

MOX-Report No. 24/2016

**A Reduced Basis Ensemble Kalman Filter for  
State/parameter Identification in Large-scale Nonlinear  
Dynamical Systems**

Pagani, S.; Manzoni, A.; Quarteroni, A.

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

[mox-dmat@polimi.it](mailto:mox-dmat@polimi.it)

<http://mox.polimi.it>

# A Reduced Basis Ensemble Kalman Filter for State/parameter Identification in Large-scale Nonlinear Dynamical Systems\*

Stefano Pagani<sup>†</sup>, Andrea Manzoni<sup>‡</sup>, Alfio Quarteroni<sup>†‡</sup>

June 2, 2016

## Abstract

The ensemble Kalman filter is nowadays widely employed to solve state and/or parameter identification problems recast in the framework of Bayesian inversion. Unfortunately its cost becomes prohibitive when dealing with systems described by parametrized partial differential equations, because of the cost entailed by each PDE query. This is even worse for nonlinear time-dependent PDEs. In this paper we propose a reduced basis ensemble Kalman filter technique to speed up the numerical solution of Bayesian inverse problems arising from the discretization of nonlinear time dependent PDEs. The reduction stage yields intrinsic approximation errors, whose propagation through the filtering process might affect the accuracy of the identified state/parameters. Since their evaluation is computationally heavy, we equip our reduced basis ensemble Kalman filter with a reduction error model based on ordinary kriging for functional-valued data, to gauge the effect of state reduction on the whole filtering process. The accuracy and efficiency of our method is then verified on two numerical test cases, dealing with the identification of uncertain parameters or fields for a FitzHugh-Nagumo model and a Fisher-Kolmogorov model.

## 1 Introduction

The solution of inverse and uncertainty quantification problems involving systems modeled by partial differential equations (PDEs) is computationally demanding. In this paper we develop an inversion technique that combines the reduced basis (RB) method and the ensemble Kalman filter to solve state/parameter identification problems for large-scale nonlinear dynamical systems arising from the discretization of nonlinear time-dependent PDEs. The parameter vector  $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^d$  can characterize physical or geometrical properties, or again boundary conditions and forcing terms. In this context, we are interested to identify the value of  $\boldsymbol{\mu}$  (in a sense to be made precise) which has generated a set of noisy measurements of an output  $\mathbf{s}_h(\boldsymbol{\mu}) \in \mathbb{R}^s$ , provided by a linear function of the PDE numerical solution  $\mathbf{u}_h(t; \boldsymbol{\mu})$  corresponding to  $\boldsymbol{\mu}$  at time  $t$ . The pedix  $h$  typically refers to the gridsize; for small values of  $h$ , we deal with large-scale inverse problems whose solution is computationally intensive. On the other hand, the distinction between state  $\mathbf{u}_h$  and parameter variables  $\boldsymbol{\mu}$  requires a suitable extension of the techniques usually exploited to face this kind of problems. Following the Bayesian approach, both parameters  $\boldsymbol{\mu}$  and outputs  $\mathbf{s}_h$  are modeled as random vectors. In this setting, starting from the observation of the noisy outputs  $\mathbf{s}_h$ , the inverse problem of state/parameter identification consists in finding the posterior distribution of  $[\boldsymbol{\mu}, \mathbf{u}_h]^T$  by combining the prior distribution and the likelihood of the measurements set, encoding the information on the forward problem (see e.g. [37, 38]).

---

\*This work was supported by the Italian “National Group of Computing Science” (GNCS-INDAM).

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

<sup>‡</sup>MOX, Dipartimento di Matematica, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133 Milano, Italy, [stefano.pagani@polimi.it](mailto:stefano.pagani@polimi.it)

If the forward problem is given by a dynamical system, the posterior distribution usually results from a sequential update, which combines at each step along the time interval the knowledge on the estimated parameters (until the current step), together with the most recent update provided by the solution of the forward problem through a new output evaluation. Due to the nonlinearity of the map  $\boldsymbol{\mu} \rightarrow \mathbf{s}_h(\boldsymbol{\mu})$ , the resulting posterior distribution cannot be written analytically; its computation is typically based on sampling techniques, such as Markov chain Monte Carlo (MCMC) methods [15], however, this requires an extremely large number (order of  $10^5 - 10^6$ ) of evaluations of the forward problem solution.

For time-dependent problems, alternative approaches like the Bayesian Kalman filters have been developed in order to provide less expensive approximations of the target distribution [21, 24]. The basic Kalman filter sequentially produces an estimate of the mean and covariance of the unknown quantities. In its canonical formulation, it requires the random vectors  $\boldsymbol{\mu}$  and  $\mathbf{s}_h$  to be normally distributed, and the forward map  $\boldsymbol{\mu} \rightarrow \mathbf{s}_h(\boldsymbol{\mu})$  to be linear with respect to the parameters. Since these assumptions are seldomly matched, suitable generalizations known as *approximated* Kalman Filters are exploited. In particular, we focus on the Ensemble Kalman filter (EnKF), which combines the filtering approach with a suitable sampling strategy: the distribution of  $[\boldsymbol{\mu}, \mathbf{u}_h]^T$  is then represented by a finite ensemble of vectors which are advanced in time according to the Kalman updating formula.

Since the EnKF also requires repetitive evaluations of solutions (and outputs) of the nonlinear dynamical system, solving an inverse problem in this context still represents a computational challenge. The cost of the inversion procedure can be considerably reduced by approximating the dynamical system through suitable surrogate or reduced order models (ROMs), see e.g. [1] for a survey of these techniques. In this work we exploit the RB method [33], relying on (i) the proper orthogonal decomposition (POD) technique for the basis functions construction and on (ii) hyper-reduction techniques, such as the discrete empirical interpolation method, for the efficient evaluation of the nonlinear terms involved in the forward problem [26, 3, 31]. Recently a great effort has been devoted to the reduction of computational complexity entailed by inverse problems dealing with PDE system. For instance, ROMs have been exploited both within MCMC [12, 11, 27, 4, 16], and Gaussian filtering [35, 5, 19] techniques to tackle state identification problems more efficiently. ROMs have also been considered in conjunction with *approximate Gaussian filters* for the sake of state estimation [18] and state/parameter estimation [25, 30]. With respect to already existing approaches, in this paper we develop for the first time a reduced basis ensemble Kalman filter (RB-EnKF) for the solution of state/parameter identification problems governed by nonlinear time-dependent PDEs. In this field, a computationally attractive alternative is represented by polynomial chaos expansion, a non-sampling-based method for uncertainty propagation in dynamical systems; this also has been applied for the efficient construction of *approximate Gaussian filters* for the sake of state estimation [32, 7] and state/parameter estimation [34].

When a full-order (or high-fidelity) model of the forward problem based on an approximation technique such as the finite element, finite volume or spectral methods, is replaced by a ROM, the propagation of the reduced-order approximation errors during the inversion procedure could lead to biased estimates of the unknown state/parameter [27, 18]. In order to enhance the accuracy of the inversion process when relying on very efficient (but very low-dimensional, too) ROM, we develop a statistical error model, to be included in the RB-EnKF to correct the bias on the output evaluation. Acting as a *calibration* of the reduced-order input/output map, this reduction error model (REM) can be obtained through a kriging interpolation procedure, thus extending a technique previously introduced in [27] to the case of forward problems described by unsteady PDEs.

To the best of our knowledge, this is the first investigation on the effect of the ROM accuracy during the filtering procedure. Our analysis shows that in the case of complex forward PDE problems, the ROM dimension could be significantly large for the sake of accuracy of the inversion procedure, thus yielding a dramatic loss of computational efficiency. On the other hand, equipping our RB-EnKF with a REM aims at improving the reliability of state/parameter identification problems also in those cases where the ROM has small dimension. The error analysis represents another original contribution of this paper: by adopting the perspective of functional data analysis, we are able to provide an efficient prediction of the propagation of the error on the estimated quantities.

