an *offline* phase, dedicated to the construction of the reduced model \tilde{u} , during
 118 which one has to solve the full dimensional problem (1) only for a reasonably
 119 small number of parameters μ_1, \dots, μ_κ ;
- 120 • an *online* phase, during which we evaluate the approximate output $\tilde{s}(\cdot) =$
 121 $\langle \ell, \tilde{u}(\cdot) \rangle$ for all queried μ .

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

127 In this article, we will not focus on the ways of constructing efficient offline-online
 128 approximation procedures for $u(\mu)$, as in *e.g.*, [12], [4], [16], [7]. Assumptions on the
 129 approximation procedure in use are very mild (see Section 3.4 and more specifically
 130 Lemma 6). Under these mild assumptions, we propose hereafter a new procedure to
 131 compute efficiently, using an online / offline decomposition, a goal-oriented proba-
 132 bilistic error bound $\epsilon(\mu; \alpha)$ which generalizes the error bound described in [5] (see also
 133 [6] for further results in control theory).

134 **3. Probabilistic nonlinear error bound.** In this section, we aim at providing
 135 a goal-oriented probabilistic error bound on the output. In [5], the authors propose
 136 such an error bound in the linear context, that is assuming that for any $\mu \in \mathcal{P}$, the
 137 operator $\mathcal{M}(\mu, \cdot) : X \rightarrow Y$ is affine (linear operator + a constant), and that the
 138 output is also linear. In the sequel we will call *linear* this case, as opposed to the
 139 nonlinear case where the model is not affine.

140 By accepting a small risk $\alpha \in (0, 1)$ that this bound could be violated, the authors
 141 avoid the use of (often pessimistic) Lipschitz bounds. In this section, we extend the
 142 results in [5] to the nonlinear context: for any $\mu \in \mathcal{P}$, the operator $\mathcal{M}(\mu, \cdot) : X \rightarrow Y$
 143 is not necessarily affine. In Section 3.2, the output is assumed to be linear, then in
 144 Section 3.3, the output may be nonlinear.

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

$$146 \quad (3) \quad r(\mu) = \mathcal{M}(\mu, \tilde{u}(\mu)) - \mathcal{M}(\mu, u(\mu)), \mu \in \mathcal{P}.$$

147 In the sequel we explain why we need to define a new adjoint. To do so we recall
 148 the computations of the linear case, in order to draw the parallel with the nonlinear
 149 case and motivate the need for a new adjoint definition.

150 In the linear case, if the model $\mathcal{M}(\mu, \cdot)$ is affine, let $A(\mu)$ be the matrix representation
 151 of $\mathcal{M}(\mu, \cdot)$ with respect to the canonical basis of X : $\mathcal{M}(\mu, u) = A(\mu)u + b$ where $b \in Y$
 152 is a given vector. We assume that for any $\mu \in \mathcal{P}$, $A(\mu)$ is invertible. In that case, the
 153 dimensions of X and Y are equal, i.e., $\mathcal{N} = \mathcal{S}$. For any matrix A let A^\top denote the
 154 transpose of A . We can define $w(\mu) \in Y$ as the solution of the so-called dual problem:

$$155 \quad (4) \quad \mathcal{M}^*(\mu, w(\mu)) = A^\top(\mu)w(\mu) = \ell$$

156 where $\ell \in X$ is the one used in the definition of the linear output in (2), and with
 157 $\mathcal{M}^*(\mu, \cdot)$ the linear adjoint of $\mathcal{M}(\mu, \cdot)$. Let $\Phi = \{\phi_1, \dots, \phi_{\mathcal{N}}\}$ denote any orthonormal
 158 basis of Y . We then have

$$159 \quad (5) \quad \begin{aligned} \tilde{s}(\mu) - s(\mu) &= \langle \ell, \tilde{u}(\mu) - u(\mu) \rangle = \langle A^\top(\mu)w(\mu), \tilde{u}(\mu) - u(\mu) \rangle \\ &= \langle w(\mu), A(\mu)\tilde{u}(\mu) - A(\mu)u(\mu) \rangle = \langle w(\mu), r(\mu) \rangle \\ &= \sum_{i=1}^{\mathcal{N}} \langle w(\mu), \phi_i \rangle \langle r(\mu), \phi_i \rangle. \end{aligned}$$

160 In order to adapt this procedure to the nonlinear context, we need to define a gener-
 161 alization of the adjoint of $\mathcal{M}^* : \mathcal{P} \times X \times X \times Y \rightarrow X$ that still allows (5) with $w(\mu)$
 162 defined by

$$163 \quad (6) \quad \mathcal{M}^*(\mu, \tilde{u}(\mu), u(\mu), w(\mu)) = \ell$$

164 which generalizes (4). It is the purpose of Section 3.1 below.

3.1. Finite difference adjoint of an operator. To generalize (5) for nonlinear
 problem, one wants to define an operator

$$\mathcal{M}^* : \mathcal{P} \times X \times X \times Y \rightarrow X,$$

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

$$\forall \mu \in \mathcal{P}, \forall x_1, x_2 \in X, \forall y \in Y,$$

165

$$166 \quad (7) \quad \langle x_1 - x_2, \mathcal{M}^*(\mu, x_1, x_2, y) \rangle = \langle \mathcal{M}(\mu, x_1) - \mathcal{M}(\mu, x_2), y \rangle.$$

167 Let us underline that previous definitions of nonlinear adjoint do not readily allow for
168 this property, such as, e.g., the one offered by Definition 2.1 in [15]:

$$169 \quad \forall \mu \in \mathcal{P}, \forall x \in X, \forall y \in Y, \langle x, \mathcal{M}^*(\mu, x, y) \rangle = \langle \mathcal{M}(\mu, x), y \rangle.$$

170 In our case the dependance in both x_1, x_2 is crucial, and missing in previous defi-
171 nitions. In Proposition 1 below, we propose a new definition for the adjoint \mathcal{M}^* :
172 $\mathcal{P} \times X \times X \times Y \rightarrow X$ and state its main properties.

173 **PROPOSITION 1** (Finite difference adjoint). *Assume that the operator $\mathcal{M} : \mathcal{P} \times$
174 $X \rightarrow Y$ is continuously differentiable with respect to x for all $x \in X$. Let $d\mathcal{M}(\mu, x) :$
175 $X \rightarrow Y$ denote the derivative of \mathcal{M} with respect to $x \in X$. Let $d\mathcal{M}^*(\mu, x) : Y \rightarrow X$
176 denote the (linear) adjoint of $d\mathcal{M}(\mu, x)$. We now define the finite difference adjoint
177 operator of \mathcal{M} by*

$$178 \quad (8) \quad \mathcal{M}^*(\mu, x_1, x_2, y) = \int_0^1 d\mathcal{M}^*(\mu, x_2 + s(x_1 - x_2))(y) ds$$

179 for all $(\mu, x_1, x_2, y) \in \mathcal{P} \times X \times X \times Y$.