The paper is organized as follows. In Sect. 2 we provide a general formulation of the class of problems we are interested to, whereas in Sect. 3 we introduce the ensemble Kalman filter, outlining the whole algorithm for state/parameter estimation. In Sect. 4 we exploit the RB method to solve nonlinear dynamical systems efficiently, and derive a RB formulation of the EnKF. Some theoretical results are also proven in order to carry out an a priori error analysis on the estimated quantities, with respect to the increasing accuracy of the underlying RB approximation. In Sect. 5 we introduce the kriging-based reduction error model, we incorporate it into the inversion procedure, and we provide an analysis of the estimation error with respect to the uncorrected RB-EnKF. In the last section, we present some numerical results exploiting the proposed procedure for the identification of unknown parameters or random fields when either a FitzHugh-Nagumo or a Fisher-Kolmogorov model is considered.

## 2 Problem formulation

In this section we introduce the class of problems we focus on, and recall some basic notions on the Bayesian approach instrumental to set up the whole framework. For the sake of simplicity, we directly deal with the algebraic formulation of the *full-order model* (FOM), obtained by a finite element approximation of the forward problem.

### 2.1 Forward problem

We consider as FOM of the forward problem a map assigning to any  $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^d$  an output vector

$$\mathbf{s}_h(\boldsymbol{\mu}) = \int_0^T \mathbf{H}\mathbf{u}_h(t; \boldsymbol{\mu}) dt \in \mathbb{R}^s, \quad (1)$$

where  $\mathbf{u}_h = \mathbf{u}_h(t; \boldsymbol{\mu})$  is the solution of a nonlinear parametrized dynamical system, arising from the spatial semi-discretization of a nonlinear parabolic PDE, under the form

$$\begin{cases} \mathbf{M}(\boldsymbol{\mu}) \frac{\partial \mathbf{u}_h}{\partial t} + \mathbf{A}(\boldsymbol{\mu})\mathbf{u}_h + \mathbf{N}(\mathbf{u}_h; \boldsymbol{\mu}) = \mathbf{f}_h(t; \boldsymbol{\mu}), & t \in (0, T) \\ \mathbf{u}_h(0; \boldsymbol{\mu}) = \mathbf{u}_0; \end{cases} \quad (2)$$

$\mathbf{H} \in \mathbb{R}^{s \times N_h}$  encodes the observation (linear) operator,  $\mathbf{M}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$  the mass matrix,  $\mathbf{f}_h \in \mathbb{R}^{N_h}$  the data of the PDE problem (source term and boundary conditions),  $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$  and  $\mathbf{N}(\cdot; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$  a parametrized linear differential operator and a nonlinear operator respectively. Problem (2) arises e.g. by applying a Galerkin-finite element (FE) method relying on a finite-dimensional space  $X_h \subset X$  of (possibly very large) dimension  $\dim(X_h) = N_h$ ;  $X = X(\Omega)$  denotes the functional space where the continuous problem is formulated, whereas  $\Omega \in \mathbb{R}^p$ ,  $p = 1, 2, 3$ , is the spatial domain;  $h$  is a parameter related to the mesh size of the computational grid.

Given a partition  $(t^{(\ell)}, t^{(\ell+1)})$ ,  $\ell = 0, \dots, N_t - 1$  of the time interval  $(0, T)$  into  $N_t$  subintervals of length  $\Delta t$ , we adopt the implicit Euler method for the time discretization, thus solving

$$\begin{cases} \left( \frac{\mathbf{M}(\boldsymbol{\mu})}{\Delta t} + \mathbf{A}(\boldsymbol{\mu}) \right) \mathbf{u}_h^{(\ell+1)} + \mathbf{N}(\mathbf{u}_h^{(\ell+1)}; \boldsymbol{\mu}) = \frac{\mathbf{M}(\boldsymbol{\mu})}{\Delta t} \mathbf{u}_h^{(\ell)} + \mathbf{f}_h^{(\ell+1)}(\boldsymbol{\mu}), & \ell = 0, \dots, N_t - 1 \\ \mathbf{u}_h^0 = \mathbf{u}_0. \end{cases} \quad (3)$$

To solve the nonlinear problem in (3), we rely on the Newton method as follows: at each time step  $\ell = 1, \dots, N_t - 1$ , while  $\|\boldsymbol{\delta}_u^i\| < tol$ , we solve

$$\mathbf{J}(\mathbf{u}_{h,i}^{(\ell+1)}) \boldsymbol{\delta}_u^i = \mathbf{r}(\mathbf{u}_{h,i}^{(\ell+1)}), \quad \mathbf{u}_{h,i+1}^{(\ell+1)} = \mathbf{u}_{h,i}^{(\ell+1)} + \boldsymbol{\delta}_u^i \quad i = 1, \dots, \quad (4)$$

being  $\mathbf{u}_{h,0}^{(\ell+1)} = \mathbf{u}_h^{(\ell)}$  and  $tol > 0$  a fixed small tolerance. The Jacobian matrix  $\mathbf{J}$  and the residual vector  $\mathbf{r}$  of the problem (2) are respectively given by

$$\mathbf{J}(\mathbf{v}) = \frac{\mathbf{M}(\boldsymbol{\mu})}{\Delta t} + \mathbf{A}(\boldsymbol{\mu}) + \mathbf{J}_N(\mathbf{v}; \boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h},$$

$$\mathbf{r}(\mathbf{v}) = \mathbf{f}_h(t^{(\ell+1)}; \boldsymbol{\mu}) - \frac{\mathbf{M}(\boldsymbol{\mu})}{\Delta t}(\mathbf{v} - \mathbf{u}_h^{(\ell)}) - \mathbf{A}(\boldsymbol{\mu})\mathbf{v} - \mathbf{N}(\mathbf{v}; \boldsymbol{\mu}) \in \mathbb{R}^{N_h},$$

being  $\mathbf{J}_N \in \mathbb{R}^{N_h \times N_h}$  the Jacobian matrix of the nonlinear term  $\mathbf{N}$ .

Since the state/parameter identification problem is performed sequentially over the time interval  $(0, T)$ , we need to evaluate the output (1) at various time steps instead than only once. Hence, we introduce a coarse partition of the time interval  $(0, T)$  into  $N_\tau$  windows  $(\tau^{(k)}, \tau^{(k+1)})$  of length  $\Delta\tau = K\Delta t$ , with  $k = 0, \dots, N_\tau - 1$  and  $K > 1$  (a sketch is reported in Fig. 1). Therefore, we consider as outputs over each window  $(\tau^{(k)}, \tau^{(k+1)})$  the values

$$\mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}) = \int_{\tau^{(k)}}^{\tau^{(k+1)}} \mathbf{H}\mathbf{u}_h(t; \boldsymbol{\mu}) dt \simeq \Delta t \mathbf{H} \sum_{\ell=Kk}^{K(k+1)} \omega_\ell \mathbf{u}_h(t^{(\ell)}; \boldsymbol{\mu}), \quad k = 0, \dots, N_\tau - 1, \quad (5)$$

where  $\omega_\ell$ ,  $\ell = Kk, \dots, K(k+1)$  are weights depending on the chosen quadrature formula.

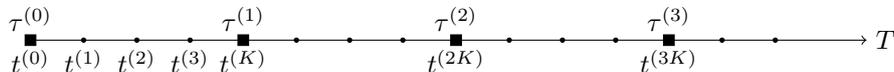


Figure 1: Example of partition of the time interval in windows of length  $\Delta\tau = K\Delta t$ ,  $K = 4$ .

**Remark 1.** An alternative choice for the output vector is given by the measured values at each  $t = \tau^{(k+1)}$ , i.e.

$$\mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}) = \mathbf{H}\mathbf{u}_h(\tau^{(k+1)}; \boldsymbol{\mu}) \quad k = 0, \dots, N_\tau - 1. \quad (6)$$

However, this choice provides less information than (5), thus affecting the estimation accuracy. We only use the outputs given by (6) when providing a detailed derivation of the Kalman update formula in the Appendix A, for the sake of simplicity.

## 2.2 Inverse problem

We formulate the problem of estimating  $\boldsymbol{\mu}^* \in \mathcal{P} \subset \mathbb{R}^d$  and  $\mathbf{u}^* \in X_h$  from noisy data  $\{\mathbf{s}^{(k)}\}_{k=1}^{N_\tau} \in \mathbb{R}^{N_\tau \times s}$  as a Bayesian inverse problem (BIP). The solution of the BIP is given by the posterior probability density function (pdf)  $\pi_{\text{post}} : \{\mathcal{P}, X_h\} \times \mathbb{R}^{N_\tau \times s} \rightarrow \mathbb{R}_0^+$  of the parameters  $\boldsymbol{\mu}^*$  and the state  $\mathbf{u}^*$  given the noisy data  $\{\mathbf{s}^{(k)}\}_{k=1}^{N_\tau}$ . Thanks to the *Bayes theorem*, we can express  $\pi_{\text{post}}$  as

$$\pi_{\text{post}} \left( \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h \end{array} \right] \middle| \left\{ \mathbf{s}^{(k)} \right\}_{k=1}^{N_\tau} \right) = \frac{1}{\eta(\mathbf{s})} \pi \left( \left\{ \mathbf{s}^{(k)} \right\}_{k=1}^{N_\tau} \middle| \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_0 \end{array} \right] \right) \pi_{\text{prior}} \left( \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_0 \end{array} \right] \right), \quad (7)$$

being  $\pi_{\text{prior}} : \{\mathcal{P}, X_h\} \rightarrow \mathbb{R}_0^+$  the *prior* pdf of the parameters,  $\pi : \mathbb{R}^{N_\tau \times s} \times \{\mathcal{P}, X_h\} \rightarrow \mathbb{R}_0^+$  the *likelihood* function and  $\eta(\mathbf{s})$  a suitable norming constant, which does not affect the inversion step. We consider the (rather classical) additive noise model:

$$\mathbf{s}^{(k)} = \mathbf{s}_h^{(k)}(\boldsymbol{\mu}) + \boldsymbol{\varepsilon}_{\text{noise}}, \quad \forall k = 1, \dots, N_\tau \quad (8)$$

where we assume that the noise is modeled by a gaussian random variable  $\boldsymbol{\varepsilon}_{\text{noise}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ .

Due to the nonlinearity of the forward map  $\boldsymbol{\mu} \rightarrow \mathbf{s}_h(\boldsymbol{\mu})$ , a closed form for the posterior distribution  $\pi_{\text{post}}$  cannot be obtained; instead, sampling methods like MCMC techniques are exploited to approximate  $\pi_{\text{post}}$ . Nevertheless, for the case at hand, the computationally expensive solution of the nonlinear dynamical system (3), required to evaluate  $\mathbf{s}_h(\boldsymbol{\mu})$ , would make MCMC-based inversion extremely inefficient. A more feasible option is represented by Bayesian filtering techniques, which are well-suited for the sequential estimation in the case of BIPs featuring nonstationary forward models. The aim of a filtering technique is to sequentially approximate the posterior pdf of  $(\boldsymbol{\mu}, \mathbf{u}_h)$  through a sequence of pdfs  $\{\pi^{(k)}\}_{k=1}^{N_\tau}$ , given the measurement vectors  $\{\mathbf{s}^{(k)}\}_{k=1}^{N_\tau}$ , exploiting an update algorithm based on the following *prediction-analysis* procedure at the generic step  $k$ :

1. *Prediction stage:* at step  $k$ , given

$$\pi^{(k)} \left( \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h^{(k)} \end{array} \right] \middle| \mathbf{s}^{(k)} \right),$$

compute  $\mathbf{u}_h^{(k+1)} = \mathbf{u}_h(\tau^{(k+1)}; \cdot)$  by solving (3) on the window  $[\tau^{(k)}, \tau^{(k+1)})$  with initial datum  $\mathbf{u}_h(\tau^{(k)}, \cdot)$ , to obtain the *prior* for the  $k+1$  step

$$\pi^{(k)} \left( \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h^{(k+1)} \end{array} \right] \middle| \mathbf{s}^{(k)} \right);$$

then, compute the corresponding output  $\mathbf{s}_h^{(k+1)}$  and evaluate the likelihood

$$\pi \left( \mathbf{s}^{(k+1)} \middle| \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h^{(k+1)} \end{array} \right] \right) = (2\pi)^{(-\frac{s}{2})} |\Gamma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{s}^{(k+1)} - \mathbf{s}_h^{(k+1)})^T \Gamma^{-1} (\mathbf{s}^{(k+1)} - \mathbf{s}_h^{(k+1)}) \right\}, \quad (9)$$

assuming that  $\boldsymbol{\varepsilon}_{noise}$  is normally distributed.

2. *Analysis stage:* update the posterior distribution according to the Bayes theorem:

$$\pi^{(k+1)} \left( \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h^{(k+1)} \end{array} \right] \middle| \mathbf{s}^{(k+1)} \right) = \frac{1}{\eta(\mathbf{s})} \pi \left( \mathbf{s}^{(k+1)} \middle| \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h^{(k+1)} \end{array} \right] \right) \pi^{(k)} \left( \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{u}_h^{(k+1)} \end{array} \right] \middle| \mathbf{s}^{(k)} \right), \quad (10)$$

where the prior  $\pi_{\text{prior}} = \pi^{(k)}$  is nothing but the posterior evaluated at stage  $k$ .

At the end of this procedure,  $\pi_{\text{post}} = \pi^{(N_\tau)}$  is the solution of the BIP. In order to efficiently update  $\{\pi^{(k)}\}_{k=1}^{N_\tau}$ , it is necessary to consider specific Bayesian filters, known as Gaussian filters, which assume the noise to be Gaussian and the output  $\mathbf{s}_h$  to be linear with respect to  $\boldsymbol{\mu}$  (see e.g. [22]). Since the map  $\boldsymbol{\mu} \rightarrow \mathbf{s}_h(\boldsymbol{\mu})$  is usually not linear, suitable approximations must be introduced to handle more general cases. Among the *approximated* Gaussian filters, we consider the *ensemble Kalman filter*, which takes advantage of a randomly generated sample to successively approximate the distribution of  $\{\pi^{(k)}\}_{k=1}^{N_\tau}$ . In the following section we recall the basic features of this technique, a more detailed overview can be found e.g. in [8, 20].