180 We then have the following properties:

- 181 1. Assume that $\mathcal{M}(\mu, \cdot)$ is linear, and let $A(\mu)$ denote its matrix representation
182 with respect to the canonical basis of X , i.e.

$$183 \quad \forall \mu \in \mathcal{P}, \quad \mathcal{M}(\mu, x) = A(\mu)x$$

184 then

$$185 \quad \forall \mu \in \mathcal{P}, \forall x_1, x_2 \in X, \forall y \in Y, \quad \mathcal{M}^*(\mu, x_1, x_2, y) = A(\mu)^T y.$$

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

- 187 3. Identity (7) is satisfied by \mathcal{M}^* .

188 *Proof of Proposition 1* The proof is postponed to the appendix. \square

189 **LEMMA 2.** *Let us now consider the adjoint problem described by (6):*

$$190 \quad \text{Find } w(\mu) \text{ solution of } \mathcal{M}^*(\mu, \tilde{u}(\mu), u(\mu), w(\mu)) = \ell.$$

191 *This problem is always linear. Let us assume that, for all $\mu \in \mathcal{P}$, it admits a solution.*

192 *Then equality (5) still holds true for all linear outputs: $s(\mu) = \langle \ell, u(\mu) \rangle$ and $\tilde{s}(\mu) =$
193 $\langle \ell, \tilde{u}(\mu) \rangle$, where $r(\mu)$ is defined in (3), and $\{\phi_1, \dots, \phi_N\}$ denotes any orthonormal
194 basis of Y .*

195 *Proof of Lemma 2* Item 2 in Proposition 1 claims that \mathcal{M}^* is linear in its fourth
196 argument, thus the adjoint problem described in (6) is linear. We assume that for all
197 $\mu \in \mathcal{P}$ it admits a solution $w(\mu)$.

198 Following the beginning of the proof of Theorem 1.1 in [5], we expand the residual
199 in the basis Φ :

$$200 \quad (9) \quad r(\mu) = \sum_{i=1}^N \langle r(\mu), \phi_i \rangle \phi_i.$$

Then:

$$\tilde{s}(\mu) - s(\mu) = \langle \ell, \tilde{u}(\mu) - u(\mu) \rangle.$$

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

$$\tilde{s}(\mu) - s(\mu) = \langle \mathcal{M}^*(\mu, \tilde{u}(\mu), u(\mu), w(\mu)), \tilde{u}(\mu) - u(\mu) \rangle.$$

Then, applying Identity (7) we obtain:

$$\tilde{s}(\mu) - s(\mu) = \langle \mathcal{M}(\mu, \tilde{u}(\mu)) - \mathcal{M}(\mu, u(\mu)), w(\mu) \rangle = \langle r(\mu), w(\mu) \rangle.$$

At last, considering the expansion (9), and as the basis Φ is orthonormal, we get:

$$\tilde{s}(\mu) - s(\mu) = \sum_{i=1}^{\mathcal{N}} \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle. \quad \square$$

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 $\Phi = \{\phi_1, \dots, \phi_{\mathcal{N}}\}$ denotes any orthonormal basis of Y . Let $K \leq \mathcal{N}$ be a ‘‘truncation index’’. For any $i \in \{1, \dots, K\}$, we define:

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

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

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

Our aim is to propose a probabilistic upper bound for $|s(\tilde{u}(\mu)) - s(u(\mu))|$. For this, let us consider the right-hand term in (5): $\sum_{i=1}^{\mathcal{N}} \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle$. In order to bound this term, up to the truncation argument K , it seems natural to define, for any $\mu \in \mathcal{P}$, and for any $1 \leq i \leq K$:

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

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

As we want a bound for $\left| \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle \right|$, we finally define:

$$T_1^{\text{up}}(\mu, K, \Phi) = \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \beta_i^{\text{up}}(\mu, \Phi), \quad T_1^{\text{low}}(\mu, K, \Phi) = \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \beta_i^{\text{low}}(\mu, \Phi),$$

and

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

To deal with the terms above the truncation argument, we define:

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

Our main result is then:

224 THEOREM 3. Let $\alpha \in (0, 1)$. We have

$$225 \quad P(|s(\mu) - \tilde{s}(\mu)| \geq \epsilon(\mu; \alpha)) \leq \alpha$$

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

$$227 \quad \epsilon(\mu; \alpha) = T_1(\mu, K, \Phi) + \frac{T_2(K, \Phi)}{\alpha}.$$

228 REMARK 1. The result of Theorem 3 is a generalization of Theorem 1.1 in [5] to
229 nonlinear operators \mathcal{M} .

230 REMARK 2. **Choice of the basis Φ .**

231 The result of Theorem 3 is true for any orthonormal basis Φ of Y . For efficiency
232 reasons, we would like to choose Φ so that the parameter-independent part $T_2(K, \Phi)$
233 is the smallest possible, for a fixed truncation index $K \in \mathbb{N}^*$.

234 To our knowledge, minimizing $T_2(K, \Phi)$ over orthonormal bases of Y is an opti-
235 mization problem for which no efficient algorithm exists. However, we can minimize
236 an upper bound of $T_2(K, \Phi)$.

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

$$238 \quad (10) \quad \forall \varphi \in Y, \quad G\varphi = \frac{1}{2} \mathbf{E}_\mu (\langle r(\mu), \varphi \rangle r(\mu) + \langle w(\mu), \varphi \rangle w(\mu)).$$

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

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

239 We can state that:

240 LEMMA 4 (Theorem 1.2. in [5]). It holds

$$241 \quad T_2(K, \Phi^G) \leq \sum_{K+1}^{\mathcal{N}} \lambda_i^2.$$

242 This lemma explains the heuristic choice of $\Phi = \Phi^G$. Indeed, if G is smooth
243 enough, its eigenvalues will decrease quickly and $T_2(K, \Phi^G)$ should be small.

244 We are now in position to prove our main result.

245 *Proof of Theorem 3* We start from the result of Lemma 2:

$$246 \quad \tilde{s}(\mu) - s(\mu) = \sum_{i=1}^{\mathcal{N}} \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle.$$

Then, we can argue as in the proof of Theorem 1.1 in [5]. By construction of
 $T_1(\mu, K, \Phi)$ one gets:

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

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

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

$$\begin{aligned} &\leq P \left(|\tilde{s}(\mu) - s(\mu)| > \left| \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle \right| + \frac{T_2(K, \Phi)}{\alpha} \right) \\ &\leq P \left(\left| \sum_{i=K+1}^{\mathcal{N}} \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle \right| > \frac{T_2(K, \Phi)}{\alpha} \right). \end{aligned}$$

where in the last inequality, Lemma 2 has been used. Then, by Markov Inequality, using $\alpha \in (0, 1)$, and by definition of $T_2(\mu, K, \Phi)$ we get:

$$\begin{aligned} &P \left(\left| \sum_{i=K+1}^{\mathcal{N}} \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle \right| > \frac{T_2(\Phi)}{\alpha} \right) \\ &\leq \frac{\mathbf{E}_\mu \left(\left| \sum_{i=K+1}^{\mathcal{N}} \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle \right| \right)}{\frac{T_2(K, \Phi)}{\alpha}} = \alpha. \quad \square \end{aligned}$$

247 **3.3. Corollary: error bound for a nonlinear output.** In this section we
248 provide an extension of Theorem 3 to the context of a nonlinear output $S(\mu)$. To do
249 so we consider the following problem:

PROBLEM 1.