### 3 Ensemble Kalman filter

The ensemble Kalman filter (EnKF) is a recursive filter, based on the original Kalman filter, widely exploited in data assimilation and forecasting, see e.g. [9, 17, 34]. This method is based on the idea of updating an *ensemble* of particles using the prediction/analysis procedure introduced in the previous section. In our case, by *ensemble* of particle we mean a sample of  $N_e$  parameter vectors

$$\mathcal{P}_h^{(k)} = \{\boldsymbol{\mu}_{h,q}^{(k)}\}_{q=1}^{N_e}, \quad k = 1, \dots, N_\tau,$$

where  $\boldsymbol{\mu}_{h,q}^{(k)}$  denotes the value of a parameter vector  $\boldsymbol{\mu}_{h,q}$  at the  $k$ -th iteration, and the associated ensemble of  $N_e$  state solutions

$$\mathcal{U}_h^{(k)} = \{\mathbf{u}_h^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{P}_h^{(k)}\}, \quad k = 1, \dots, N_\tau.$$

Moreover, let us introduce, for any  $k = 1, \dots, N_\tau$ , the sample mean vectors

$$\bar{\mathbf{u}}_h^{(k)} = \frac{1}{N_e} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} \mathbf{u}_h^{(k)}(\boldsymbol{\mu}), \quad \bar{\mathbf{s}}_h^{(k)} = \frac{1}{N_e} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} \mathbf{s}_h^{(k)}(\boldsymbol{\mu}), \quad \bar{\boldsymbol{\mu}}_h^{(k)} = \frac{1}{N_e} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k)}} \boldsymbol{\mu}, \quad (11)$$

the sample covariance of the outputs

$$\mathbf{C}_{s_h s_h}^{(k)} = \frac{1}{N_e - 1} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} (\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{s}}_h^{(k)}) (\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{s}}_h^{(k)})^T \in \mathbb{R}^{s \times s} \quad (12)$$

and the sample cross-covariances

$$\mathbf{C}_{\mu_h s_h}^{(k)} = \frac{1}{N_e - 1} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} (\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}^{(k-1)}) (\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{s}}_h^{(k)})^T \in \mathbb{R}^{d \times s}, \quad (13)$$

$$\mathbf{C}_{u_h s_h}^{(k)} = \frac{1}{N_e - 1} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} (\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{u}}_h^{(k)}) (\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{s}}_h^{(k)})^T \in \mathbb{R}^{N_h \times s}. \quad (14)$$

All these quantities will be used in the following to characterize the updating formula of the Kalman Filter. Hence, starting from the initial ensemble  $\{\mathcal{P}_h^{(0)}, \mathcal{U}_h^{(0)}\}$  sampled from the prior distribution, the *prediction-analysis* procedure of the EnKF is given by the following two stages recursion:

1. *prediction stage*:

- compute the solution  $\mathbf{u}_h^{(k+1)}(\boldsymbol{\mu})$  of the forward problem over  $[\tau^{(k)}, \tau^{(k+1)})$  by solving (3) with initial datum  $\mathbf{u}_h^{(k)}(\boldsymbol{\mu}) \in \mathcal{U}^{(k)}$ , and the relative output  $\mathbf{s}_h^{(k+1)}(\boldsymbol{\mu})$  for each  $\boldsymbol{\mu} \in \mathcal{P}_h^{(k)}$ ;
- compute the sample means  $\bar{\mathbf{u}}_h^{(k+1)}$ ,  $\bar{\mathbf{s}}_h^{(k+1)}$  and  $\bar{\boldsymbol{\mu}}^{(k)}$ ;
- compute the sample covariance  $\mathbf{C}_{s_h s_h}^{(k+1)}$  and the cross-covariances  $\mathbf{C}_{\mu_h s_h}^{(k+1)}$  and  $\mathbf{C}_{u_h s_h}^{(k+1)}$ .

2. *analysis stage*: update the state/parameters ensemble by taking advantage of the new information from the prediction stage, through the following *Kalman formula*:

$$\begin{bmatrix} \boldsymbol{\mu}_{h,q}^{(k+1)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_{h,q}^{(k+1)}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{h,q}^{(k)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_{h,q}^{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mu_h s_h}^{(k+1)} \\ \mathbf{C}_{u_h s_h}^{(k+1)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k+1)})^{-1} (\mathbf{s}^{(k+1)} - \mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}_{h,q}^{(k)})), \quad (15)$$

for each  $q = 1, \dots, N_e$ .

Note that even though the algorithm is derived under Gaussian assumptions, the ensemble is not prescribed to be Gaussian as it evolves under nonlinear dynamics. Then, at each iteration we estimate the unknown parameter vector by computing the sample mean  $\hat{\boldsymbol{\mu}}_h = \bar{\boldsymbol{\mu}}_h$ , and compute an empirical confidence region from the updated ensemble  $\mathcal{P}_h^{(k+1)}$ . For a formal derivation of the EnKF algorithm see Appendix A, while for further properties and results, see e.g. [2, 23, 6]. A detailed description of the procedure is reported in Algorithm 1.

---

**Algorithm 1** Full-order Ensemble Kalman filter procedure

---

- 1: **procedure** ENKF
  - 2:   *Initialization*
  - 3:    $\{\mathcal{P}_h^{(0)}, \mathcal{U}_h^{(0)}\} \leftarrow$  sampling  $N_e$  particles from  $\pi_{\text{prior}}$
  - 4:   **for**  $k = 0 : N_\tau - 1$  **do**
  - 5:     *Prediction stage*:
  - 6:     **for**  $[\boldsymbol{\mu}, \mathbf{u}_h^{(k)}(\boldsymbol{\mu})]^T \in \{\mathcal{P}_h^{(k)}, \mathcal{U}_h^{(k)}\}$  **do**
  - 7:        $\mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}) \leftarrow$  solve forward problem (3) with initial datum  $\mathbf{u}_h^{(k)}(\boldsymbol{\mu})$
  - 8:       compute means  $\bar{\mathbf{u}}_h^{(k+1)}$ ,  $\bar{\mathbf{s}}_h^{(k+1)}$ , and  $\bar{\boldsymbol{\mu}}_h^{(k)}$  by (11)
  - 9:       compute covariance  $\mathbf{C}_{s_h s_h}^{(k+1)}$  by (12)
  - 10:       compute cross-covariances  $\mathbf{C}_{u_h s_h}^{(k+1)}$  and  $\mathbf{C}_{\mu_h s_h}^{(k+1)}$  by (13)
  - 11:     *Update stage*:
  - 12:     **for**  $[\boldsymbol{\mu}] \in \{\mathcal{P}_h^{(k)}\}$  **do**
  - 13:       update each state/parameters particle using (15)
- 

## 4 Reduced-order model

Even if the sequential update through the EnKF requires, at each step, the solution of the forward problem on the window  $[\tau^{(k)}, \tau^{(k+1)})$  for each particle in the ensemble, the large size of this latter entails very large computational costs. The nonlinear nature of the forward problem makes the

whole estimation process even more challenging. For these reasons we rather rely on the RB method for the fast evaluation of the *prediction* stage, taking advantage of a further hyper-reduction procedure based on the (discrete) empirical interpolation method for managing nonlinear terms more efficiently.

#### 4.1 Reduced basis method

The RB method is a projection-based ROM which computes an approximation  $\mathbf{u}_n(t; \boldsymbol{\mu})$  of  $\mathbf{u}_h(t; \boldsymbol{\mu})$  (as well as an approximation  $\mathbf{s}_n(\boldsymbol{\mu})$  of the output  $\mathbf{s}_h(\boldsymbol{\mu})$ ) by means of a Galerkin projection on a reduced subspace  $X_n \subset X_h$  of very small dimension  $n \ll N_h$  (see e.g. [33] for a detailed overview). Here we construct such a space by means of the *proper orthogonal decomposition* (POD) technique. This latter selects as basis functions the first  $n$  singular vectors (corresponding to the largest  $n$  singular values) of the snapshot matrix, whose columns are given by the full-order solution  $\mathbf{u}_h(t^{(\ell)}; \boldsymbol{\mu}_j)$ , computed for each time-step  $t^{(\ell)}$ ,  $\ell = 1, \dots, N_t$  and a large training set  $\boldsymbol{\mu}_j \in S_{train} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{N_{train}}\}$ . Given the matrix  $\mathbf{V} \in \mathbb{R}^{N_h \times n}$  which collects the basis functions, we approximate the full-order solution  $\mathbf{u}_h(t; \boldsymbol{\mu}) \approx \mathbf{V}\mathbf{u}_n$ . By projecting (2) onto the space generated by the columns of  $\mathbf{V} \in \mathbb{R}^{N_h \times n}$  we obtain: find  $\mathbf{u}_n(t; \boldsymbol{\mu}) \in \mathbb{R}^n$  as the solution of the following reduced nonlinear parametrized dynamical system:

$$\begin{cases} \mathbf{M}_n(\boldsymbol{\mu}) \frac{\partial \mathbf{u}_n}{\partial t} + \mathbf{A}_n(\boldsymbol{\mu})\mathbf{u}_n + \mathbf{V}^T \mathbf{N}(\mathbf{V}\mathbf{u}_n; \boldsymbol{\mu}) = \mathbf{f}_n(t; \boldsymbol{\mu}), & t \in (0, T) \\ \mathbf{u}_n(0; \boldsymbol{\mu}) = \mathbf{V}^T \mathbf{u}_0 \end{cases} \quad (16)$$

where the reduced arrays are given by

$$\mathbf{M}_n(\boldsymbol{\mu}) = \mathbf{V}^T \mathbf{M}(\boldsymbol{\mu}) \mathbf{V}, \quad \mathbf{A}_n(\boldsymbol{\mu}) = \mathbf{V}^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{V}, \quad \mathbf{f}_n = \mathbf{V}^T \mathbf{f}_h.$$

Proceeding similarly to the full-order case concerning time discretization, the implicit Euler method applied to (16) yields the following dynamical system:

$$\begin{cases} \left( \frac{\mathbf{M}_n(\boldsymbol{\mu})}{\Delta t} + \mathbf{A}_n(\boldsymbol{\mu}) \right) \mathbf{u}_n^{(\ell+1)} + \mathbf{V}^T \mathbf{N}(\mathbf{V}\mathbf{u}_n^{(\ell+1)}; \boldsymbol{\mu}) \\ = \frac{\mathbf{M}_n(\boldsymbol{\mu})}{\Delta t} \mathbf{u}_n^{(\ell)} + \mathbf{f}_n(t^{(\ell+1)}; \boldsymbol{\mu}), & \ell = 0, \dots, N_t - 1 \\ \mathbf{u}_h^0 = \mathbf{u}_0. \end{cases} \quad (17)$$

Due to the presence of the nonlinear term  $\mathbf{N}(\cdot; \boldsymbol{\mu})$ , we use the Newton method as follows: while  $\|\boldsymbol{\delta}_{u_n}^i\| < tol$ , we solve

$$\mathbf{J}_n(\mathbf{u}_{n,i}^{(\ell+1)}) \boldsymbol{\delta}_{u_n}^i = \mathbf{r}_n(\mathbf{u}_{n,i}^{(\ell+1)}), \quad \mathbf{u}_{n,i+1}^{(\ell+1)} = \mathbf{u}_{n,i}^{(\ell+1)} + \boldsymbol{\delta}_{u_n}^i \quad i = 1, \dots, \quad (18)$$

with  $\mathbf{u}_{n,0}^{(\ell+1)} = \mathbf{u}_n^{(\ell)}$ . The reduced Jacobian matrix  $\mathbf{J}_n \in \mathbb{R}^{n \times n}$  and the reduced residual vector  $\mathbf{r}_n \in \mathbb{R}^n$  of the problem (17) are

$$\mathbf{J}_n(\mathbf{v}) = \frac{\mathbf{M}_n(\boldsymbol{\mu})}{\Delta t} + \mathbf{A}_n(\boldsymbol{\mu}) + \mathbf{V}^T \mathbf{J}_N(\mathbf{V}\mathbf{v}; \boldsymbol{\mu}), \quad \mathbf{v} \in \mathbb{R}^n,$$

and

$$\mathbf{r}_n(\mathbf{v}) = \mathbf{f}_n(t^{(\ell+1)}; \boldsymbol{\mu}) - \frac{\mathbf{M}_n(\boldsymbol{\mu})}{\Delta t} (\mathbf{v} - \mathbf{u}_n^{(\ell)}) - \mathbf{A}_n(\boldsymbol{\mu})\mathbf{v} - \mathbf{N}_n(\mathbf{v}; \boldsymbol{\mu}),$$

respectively. For the ROM output evaluation, we consider the integral of the reduced output vector over each interval  $(\tau^{(k)}, \tau^{(k+1)})$ , that is

$$\mathbf{s}_n^{(k+1)}(\boldsymbol{\mu}) = \int_{\tau^{(k)}}^{\tau^{(k+1)}} \mathbf{H}_n \mathbf{u}_n(t; \boldsymbol{\mu}) dt, \quad \forall k = 0, \dots, N_\tau - 1,$$

where the reduced output operator is defined as  $\mathbf{H}_n = \mathbf{H}\mathbf{V} \in \mathbb{R}^{s \times n}$ . As a consequence the additive

noise model (8) is now replaced by

$$\mathbf{s}^{(k+1)} = \mathbf{s}_n^{(k+1)}(\boldsymbol{\mu}) + \boldsymbol{\varepsilon}_{\text{noise}}, \quad \forall k = 0, \dots, N_\tau - 1.$$

The efficient evaluation of the reduced arrays appearing in (17) as time and parameters vary is still a challenging task in order to achieve an efficient online evaluation of a ROM when dealing with nonlinear (and/or complex nonaffine) terms. Indeed, under the assumption of *affine parametric dependence*, those arrays can be expressed as the finite sum of products between  $\boldsymbol{\mu}$ -dependent functions and  $\boldsymbol{\mu}$ -independent operators [33]. In nonaffine cases, an (approximate) affine approximation can be recovered by means of the *empirical interpolation method* (EIM), see e.g. [26]. For instance, given a nonaffine operator  $\mathbf{A}(\boldsymbol{\mu})$ , the EIM approximation reads

$$\mathbf{A}(\boldsymbol{\mu}) \approx \sum_{j=1}^{m_{\text{EIM}}} \beta_j(\boldsymbol{\mu}) \mathbf{A}^j$$

where for any  $\boldsymbol{\mu} \in \mathcal{P}$ , the coefficients  $\{\beta_j(\boldsymbol{\mu})\}_{j=1}^{m_{\text{EIM}}}$  are evaluated by solving a linear system of dimension  $m_{\text{EIM}} \times m_{\text{EIM}}$ , arising by the imposition of  $m_{\text{EIM}}$  interpolation constraints over a set of  $m_{\text{EIM}}$  magic points selected according to a suitable greedy procedure (see [26]). Given this approximation, the reduced operator can then be obtained by projecting each  $\boldsymbol{\mu}$ -independent operators  $\mathbf{A}^j$ , that is,

$$\mathbf{A}_n(\boldsymbol{\mu}) = \sum_{k=1}^{m_{\text{EIM}}} \beta_k(\boldsymbol{\mu}) \mathbf{A}_n^k, \quad \mathbf{A}_n^j = \mathbf{V}^T \mathbf{A}^j \mathbf{V}.$$