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

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

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

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

252

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

254 LEMMA 5. Problem 1 can be written in the framework of a non necessarily linear
255 model $\mathcal{M} : \mathcal{P} \times (X \times \mathbb{R}) \rightarrow Y$ and of a linear output $s(\mu) = \langle \ell, u(\mu) \rangle$ with $\ell \in X \times \mathbb{R}$.

256 Proof of Lemma 5 The idea consists in augmenting the state vector $v(\mu)$ with the
257 output $S(\mu)$:

$$258 \quad u(\mu) = \begin{pmatrix} v(\mu) \\ S(\mu) \end{pmatrix} = \begin{pmatrix} \bar{u}(\mu) \\ \underline{u}(\mu) \end{pmatrix} \in X \times \mathbb{R}$$

259 where $\bar{u}(\mu) \in X$ denotes the first component of $u(\mu)$ (corresponding to $v(\mu)$) and
260 $\underline{u}(\mu) \in \mathbb{R}$ its last component (corresponding to $S(\mu)$). We then define $\mathcal{M} : \mathcal{P} \times (X \times$
261 $\mathbb{R}) \rightarrow Y$ by:

$$262 \quad \mathcal{M}(\mu, u(\mu)) = \begin{pmatrix} \mathcal{H}(\mu, \bar{u}(\mu)) \\ f(\bar{u}(\mu)) - \underline{u}(\mu) \end{pmatrix},$$

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 1 is then equivalent to:

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

263 This concludes the proof of Lemma 5. \square

264 By combining Lemma 5 with Theorem 3, we get an error bound in the context of
265 a nonlinear output $S(\mu)$. This gives a solution to Problem 1.

266

267 **Computation of the finite difference adjoint of \mathcal{M} .**

268 Except in some particular cases there exists no explicit formulation of the adjoint
269 of \mathcal{M} in the context of Proposition 1. To illustrate this purpose, let us consider the
270 case where \mathcal{H} is linear (with respect to the second argument), with $B(\mu)$ denoting its
271 matrix representation with respect to the canonical basis of X . For sake of simplicity,
272 let us fix $X = \mathbb{R}^{\mathcal{N}}$. Even in that case, as the output is nonlinear, the operator \mathcal{M}
273 is also nonlinear. We want to provide an explicit formulation for the adjoint of the
274 operator \mathcal{M} , starting from (8). We first consider $d\mathcal{M}(\mu, \cdot)$. For $v \in \mathbb{R}^{\mathcal{N}+1}$, recall that:

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

275 which leads immediately to:

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

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

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

276 where the top left block has size $\mathcal{N} \times \mathcal{N}$, the top right block $\mathcal{N} \times 1$, the bottom left
277 $1 \times \mathcal{N}$ (as $f : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}$) and the bottom right lives in \mathbb{R} . Then we have, for $x, x' \in \mathbb{R}^{\mathcal{N}}$:

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

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

281 Below we provide examples for which an explicit formulation for the integral

282 $\int_0^1 df(x' + \alpha(x - x')) d\alpha$ is available.

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

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

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

285 EXAMPLE 2 (Special cases $\int f$ explicit). *In some cases the above integral can*
 286 *also be explicitly computed. We give a few nonlinear examples below.*

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

$$\begin{aligned} \int_0^1 df^T(x' + \alpha(x - x')) d\alpha &= \int_0^1 (f'_1(x'_1 + \alpha(x_1 - x'_1)), \dots) d\alpha \\ &= \left(\frac{f_1(x_1) - f_1(x'_1)}{x_1 - x'_1}, \dots, \frac{f_{\mathcal{N}}(x_{\mathcal{N}}) - f_{\mathcal{N}}(x'_{\mathcal{N}})}{x_{\mathcal{N}} - x'_{\mathcal{N}}} \right) \end{aligned}$$

287 *For example:*

(a) $f : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}, x \mapsto f(x) = \sum_{i=1}^{\mathcal{N}} x_i^2$

$$\int_0^1 df^T(x' + \alpha(x - x')) d\alpha = (x_1 + x'_1, x_2 + x'_2, \dots, x_{\mathcal{N}} + x'_{\mathcal{N}})$$

(b) $f : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}, x \mapsto f(x) = \sum_{i=1}^{\mathcal{N}} e^{x_i}$

$$\int_0^1 df^T(x' + \alpha(x - x')) d\alpha = \left(\frac{e^{x_1} - e^{x'_1}}{x_1 - x'_1}, \dots, \frac{e^{x_{\mathcal{N}}} - e^{x'_{\mathcal{N}}}}{x_{\mathcal{N}} - x'_{\mathcal{N}}} \right)$$

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

$$\begin{aligned} &\int_0^1 df^T(x' + \alpha(x - x')) d\alpha \\ &= \int_0^1 \frac{1}{\left(\sum_{i=1}^{\mathcal{N}} (x'_i + \alpha(x_i - x'_i))^2 \right)^{1/2}} (x'_1 + \alpha(x_1 - x'_1), \dots) d\alpha \end{aligned}$$

288 *which can therefore be explicitly computed as a function of x and x' coordi-*
 289 *nates:*

291
$$\left(x'_i \gamma a^{-1/2} - x \sqrt{ca}^{-1} + \frac{1}{2} \gamma a^{-3/2} + x'_i \sqrt{ca}^{-1} - \frac{1}{2} x'_i b \gamma a^{-3/2} + x'_i \delta a^{-1/2} \right.$$

 292
$$\left. + x_i \sqrt{a+b+ca}^{-1} - \frac{1}{2} x_i b \delta a^{-3/2} - x'_i \sqrt{a+b+ca}^{-1} + \frac{1}{2} x'_i b \delta a^{-3/2} \right)_{i=1, \dots, \mathcal{N}}$$

294 *where:*

295
$$a = \sum_{i=1}^{\mathcal{N}} (x_i - x'_i)^2, \quad b = 2 \sum_{i=1}^{\mathcal{N}} x_i (x_i - x'_i), \quad c = \sum_{i=1}^{\mathcal{N}} x_i^2,$$

296

297

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

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:

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

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:

$$\left\{ \begin{array}{l} B^\top(\mu)\bar{w} + \int_0^1 df^\top(u + s(\tilde{u} - u))\underline{w} ds = 0 \\ -\underline{w} = 1 \end{array} \right\} \begin{array}{l} \mathcal{N} \text{ equations} \\ 1 \text{ equation} \end{array} ,$$

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

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

3.4. Efficient bound evaluation in a many-query or real-time context.

In practice, the error bound $\epsilon(\mu; \alpha)$ used in Theorem 3 can not be directly evaluated, and one has to define a computable approximation $\hat{\epsilon}(\mu; \alpha)$. Our approximation is justified and commented in [5] Section 1.3, and we recall it here for sake of self-containedness. We end this section with Lemma 6, which gives sufficient conditions to ensure efficient computation of our online error bound.

Estimation of Φ^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 $G : Y \rightarrow Y$ by a linear operator $\hat{G} : Y \rightarrow Y$ defined as:

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

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 [5, Section 1.3] for more details).

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

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

with

$$\left\{ \begin{array}{l} T_1^{up}(\mu, K, \Phi) = \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \beta_i^{up}(\mu, \Phi), \\ T_1^{low}(\mu, K, \Phi) = \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \beta_i^{low}(\mu, \Phi). \end{array} \right.$$

The $\beta(\mu, \Phi)$ constants can be approximated using a simple discrete minimization (ie., replacing \mathcal{P} by a discrete sample Ξ in the minimum/maximum defining $\beta^{max}(\Phi)$ and $\beta^{min}(\Phi)$). In some cases, one can use a continuous optimization method to solve

316 these minimum/maximum problems. It is clear that all these computations can be
317 done during the offline phase.

318 We now discuss the computation of the K scalar products $\langle r(\mu), \phi_i \rangle$ ($i = 1, \dots, K$)
319 with an offline/online procedure.

320 LEMMA 6. *Let $\{y_1, \dots, y_S\}$ denote an orthonormal basis of Y and $\{x_1, \dots, x_N\}$
321 denote an orthonormal basis of X . Assume that $\mathcal{M} : \mathcal{P} \times X \rightarrow Y$ is defined by:*

$$322 \quad (11) \quad \mathcal{M} \left(\mu, \sum_{i=1}^{\mathcal{N}} v_i x_i \right) = \sum_{j=1}^{\mathcal{S}} m_j(\mu, v_1, \dots, v_{\mathcal{N}}) y_j.$$

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

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

$$325 \quad (12) \quad m_j(\mu, v_1, \dots, v_{\mathcal{N}}) = \sum_{k=0}^{T_j} Q_{k,j}(v_1, \dots, v_{\mathcal{N}}) h_k(\mu)$$

with

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

326 and

$$327 \quad (13) \quad Q_{k,j}(v_1, \dots, v_{\mathcal{N}}) = \sum_{\alpha=(\alpha_1, \dots, \alpha_{\mathcal{N}}) \in I_{j,k}} q_{j,k,\alpha} \prod_{l \in V_{\alpha}} v_l^{\alpha_l}$$

328 where:

$$329 \quad (14) \quad \begin{aligned} I_{j,k} &\subset \mathbb{N}^{\mathcal{N}}, \quad I = \bigcup_{j=1}^{\mathcal{N}} \bigcup_{k=1}^{T_j} I_{j,k}, \quad \#I = M \\ V_{\alpha} &\subset \{1, \dots, \mathcal{N}\}, \quad \max_{\alpha \in I} \# \{V_{\alpha}\} = L \end{aligned}$$

We set

$$T = \max_{j=1, \dots, \mathcal{S}} T_j$$

Recall that $N \ll \mathcal{N}$ is the dimension of \tilde{X} . Let

$$R = \max_{\alpha \in I} \max_{l \in V_{\alpha}} \alpha_l \binom{\alpha_l + N - 1}{N - 1}$$

Then, it is possible to compute all the scalar products $\langle r(\mu), \phi_i \rangle$ ($i = 1, \dots, K$) with an
offline/online procedure whose online phase has a cost of the size

$$\mathcal{O}(T \times M \times L \times R)$$

REMARK 3. *The decomposition*

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

with

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

330 plays an analogous role to the ‘‘affine parameter dependence’’ that is commonly as-
331 sumed in the linear litterature (see, e.g., [12], page 1526).

332 **REMARK 4.** *Let us emphasize that, in the previous result, the cost does not depend*
 333 *on the high dimension \mathcal{N} . Therefore if we assume that T, M, L and $R \ll \mathcal{N}$, then*
 334 *it is possible to compute the K scalar products, with an offline/online procedure with*
 335 *a small cost (with respect to \mathcal{N}).*

336 **REMARK 5.** *Note that it is possible to work with the K scalar products themselves,*
 337 *without any approximation, especially in the case where the polynomial decomposition*
 338 *presented above is not valid. In that case, the cost of the online phase is $\mathcal{O}(\mathcal{N})$, which*
 339 *is still better than the full problem, whose complexity is $\mathcal{O}(\mathcal{N}^\alpha)$ with $\alpha \geq 2$ in most*
 340 *cases.*

341 *In the polynomial case, Lemma 6 above allows to reduce the cost of the online phase*
 342 *to a cost which does not depend on the high dimension \mathcal{N} anymore.*

343 *Proof of Lemma 6* The proof is postponed to Appendix A.2. \square

344 **Approximation of $T_2(K, \Phi)$.** A Monte-Carlo estimator of $T_2(K, \Phi)$ is used:

$$345 \quad \hat{T}_2(K, \Phi) = \frac{1}{2\#\Xi} \sum_{\mu \in \Xi} \left| \tilde{s}(\mu) - s(\mu) - \sum_{i=1}^K \langle r(\mu), \phi_i \rangle \langle w(\mu), \phi_i \rangle \right|,$$

346 where Ξ is a sample of \mathcal{P} .

347 As this quantity is μ -independent, it can be computed for once during the offline
 348 phase. The error analysis, which is related to the central limit theorem, is discussed
 349 in [5, Section A].

350 **Computable error bound** We now rely on Proposition 3 and set:

$$351 \quad \hat{\epsilon}(\mu; \alpha) = T_1(\mu, K, \Phi) + \frac{\hat{T}_2(K, \Phi)}{\alpha}.$$

352 It is an estimator for the error bound $\epsilon(\mu; \alpha)$ in Theorem 3.

353 4. Numerical experiments.

354 **4.1. First experiments with a toy model.** We now apply our error bound
 355 on a non-homogeneous linear transport equation with a nonlinear output. We use the
 356 results of the corollary (Section 3.3).

357 **4.1.1. Toy model.** Let $u_e = u_e(x, t)$ be the solution of the linear transport
 358 equation:

$$359 \quad \frac{\partial u_e}{\partial t}(x, t) + \mu \frac{\partial u_e}{\partial x}(x, t) = \sin(x) \exp(-x)$$

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

$$361 \quad u_e(x, t = 0) = x(1 - x) \quad \forall x \in [0, 1],$$

362 and boundary condition:

$$363 \quad u_e(x = 0, t) = 0 \quad \forall t \in [0, 1].$$

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

365 We choose a number of timesteps N_t and a number of space points N_x , we set $\Delta_t =$
 366 $1/N_t$ and $\Delta_x = 1/N_x$ and we introduce our discrete unknown $u = (u_i^n)_{i=0, \dots, N_x; n=0, \dots, N_t}$.
 367 We note here that the considered PDE is an hyperbolic evolution equation, and
 368 that we perform the reduction on the space-time unknown u , of dimension $\mathcal{N} =$

369 $(N_x + 1) \cdot (N_t + 1)$. This is different from reducing the space-discretized equation at
 370 each time step.