In this way, computations can be decoupled into an expensive  $\boldsymbol{\mu}$ -independent *offline* stage and a very inexpensive  $\boldsymbol{\mu}$ -dependent *online* stage, to be performed several times during the inversion algorithm. Unfortunately, when dealing with nonlinear operators, evaluating  $\mathbf{V}^T \mathbf{N}(\mathbf{V}; \boldsymbol{\mu})$  would also depend on the FOM size  $N_h$ , and therefore still very expensive. To overcome this problem, the (*discrete*) *empirical interpolation method* (DEIM) can be exploited at each iteration of the Newton algorithm to handle the  $\boldsymbol{\mu}$ -dependent nonlinear terms efficiently, as proposed in [3] and further discussed in [31]. In particular, the DEIM approximation of a nonlinear operator  $\mathbf{N} : \mathbb{R}^{N_h} \rightarrow \mathbb{R}^{N_h}$  is given by

$$\mathbf{N}(\mathbf{u}_h; \boldsymbol{\mu}) \approx \sum_{j=1}^{m_D} c_j(\mathbf{u}_h; \boldsymbol{\mu}) \phi_j(\mathbf{x}),$$

where the coefficients  $\{c_j\}_{k=1}^{m_D}$  can be computed at each iteration by solving a linear system of dimension  $m_D \times m_D$ . In particular, by defining the basis matrix  $\mathbf{U} = [\phi_1, \dots, \phi_{m_D}]$  and the index matrix  $\mathbf{P} = [\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_{m_D}}]$ , we get the following reduced approximation of the nonlinear function

$$\mathbf{N}(\mathbf{u}_n) \approx \underbrace{\mathbf{V}^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1}}_{n \times m_D} \underbrace{\mathbf{N}(\mathbf{P}^T \mathbf{V} \mathbf{u}_n; \boldsymbol{\mu})}_{m_D \times 1}.$$

In the case of the implicit Euler method, also the Jacobian  $\mathbf{J}_N$  has to be assembled efficiently at each Newton step. For the case at hand, we can directly differentiate the previous formula, yielding

$$\mathbf{J}_N(\mathbf{u}_n; \boldsymbol{\mu}) \approx \underbrace{\mathbf{V}^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1}}_{n \times m_D} \underbrace{\mathbf{J}_N(\mathbf{P}^T \mathbf{V} \mathbf{u}_n; \boldsymbol{\mu})}_{m_D \times m_D} \underbrace{\mathbf{P}^T \mathbf{V}}_{m_D \times n}.$$

Alternative solutions can be obtained by considering an extension of DEIM for sparse Jacobians, known as *matrix* DEIM; see also the related discussion in [31, 36].

## 4.2 Reduced basis Ensemble Kalman filter

Given a suitable ROM for approximating the solution of the forward problem, a reduced-order EnKF can be obtained by replacing the full-order output evaluation with the reduced-order one. Since we have adopted the RB method, we will refer to the resulting procedure as to the reduced basis ensemble Kalman filter (RB-EnKF). We define the ensemble of  $N_e$  parameters as

$$\mathcal{P}_n^{(k)} = \{\boldsymbol{\mu}_{n,q}^{(k)}\}_{q=1}^{N_e}, \quad k = 0, \dots, N_\tau,$$

and the associated ensemble of reduced state solution

$$\mathcal{U}_n^{(k)} = \{\mathbf{u}_n(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{P}_n^{(k)}\}, \quad k = 0, \dots, N_\tau.$$

Consequently, we also compute the means (11), the covariance (12) and the cross-covariances (13) by relying on the reduced-order quantities. Hence, starting from the initial ensemble  $\{\mathcal{P}_n^{(0)}, \mathcal{U}_n^{(0)}\}$ , directly sampled from the prior, the RB-EnKF can be built with a two-stage recursion, similarly to what we did in Section 3:

1. *prediction stage*:

- compute the reduced solution  $\mathbf{u}_n^{(k+1)}$  of the forward ROM (17) on the window  $[\tau^{(k)}, \tau^{(k+1)})$  with initial datum  $\mathbf{u}_n^{(k)}(\boldsymbol{\mu}) \in \mathcal{U}_n^{(k)}$ , and the relative output  $\mathbf{s}_n^{(k+1)}$  for each  $\boldsymbol{\mu} \in \mathcal{P}_n^{(k)}$ ;
- compute the sample means  $\bar{\mathbf{u}}_n^{(k+1)}$ ,  $\bar{\mathbf{s}}_n^{(k+1)}$  and  $\bar{\boldsymbol{\mu}}_n^{(k)}$ ;
- compute the sample covariance  $\mathbf{C}_{s_n s_n}^{(k+1)} \in \mathbb{R}^{s \times s}$  and the cross-covariance matrices  $\mathbf{C}_{\mu_n s_n}^{(k)} \in \mathbb{R}^{d \times s}$  and  $\mathbf{C}_{u_n s_n}^{(k+1)} \in \mathbb{R}^{n \times s}$ , using formulas (12)–(14) by substituting  $\mathbf{s}_h$ ,  $\mathbf{u}_h$  and  $\mathcal{P}_h^{(k-1)}$  with  $\mathbf{s}_n$ ,  $\mathbf{u}_n$  and  $\mathcal{P}_n^{(k-1)}$ , respectively;

2. *analysis stage*: update the state/parameters ensemble through the following *reduced Kalman formula*:

$$\begin{bmatrix} \boldsymbol{\mu}_{n,q}^{(k+1)} \\ \mathbf{u}_n^{(k+1)}(\boldsymbol{\mu}_{n,q}^{(k+1)}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{n,q}^{(k)} \\ \mathbf{u}_n^{(k+1)}(\boldsymbol{\mu}_{n,q}^{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mu_n s_n}^{(k+1)} \\ \mathbf{C}_{u_n s_n}^{(k+1)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k+1)})^{-1} (\mathbf{s}^{(k+1)} - \mathbf{s}_n^{(k+1)}(\boldsymbol{\mu}_{n,q}^{(k)})), \quad (19)$$

for each  $q = 1, \dots, N_e$ .

In this way, we are neglecting the error between the ROM and the FOM, which nevertheless might affect the accuracy of the inverse problem solution, as shown in the following section.

### 4.3 Effectivity of the RB-EnKF

We now want to prove that the solution of the inverse problem given by the RB-EnKF converges, as long as the ROM dimension increases, to the one which would be obtained by relying on the full-order EnKF. For that we compare the resulting state/parameters ensemble  $\{\mathcal{P}^{(N_\tau)}, \mathcal{U}^{(N_\tau)}\}$  with the reduced ones  $\{\mathcal{P}_n^{(N_\tau)}, \mathcal{U}_n^{(N_\tau)}\}$ , for any dimension  $n$ . To this goal, let us denote by

$$\mathbf{e}^{(k)} = \begin{bmatrix} \mathbf{e}_\mu^{(k)} \\ \mathbf{e}_u^{(k)} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\mu}}_n^{(k)} \\ \hat{\mathbf{u}}_n^{(k)} \end{bmatrix} - \begin{bmatrix} \hat{\boldsymbol{\mu}}_n^{(k)} \\ \mathbf{V} \hat{\mathbf{u}}_n^{(k)} \end{bmatrix}, \quad k = 1, \dots, N_\tau - 1,$$

the error between the means  $[\hat{\boldsymbol{\mu}}_n^{(k)}, \mathbf{V} \hat{\mathbf{u}}_n^{(k)}]^T$  and  $[\hat{\boldsymbol{\mu}}_n^{(k)}, \hat{\mathbf{u}}_n^{(k)}]^T$  computed over the respective ensembles. Then, let us denote by

$$\gamma_h^{(k)} = \|(\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1}\| \quad \gamma_n^{(k)} = \|(\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1}\|,$$

being  $\|\cdot\|$  the Euclidean norm. Then, at each step we can bound the error  $\mathbf{e}^{(k)}$  as follows.