371 The u vector satisfies the discretized initial-boundary conditions:

$$372 \quad (15) \quad \forall i, \quad u_i^0 = i\Delta_x(1 - i\Delta_x)$$

373

$$374 \quad (16) \quad \forall n, \quad u_0^n = 0$$

375 and the first-order upwind scheme implicit relation:

$$376 \quad (17) \quad \forall i, n \quad \frac{u_{i+1}^{n+1} - u_{i+1}^n}{\Delta_t} + \mu \frac{u_{i+1}^{n+1} - u_i^{n+1}}{\Delta_x} = \sin(i\Delta_x) \exp(-i\Delta_x).$$

377 Let $B(\mu)$ (resp. ϕ) be the matrix (resp. the vector) so that (15),(16) and (17) are
 378 equivalent to:

$$379 \quad (18) \quad B(\mu)u = \phi \in \mathbb{R}^N \quad \text{with } N = N_x \times N_t.$$

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

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

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

385 **4.1.2. Reduction.** The approximation \tilde{u} of u is computed by using a “reduced
 386 basis” approach [12]. To be more specific, \tilde{u} is the solution of:

$$387 \quad Z^t B(\mu) Z \tilde{u} = Z^t \phi,$$

388 where Z is an appropriate matrix found by Proper Orthogonal Decomposition (POD)
 389 (see [17] for instance). The Z matrix is the matrix of an orthogonal set of n vectors
 390 in $X = \mathbb{R}^N$, endowed with the Euclidian scalar product. The n number is called the
 391 *reduced basis size*. The larger n is, the more precise the approximation $\tilde{u} \approx u$ is, but
 392 also the the more expensive the computation of \tilde{u} is, so that a compromise must be
 393 found.

394 The Z matrix is computed using a POD snapshot of size 70, and $N = 20$ retained
 395 $\hat{\phi}_i^G$ vectors. We took a very low risk level $\alpha = 0.0001$.

396 **4.1.3. Results.** In Figure 1, we plotted, as functions of the reduced basis size,
 397 the true error and the error bound means on a sample of 200 random parameter values,
 398 for the three different output cases (square, exponential and triple exponential).

399 The graphs show that our error bound remains accurate and sharp with respect
 400 to the true error, despite the highly-nonlinear output functions that have been chosen
 401 (yet, it seems almost unaffected by the degree of nonlinearity in the output).

402 **4.2. Burgers experiment.** In this section, we are interested in the discretized
 403 Burgers’ equation, as an example of nonlinear model.

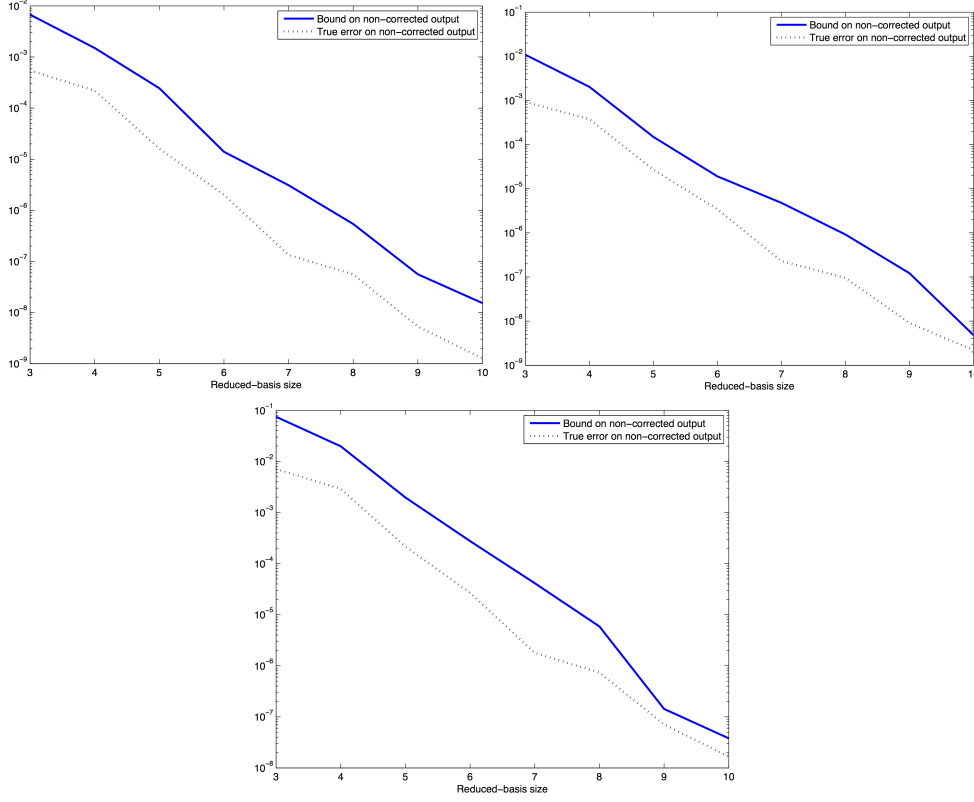


FIG. 1. Comparison between the mean error bound and the true error, for different reduced basis sizes, in the square (top left), the exponential (top right) and the triple exponential (down) output case.

404 **4.2.1. Description of the model and output of interest.** We are looking
 405 for $u = u(t, x)$ satisfying:

$$406 \quad \begin{cases} \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial(u^2)}{\partial x} = 0 \\ u(t, x = 0) = 1 \quad \forall t \\ u(t = 0, x) = \cos^2(\alpha x) + \beta x \end{cases}$$

407 where the parameter vector $\mu = (\alpha, \beta)$ belongs to $[0, 1] \times [0, 1]$.

408 We discretize the above equation by using an upwind scheme. We choose a number
 409 of timesteps N_t and a number of space points N_x , and we set $\Delta_t = 1/N_t$ and $\Delta_x =$
 410 $1/N_x$, 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:

$$411 \quad \begin{cases} \frac{u_i^{n+1} - u_i^n}{\Delta_t} + u_i^{n+1} \frac{u_i^{n+1} - u_{i-1}^{n+1}}{\Delta_x} = 0 \quad \forall i \geq 1 \\ u_0^n = 1 \quad \forall n \\ u_i^0 = \cos^2(\alpha i \Delta_x) + \beta i \Delta_x \quad \forall i \end{cases}$$

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

$$413 \quad \ell_i^n = \begin{cases} 1 & \text{if } I(i, n) \in \{[N_t \times N_x/3] - 1, [N_t \times N_x/5] - 1\} \\ 0 & \text{else.} \end{cases}$$

414 where $I(i, n) = n * N_x + i$, and $\lfloor y \rfloor$ denotes the floor of y .

415 **4.2.2. Reduction.** As for the toy model, the reduction is performed on the full
 416 space-time state vector $(u_i^n)_{i,n}$. We also choose a Z matrix by a POD procedure, then
 417 define the reduced state vector $(\tilde{u}_i^n)_{i,n}$ as:

$$418 \quad (\tilde{u}_i^n)_{i,n}(\mu) = \operatorname{argmin}_{v \in \operatorname{Range}(Z)} \|\mathcal{M}(\mu, v)\|^2$$

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

420 **4.2.3. Numerical experiments.** Table 1 gives the name and description of
 421 the various parameters used in the numerical code. Table 2 describes the various
 422 experiments that have been performed and links to the associated figures.

Parameter	Description	Usual range
N_x	Number of space discretization points	10 – 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) $t10 \times x10$	10	10	100	70	8	2
(b) $t20 \times x10$	20	10	100	70	8	2
(c) $t10 \times x20$	10	20	100	70	8	3
(d) $t20 \times x20$	20	20	100	70	8	3
(e) $t10 \times x40$	10	40	100	70	8	4
(f) $t20 \times x40$	20	40	200	150	12	4
(g) $t10 \times x80$	10	80	100	70	8	5
(h) $t20 \times x80$	20	80	200	150	12	5

TABLE 2
 Numerical setup of the different experiments.

423 Figures 2, 3, 4 and 5 present the true error and the error bound for a size of the
 424 POD truncated basis varying from 3 to 10, with $N_t = 10, 20$, $N_x = 10, 20, 40, 80$ and
 425 other parameters described in Table 2.

To quantify the computing gain we define and compute the following speed up ratios. The first ratio r_1 is fitted 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.

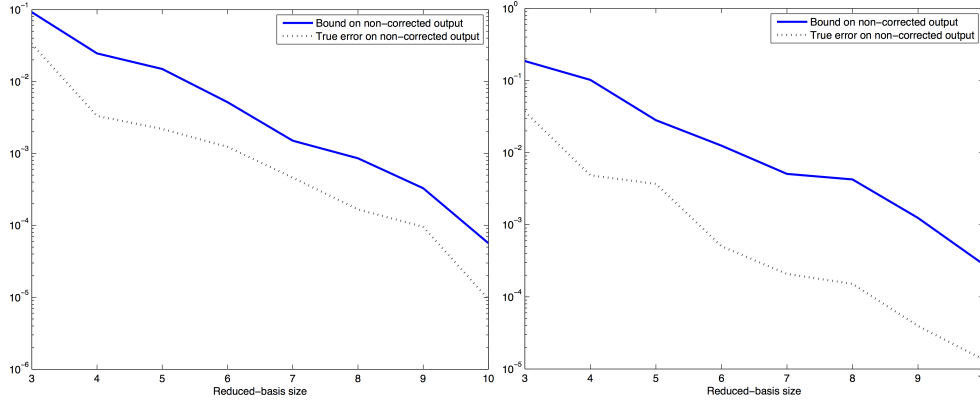


FIG. 2. True error (dashed line) and error bound (plain line) for experiments (a) $t_{10} \times x_{10}$ (left) and (b) $t_{20} \times x_{10}$ (right).

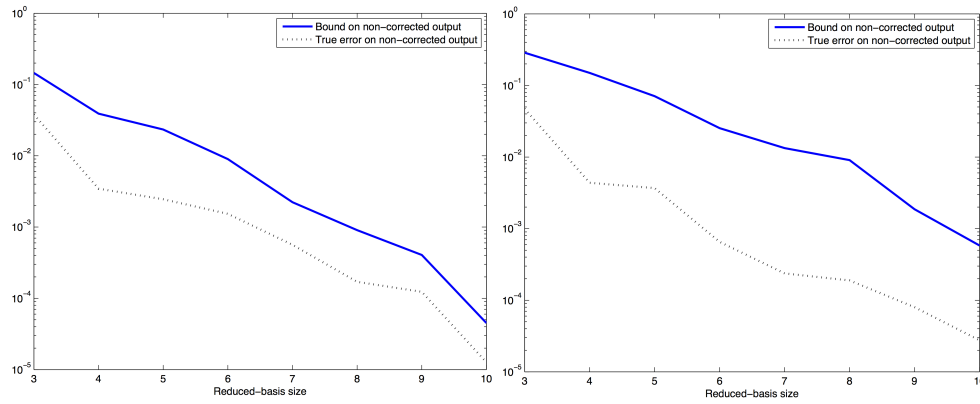


FIG. 3. True error (dashed line) and error bound (plain line) for experiments (c) $t_{10} \times x_{20}$ (left) and (d) $t_{20} \times x_{20}$ (right).

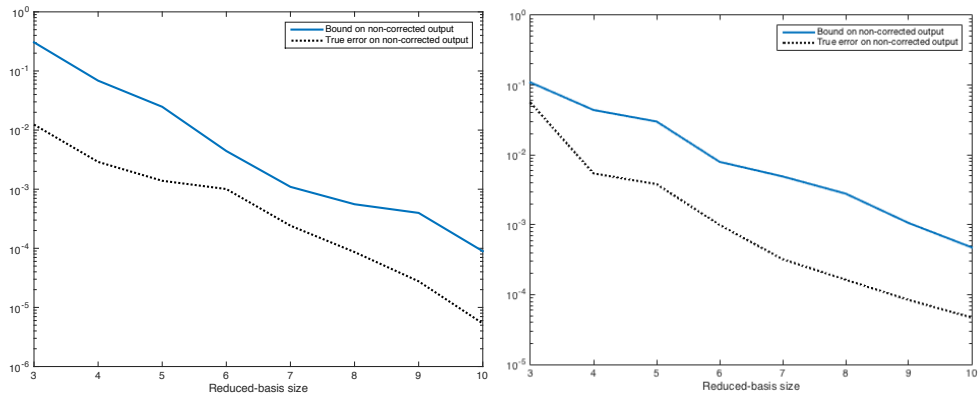


FIG. 4. True error (dashed line) and error bound (plain line) for experiments (e) $t_{10} \times x_{40}$ (left) and (f) $t_{20} \times x_{40}$ (right).

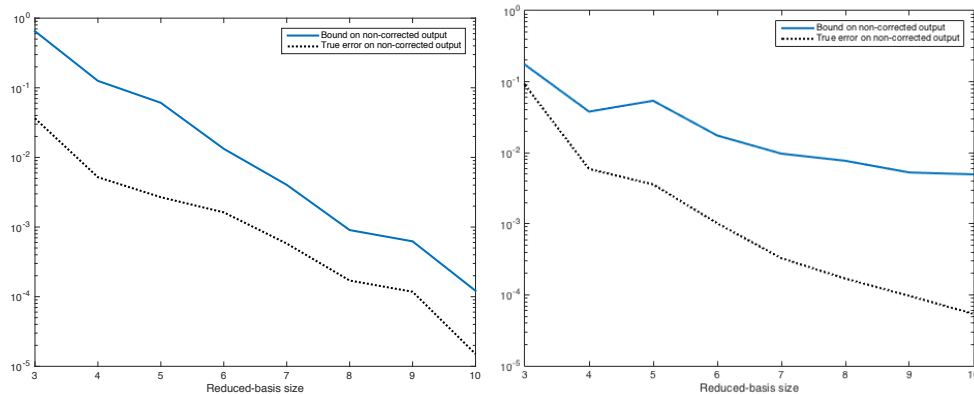


FIG. 5. True error (dashed line) and error bound (plain line) for experiments (g) $t_{10} \times x_{80}$ (left) and (h) $t_{20} \times x_{80}$ (right).

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 :

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

426 with $K = 1000$.

427 The larger the speed ratios, the more efficient the use of a reduction procedure is. In
 428 our experiments, the computing time were real elapsed times computed using Matlab
 429 tic and toc functions. We summarize in Tables 3 and 4 the full, online and offline
 430 costs, as well as the speed up ratios, for the various experiments described in Table 1.