**Theorem 2.** *For any  $0 < n < N_h$ ,  $k = 1, \dots, N_\tau$ , the following relationships hold:*

$$\|\mathbf{e}_\mu^{(k)}\| \leq \|\mathbf{e}_\mu^{(k-1)}\| + c_{\mu,1}^{(k)} \|\bar{\mathbf{s}}_h^{(k)} - \bar{\mathbf{s}}_n^{(k)}\| + c_{\mu,2}^{(k)} \|\mathbf{C}_{\mu_h s_h}^{(k)} - \mathbf{C}_{\mu_n s_n}^{(k)}\| + c_{\mu,3}^{(k)} \|\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}\|, \quad (20)$$

$$\|\mathbf{e}_u^{(k)}\| \leq \|\mathbf{e}_u^{(k-1)}\| + c_{u,1}^{(k)} \|\bar{\mathbf{s}}_h^{(k)} - \bar{\mathbf{s}}_n^{(k+1)}\| + c_{u,2}^{(k)} \|\mathbf{C}_{u_h s_h}^{(k)} - \mathbf{V} \mathbf{C}_{u_n s_n}^{(k)}\| + c_{u,3}^{(k)} \|\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}\|, \quad (21)$$

where

$$\begin{aligned} c_{\mu,1}^{(k)} &= \gamma_h^{(k)} \|\mathbf{C}_{\mu_h s_h}^{(k)}\|, & c_{u,1}^{(k)} &= \gamma_h^{(k)} \|\mathbf{C}_{u_h s_h}^{(k)}\|, \\ c_{\mu,2}^{(k)} &= c_{u,2}^{(k)} = \gamma_h^{(k)} \|\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}\|, \\ c_{\mu,3}^{(k)} &= \gamma_n^{(k)} \gamma_h^{(k)} \|\mathbf{C}_{\mu_n s_n}^{(k)}\|_{\mathbb{R}^{d \times s}} \|\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}\|, & c_{u,3}^{(k)} &= \gamma_n^{(k)} \gamma_h^{(k)} \|\mathbf{V} \mathbf{C}_{u_n s_n}^{(k)}\| \|\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}\|. \end{aligned}$$

*Proof.* By averaging over the sample  $\mathcal{P}^{(k-1)}$  of (15), we obtain the following update equation for the estimate  $[\hat{\boldsymbol{\mu}}_h; \hat{\mathbf{u}}_h]$

$$\begin{bmatrix} \hat{\boldsymbol{\mu}}_h^{(k)} \\ \hat{\mathbf{u}}_h^{(k)} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\mu}}_h^{(k-1)} \\ \hat{\mathbf{u}}_h^{(k-1)} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mu_h s_h}^{(k)} \\ \mathbf{C}_{u_h s_h}^{(k)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1} (\mathbf{s}^{(k)} - \bar{\mathbf{s}}_h^{(k)}). \quad (22)$$

By doing the same on (19) we have

$$\begin{bmatrix} \hat{\boldsymbol{\mu}}_n^{(k)} \\ \hat{\mathbf{u}}_n^{(k)} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\mu}}_n^{(k-1)} \\ \hat{\mathbf{u}}_n^{(k-1)} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mu_n s_n}^{(k)} \\ \mathbf{C}_{u_n s_n}^{(k)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1} (\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}). \quad (23)$$

By subtracting (23) from (22), we can express  $\mathbf{e}^{(k)} = \mathbf{e}^{(k-1)} + \mathbf{e}_I^{(k)} + \mathbf{e}_{II}^{(k)}$ , being

$$\begin{aligned} \mathbf{e}_I^{(k)} &= \begin{bmatrix} \mathbf{C}_{\mu_h s_h}^{(k)} \\ \mathbf{C}_{u_h s_h}^{(k)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1} (\bar{\mathbf{s}}_n^{(k)} - \bar{\mathbf{s}}_h^{(k)}), \\ \mathbf{e}_{II}^{(k)} &= \left( \begin{bmatrix} \mathbf{C}_{\mu_h s_h}^{(k)} \\ \mathbf{C}_{u_h s_h}^{(k)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1} - \begin{bmatrix} \mathbf{C}_{\mu_n s_n}^{(k)} \\ \mathbf{V}\mathbf{C}_{u_n s_n}^{(k)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1} \right) (\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}). \end{aligned} \quad (24)$$

Then, the following error estimates hold

$$\|\mathbf{e}_{I,\mu}^{(k)}\| \leq c_{\mu,1}^{(k)} \|\bar{\mathbf{s}}_n^{(k)} - \bar{\mathbf{s}}_h^{(k)}\|, \quad c_{\mu,1}^{(k)} = \gamma_h^{(k)} \|\mathbf{C}_{\mu_h s_h}^{(k)}\|, \quad (25)$$

$$\|\mathbf{e}_{I,u}^{(k)}\| \leq c_{u,1}^{(k)} \|\bar{\mathbf{s}}_n^{(k)} - \bar{\mathbf{s}}_h^{(k)}\|, \quad c_{u,1}^{(k)} = \gamma_h^{(k)} \|\mathbf{C}_{u_h s_h}^{(k)}\|, \quad (26)$$

respectively. On the other hand, by adding and subtracting in (24) the quantity

$$\begin{bmatrix} \mathbf{C}_{\mu_n s_n}^{(k)} \\ \mathbf{V}\mathbf{C}_{u_n s_n}^{(k)} \end{bmatrix} (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1}$$

and rewriting the expression (24) as  $\mathbf{e}_{II}^{(k)} = \mathbf{e}_i^{(k)} + \mathbf{e}_{ii}^{(k)}$ , with

$$\begin{aligned} \mathbf{e}_i^{(k)} &= \left( \begin{bmatrix} \mathbf{C}_{\mu_h s_h}^{(k)} \\ \mathbf{C}_{u_h s_h}^{(k)} \end{bmatrix} - \begin{bmatrix} \mathbf{C}_{\mu_n s_n}^{(k)} \\ \mathbf{V}\mathbf{C}_{u_n s_n}^{(k)} \end{bmatrix} \right) (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1} (\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}), \\ \mathbf{e}_{ii}^{(k)} &= \begin{bmatrix} \mathbf{C}_{\mu_n s_n}^{(k)} \\ \mathbf{V}\mathbf{C}_{u_n s_n}^{(k)} \end{bmatrix} \left( (\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1} - (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1} \right) (\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)}), \end{aligned}$$

we have

$$\begin{aligned} \|\mathbf{e}_{i,\mu}^{(k)}\| &\leq c_{\mu,2}^{(k)} \|\mathbf{C}_{\mu_h s_h}^{(k)} - \mathbf{C}_{\mu_n s_n}^{(k)}\| & c_{\mu,2}^{(k)} &= \gamma_h^{(k)} \|(\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)})\|, \\ \|\mathbf{e}_{i,u}^{(k)}\| &\leq c_{u,2}^{(k)} \|\mathbf{C}_{u_h s_h}^{(k)} - \mathbf{V}\mathbf{C}_{u_n s_n}^{(k)}\| & c_{u,2}^{(k)} &= \gamma_h^{(k)} \|(\mathbf{s}^{(k)} - \bar{\mathbf{s}}_n^{(k)})\|. \end{aligned} \quad (27)$$

By applying the Sherman-Morrison-Woodbury Formula (see, e.g., Sect. 2.4 in [14]) we have that

$$(\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1} - (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1} = (\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1} (\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}) (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1},$$

and, consequently,

$$\|(\boldsymbol{\Gamma} + \mathbf{C}_{s_n s_n}^{(k)})^{-1} - (\boldsymbol{\Gamma} + \mathbf{C}_{s_h s_h}^{(k)})^{-1}\| \leq \gamma_n^{(k+1)} \gamma_h^{(k)} \|\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}\|,$$

whence the following bounds hold for  $\mathbf{e}$

$$\|\mathbf{e}_{ii,\mu}^{(k)}\| \leq c_{\mu,3}^{(k)} \|\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}\|, \quad \|\mathbf{e}_{II,u}^{(k)}\| \leq c_{u,3}^{(k)} \|\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}\|. \quad (28)$$

Finally, by combining (25), (26), (27) and (28), we obtain (20)–(21).  $\square$

In order to obtain accurate state/parameters estimates when employing our proposed RB-EnKF, we thus require the ROM to be able to generate similar means  $\bar{\mathbf{s}}_n^{(k)}(\boldsymbol{\mu}) \simeq \bar{\mathbf{s}}_h^{(k)}(\boldsymbol{\mu})$  and similar covariance matrices  $\mathbf{C}_{s_n s_n}^{(k)} \simeq \mathbf{C}_{s_h s_h}^{(k)}$  and  $\mathbf{C}_{\mu_n s_n}^{(k)} \simeq \mathbf{C}_{\mu_h s_h}^{(k)}$ , for each  $k = 1, \dots, N_\tau$ , to the ones which would have been provided by the FOM.