Experiment name	(a) $t_{10} \times x_{10}$	(b) $t_{20} \times x_{10}$	(c) $t_{20} \times x_{10}$	(d) $t_{20} \times x_{20}$
full pb comp. time	9.4	17.2	16.2	40.2
offline comp. time	114.3	202.2	196.8	437.3
online comp. time	6.3	9.4	9.0	15.6
speed-up ratio r_1	1.5	1.8	1.8	2.6
speed-up ratio r_2	1.5	1.8	1.8	2.5
Figure	2	2	3	3

TABLE 3

Table of costs, for a size of the truncated POD equal to 8.

431 **5. Conclusion.** A class of nonlinear problems depending on a probabilistic vec-
 432 tor has been considered, and a numerically efficient method has been designed to
 433 compute the error estimation, when approximating the output error. This method is
 434 based on two phases. The offline phase requires to compute the solution of a high-
 435 dimensional problem, and the online phase is based on the computation of the solution
 436 of a reduced-order problem. This approach has been applied to a toy model and to

Experiment name	(e) $t_{10} \times x_{40}$	(f) $t_{20} \times x_{40}$	(g) $t_{10} \times x_{80}$	(h) $t_{20} \times x_{80}$
full pb computing time	33.4	311	174.8	1500
offline computing time	329.0	2274	1205	8789
online computing time	7.6	17.2	8.0	16.9
speed-up ratio r_1	4.4	18.1	21.9	88.8
speed-up ratio r_2	4.2	16.0	19.0	58.4
Figure	4	4	5	5

TABLE 4

Table of costs, for a size of the truncated POD equal to 8.

437 a nonlinear partial differential equation, namely the Burgers equation parametrized
 438 by two probabilistic coefficients. An application of this numerical method to other
 439 mathematical problems is under investigation, more precisely, it could be fruitful to
 440 investigate the impact of this new result in control theory (as done in [6] for a linear
 441 problem). Perspectives in environmental modelling, among other domains where the
 442 sensitivity analysis is crucial, are also worth considering.

443

REFERENCES

- 444 [1] T. BUI-THANH, K. WILLCOX, O. GHATTAS, AND B. VAN BLOEMEN WAANDERS, *Goal-oriented,*
 445 *model-constrained optimization for reduction of large-scale systems*, Journal of Computa-
 446 tional Physics, 224 (2007), pp. 880–896.
- 447 [2] R. G. GHANEM AND P. D. SPANOS, *Stochastic finite elements: a spectral approach*,
 448 Springer-Verlag, New York, 1991, doi:10.1007/978-1-4612-3094-6, [http://dx.doi.org/10.](http://dx.doi.org/10.1007/978-1-4612-3094-6)
 449 [1007/978-1-4612-3094-6](http://dx.doi.org/10.1007/978-1-4612-3094-6).
- 450 [3] M. ILAK AND C. ROWLEY, *Modeling of transitional channel flow using balanced proper orthog-*
 451 *onal decomposition*, Physics of Fluids, 20 (2008), p. 034103.
- 452 [4] A. JANON, M. NODET, AND C. PRIEUR, *Reduced-basis solutions of affinely-parametrized linear*
 453 *partial differential equations*, ESAIM: Mathematical Modelling and Numerical Analysis,
 454 47 (2013), pp. 317–348.
- 455 [5] A. JANON, M. NODET, AND C. PRIEUR, *Goal-oriented error estimation for the reduced ba-*
 456 *sis method, with application to sensitivity analysis*, Journal of Scientific Computing, (to
 457 appear).
- 458 [6] A. JANON, M. NODET, C. PRIEUR, AND C. PRIEUR, *Global sensitivity analysis for the boundary*
 459 *control of an open channel*, Mathematics of Control, Signals, and Systems, 28 (2016),
 460 pp. 1–27.
- 461 [7] J. KLEIJNEN, *Design and analysis of simulation experiments*, Springer Publishing Company,
 462 Incorporated, 2007.
- 463 [8] J. KLEIJNEN, *Simulation experiments in practice: statistical design and regression analysis*,
 464 Journal of Simulation, 2 (2008), pp. 19–27.
- 465 [9] O. P. LE MAÎTRE, O. M. KNIO, B. J. DEBUSSCHERE, H. N. NAJM, AND R. G. GHANEM,
 466 *A multigrid solver for two-dimensional stochastic diffusion equations*, Comput. Methods
 467 Appl. Mech. Engrg., 192 (2003), pp. 4723–4744, doi:10.1016/S0045-7825(03)00457-2, [http:](http://dx.doi.org/10.1016/S0045-7825(03)00457-2)
 468 [//dx.doi.org/10.1016/S0045-7825\(03\)00457-2](http://dx.doi.org/10.1016/S0045-7825(03)00457-2).
- 469 [10] Y. MADAY, A. PATERA, AND D. ROVAS, *A blackbox reduced-basis output bound method*
 470 *for noncoercive linear problems*, in Nonlinear Partial Differential Equations and their
 471 Applications Collège de France Seminar Volume XIV, D. Cioranescu and J.-L. Lions,
 472 eds., vol. 31 of Studies in Mathematics and Its Applications, Elsevier, 2002, pp. 533 –
 473 569, doi:10.1016/S0168-2024(02)80025-X, [http://www.sciencedirect.com/science/article/](http://www.sciencedirect.com/science/article/pii/S016820240280025X)
 474 [pii/S016820240280025X](http://www.sciencedirect.com/science/article/pii/S016820240280025X).
- 475 [11] B. MOORE, *Principal component analysis in linear systems: Controllability, observability, and*
 476 *model reduction*, IEEE Transactions on Automatic Control, 26 (1981), pp. 17–31.
- 477 [12] N. NGUYEN, K. VEROY, AND A. PATERA, *Certified real-time solution of parametrized partial*
 478 *differential equations*, in Handbook of Materials Modeling, Springer, 2005, pp. 1523–1558.

- 479 [13] A. NOUY, *Low-rank tensor methods for model order reduction*, arXiv preprint arXiv:1511.01555,
480 (2015).
- 481 [14] T. SANTNER, B. WILLIAMS, AND W. NOTZ, *The design and analysis of computer experiments*
482 *springer-verlag*, New York. 283pp, (2003).
- 483 [15] J. M. A. SCHERPEN AND W. S. GRAY, *Nonlinear Hilbert adjoints: properties and applications*
484 *to Hankel singular value analysis*, Nonlinear Anal., 51 (2002), pp. 883–901.
- 485 [16] M. SCHEUERER, R. SCHABACK, AND M. SCHLATHER, *Interpolation of spatial data – a stochastic*
486 *or a deterministic problem ?*, Preprint, Universität Göttingen. [http://num.math.](http://num.math.uni-goettingen.de/schaback/research/papers/loSD.pdf)
487 [uni-goettingen.de/schaback/research/papers/loSD.pdf](http://num.math.uni-goettingen.de/schaback/research/papers/loSD.pdf), (2011).
- 488 [17] L. SIROVICH, *Turbulence and the dynamics of coherent structures. part i-ii*, Quarterly of applied
489 mathematics, 45 (1987), pp. 561–590.
- 490 [18] C. SOIZE AND R. GHANEM, *Physical systems with random uncertainties: chaos representations*
491 *with arbitrary probability measure*, SIAM J. Sci. Comput., 26 (2004), pp. 395–410, doi:10.
492 [1137/S1064827503424505](http://dx.doi.org/10.1137/S1064827503424505), <http://dx.doi.org/10.1137/S1064827503424505>.
- 493 [19] S. VOLKWEIN, *Proper orthogonal decomposition and singular value decomposition*. Spezial-
494 forschungsbereich F003 Optimierung und Kontrolle, Projektbereich Kontinuierliche Opti-
495 mierung und Kontrolle, Bericht. Nr. 153, Graz, 1999.
- 496 [20] K. WILLCOX AND J. PERAIRE, *Balanced model reduction via the proper orthogonal decomposi-*
497 *tion*, AIAA journal, 40 (2002), pp. 2323–2330.

498 Appendix A. Postponed proofs.

499 **A.1. Proof of Proposition 1.** We prove each item of Proposition 1 separately.

1. If $\mathcal{M}(\mu, \cdot)$ is linear, then $d\mathcal{M}(\mu, x) = A(\mu)$ for all $x \in X$ and $\mu \in \mathcal{P}$, therefore the adjoint is simply the matrix transpose: $d\mathcal{M}^*(\mu, x) = A(\mu)^T$, so that

$$\mathcal{M}^*(\mu, x, y, z) = \int_0^1 A(\mu)^T z \, ds = A(\mu)^T z$$

- 500 2. As for all $x \in X$ the adjoint operator $z \mapsto d\mathcal{M}^*(x)(z)$ is linear in z , \mathcal{M}^* is
501 clearly linear in z as well.
3. Let us prove (7). For all $\mu \in \mathcal{P}, x, y \in X, z \in Y$ we have:

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

502 This concludes the proof of Proposition 1. \square

503 **A.2. Proof of Lemma 6.** Let us recall the formula for the residual:

$$504 \quad (19) \quad r(\mu) = \mathcal{M}(\mu, \tilde{u}(\mu))$$

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

$$506 \quad (20) \quad \langle r(\mu), \phi_i \rangle = \langle \mathcal{M}(\mu, \tilde{u}(\mu)), \phi_i \rangle$$

507 Here we describe the online/offline procedure to compute

$$508 \quad (21) \quad \langle \mathcal{M}(\mu, v), \phi_i \rangle$$

509 where $v \in \tilde{X}$ and $\mu \in \mathcal{P}$ are given.

510 We also make all the assumptions of Lemma 6 regarding the decomposition of \mathcal{M} and

511 m_j s. Using the decomposition 11 we have

$$512 \quad (22) \quad \langle \mathcal{M}(\mu, v), \phi_i \rangle = \sum_{j=1}^S m_j(\mu, v) \langle y_j, \phi_i \rangle$$

We then decompose v onto a basis $\{f_1, \dots, f_N\}$ of $\tilde{X} \subset X$. First we write each f_k in the basis $\{x_1, \dots, x_N\}$ of X :

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

513 Then we write v :

$$514 \quad (23) \quad v = \sum_{k=1}^N v'_k f_k = \sum_{k=1}^N \sum_{i=1}^N f_{k,i} v'_k x_i$$

515 so that we can write:

$$516 \quad (24) \quad v = \sum_{i=1}^N v_i x_i, \quad \text{with } v_i = \sum_{k=1}^N f_{k,i} v'_k$$

517 Formula (13) requires v_l to the power α_l , so we use the multinomial formula to get:

$$518 \quad (25) \quad v_l^{\alpha_l} = \left(\sum_{k=1}^N f_{k,\alpha_l} v'_k \right)^{\alpha_l} = \sum_{\beta \in B(N, \alpha_l)} \binom{\alpha_l}{\beta} \prod_{k=1}^N (v'_k f_{k,l})^{\beta_k}$$

519 using the multinomial indices and coefficients:

$$520 \quad (26) \quad B(N, \alpha_l) = \{ \beta = (\beta_1, \dots, \beta_N) \in \mathbb{N}^N, \sum_{k=1}^N \beta_k = \alpha_l \}$$

$$\binom{\alpha_l}{\beta} = \frac{\alpha_l!}{\beta_1! \dots \beta_N!}$$

521 We replace (12) and (13) in (22):

$$522 \quad (27) \quad \langle \mathcal{M}(\mu, v), \phi_i \rangle = \sum_{j=1}^S \sum_{k=0}^{T_j} Q_{k,j}(v) h_k(\mu) \langle y_j, \phi_i \rangle$$

$$= \sum_{j=1}^S \sum_{k=0}^{T_j} \sum_{\alpha \in I_{j,k}} q_{j,k,\alpha} \left(\prod_{l \in V_\alpha} v_l^{\alpha_l} \right) h_k(\mu) \langle y_j, \phi_i \rangle$$

Now we set:

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

523 to get

$$524 \quad (28) \quad \langle \mathcal{M}(\mu, v), \phi_i \rangle = \sum_{j=1}^S \sum_{k=0}^{T_j} \sum_{\alpha \in I} q_{j,k,\alpha} \left(\prod_{l \in V_\alpha} v_l^{\alpha_l} \right) h_k(\mu) \langle y_j, \phi_i \rangle$$

$$= \sum_{k=0}^T h_k(\mu) \sum_{\alpha \in I} \left(\prod_{l \in V_\alpha} v_l^{\alpha_l} \right) \sum_{j=1}^S q_{j,k,\alpha} \langle y_j, \phi_i \rangle$$

525 During the online phase we are given μ and v . The following quantities are indepen-
526 dent of μ and v , therefore can be computed during the offline phase:

$$527 \quad (29) \quad G_{\alpha,k,i} = \sum_{j=1}^S q_{j,k,\alpha} \langle y_j, \phi_i \rangle \text{ for all } k \in \{0, \dots, T\}, i \in \{1, \dots, \mathcal{N}\}, \alpha \in V_\alpha$$

528 and the online computation then writes:

$$529 \quad (30) \quad \langle \mathcal{M}(\mu, v), \phi_i \rangle = \sum_{k=0}^T h_k(\mu) \sum_{\alpha \in I} \left(\prod_{l \in V_\alpha} v_l^{\alpha_l} \right) G_{\alpha,k,i}$$

Let us now proceed to the complexity of this computation, that is its operation count. First we consider the computation of $v_l^{\alpha_l}$, using equation (25):

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

The product $\prod_{k=1}^N (v'_k f_{k,l})^{\beta_k}$ costs (up to a multiplicative constant) $\beta_1 + \dots + \beta_N = \alpha_l$ 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

$$\#B(N, \alpha_l) = \binom{\alpha_l + N - 1}{N - 1}$$

so if we set

$$R = \max_{\alpha \in I} \max_{l \in V_\alpha} \alpha_l \binom{\alpha_l + N - 1}{N - 1}$$

then the cost of computing $v_l^{\alpha_l}$ is (up to a multiplicative constant) bounded by R . Looking back to (30) and using notations (14), the total operation count for the *online phase* is bounded by:

$$\text{const.} \times T \times M \times L \times R$$

530 This concludes the proof of Lemma 6. \square