As a matter of fact, from the previous proposition we can also prove an *asymptotic consistency* property, ensuring that the state/parameters estimated through the RB-EnKF converge to the ones estimated by the full-order EnKF, as soon as  $n, m_D \rightarrow N_h$ .

**Corollary 3.** *By assuming that for any RB dimension  $n = 1, \dots, N_h$  and DEIM dimension  $m_D = 1, \dots, N_h$ , there exists  $\epsilon^{(\ell)}(n, m_D) > 0$  such that*

$$\|\mathbf{u}_h^{(\ell)}(\boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu})\| \leq \epsilon^{(\ell)}(n, m_D), \quad \forall \ell = 0, \dots, N_t \quad \forall \boldsymbol{\mu} \in \mathcal{P}$$

and  $\epsilon^{(\ell)}(n, m_D) \rightarrow 0$  for  $n, m_D \rightarrow N_h$ , then it follows that

$$\|\hat{\boldsymbol{\mu}}_n - \hat{\boldsymbol{\mu}}_h\| \rightarrow 0 \quad \|\mathbf{V}\hat{\mathbf{u}}_n - \hat{\mathbf{u}}_h\| \rightarrow 0 \quad \text{for } n, m_D \rightarrow N_h.$$

*Proof.* Since the outputs  $\mathbf{s}_h^{(k)}(\boldsymbol{\mu})$ ,  $\mathbf{s}_n^{(k)}(\boldsymbol{\mu})$  are linear with respect to the solution  $\mathbf{u}(t; \boldsymbol{\mu})$  of the dynamical system, it follows that, for any  $\boldsymbol{\mu} \in \mathcal{P}$ ,

$$\|\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \mathbf{s}_n^{(k)}(\boldsymbol{\mu})\| \leq \Delta t \|\mathbf{H}\| \sum_{K(k-1) \leq \ell \leq K(k)} \epsilon^{(\ell)}(n, m_D) \quad \forall k = 1, \dots, N_\tau. \quad (29)$$

Since the means and the covariance matrices of the quantities of interest are evaluated on different subsets  $\mathcal{P}_h^{(k)}$  and  $\mathcal{P}_n^{(k)}$  for each  $k = 1, \dots, N_\tau$ , let us denote by

$$(\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h} = \frac{1}{N_e} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} \mathbf{s}_n^{(k)}(\boldsymbol{\mu}),$$

$$(\mathbf{C}_{s_n s_n}^{(k)})_{\mathcal{P}_h} = \frac{1}{N_e - 1} \sum_{\boldsymbol{\mu} \in \mathcal{P}_h^{(k-1)}} (\mathbf{s}_n^{(k)}(\boldsymbol{\mu}) - (\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h})(\mathbf{s}_n^{(k)}(\boldsymbol{\mu}) - (\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h})^T$$

the mean and the covariance of the reduced output over the full-order ensemble  $\mathcal{P}_h$ , respectively. Then, we can control the difference between the output means as

$$\|\bar{\mathbf{s}}_h^{(k)} - \bar{\mathbf{s}}_n^{(k)}\| \leq \|\bar{\mathbf{s}}_h^{(k)} - (\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h}\| + \|(\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h} - \bar{\mathbf{s}}_n^{(k)}\|$$

so that, by averaging (29) with respect to  $\mathcal{P}_h$ , we bound the first term as

$$\|\bar{\mathbf{s}}_h^{(k)} - (\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h}\| \leq \Delta t \|\mathbf{H}\| \sum_{K(k-1) \leq \ell \leq K(k)} \epsilon^{(\ell)}(n, m_D) \quad \forall k = 1, \dots, N_\tau.$$

Similarly, the difference between the covariance matrices can be bounded as

$$\|\mathbf{C}_{s_h s_h}^{(k)} - \mathbf{C}_{s_n s_n}^{(k)}\| \leq \|\mathbf{C}_{s_h s_h}^{(k)} - (\mathbf{C}_{s_n s_n}^{(k)})_{\mathcal{P}_h}\| + \|(\mathbf{C}_{s_n s_n}^{(k)})_{\mathcal{P}_h} - \mathbf{C}_{s_n s_n}^{(k)}\|$$

where the first term can be bounded as

$$\|\mathbf{C}_{s_h s_h}^{(k)} - (\mathbf{C}_{s_n s_n}^{(k)})_{\mathcal{P}_h}\| \leq c_{ss}^{(k)} \sum_{K(k-1) \leq \ell \leq K(k)} \epsilon^{(\ell)}(n, m_D) \quad \forall k = 1, \dots, N_\tau,$$

where

$$c_{ss}^{(k)} = 4\Delta t \max_{\boldsymbol{\mu} \in \mathcal{P}^{(k-1)}} (\|\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{s}}_h^{(k)}\| + \|\mathbf{s}_n^{(k)}(\boldsymbol{\mu}) - \bar{\mathbf{s}}_n^{(k)}\|),$$

being, for any couple of random vectors  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ ,  $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2]$ ,

$$\|C_{\mathbf{x}_1 \mathbf{y}_1} - C_{\mathbf{x}_2 \mathbf{y}_2}\| \leq (\|\mathbf{y}_1 - \bar{\mathbf{y}}_1\| + \|\mathbf{x}_2 - \bar{\mathbf{x}}_2\|)(\|\mathbf{x}_1 - \mathbf{x}_2\| + \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2\| + \|\mathbf{y}_1 - \mathbf{y}_2\| + \|\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2\|),$$

where  $C_{\mathbf{x}\mathbf{y}}$  denotes the cross-covariance matrix between  $\mathbf{x}$  and  $\mathbf{y}$ ; see e.g. [6].

Provided that  $\mathcal{P}_h^{(0)} = \mathcal{P}_n^{(0)}$  and assuming that  $\epsilon^{(\ell)}(n, m_D) \rightarrow 0$  for  $n, m_D \rightarrow N_h$ , we have that

$$\|\bar{\mathbf{s}}_h^{(k)} - (\bar{\mathbf{s}}_n^{(k)})_{\mathcal{P}_h}\| \rightarrow 0, \quad \|\mathbf{C}_{s_h s_h}^{(k)} - (\mathbf{C}_{s_n s_n}^{(k)})_{\mathcal{P}_h}\| \rightarrow 0, \quad k = 1, \dots, N_\tau$$

and, consequently,

$$\|\bar{\mathbf{s}}_n^{(k)} - \bar{\mathbf{s}}_n^{(k)}\| \rightarrow 0, \quad \|(\mathbf{C}_{s_n s_n}^{(k)})_{\mathcal{P}_h} - \mathbf{C}_{s_n s_n}^{(k)}\| \rightarrow 0, \quad k = 1, \dots, N_\tau.$$

In the same way, we can conclude that  $\|\mathbf{C}_{u_h s_h}^{(k+1)} - \mathbf{C}_{u_n s_n}^{(k+1)}\|$  and  $\|\mathbf{C}_{\mu_h s_h}^{(k+1)} - \mathbf{C}_{\mu_n s_n}^{(k+1)}\|$  are also controlled by  $\epsilon^{(\ell)}(n, m_D)$ ,  $\ell = Kk, \dots, K(k+1)$ , thus yielding the fact that the right-hand sides of both (20) and (21) go to zero in the limit  $n, m_D \rightarrow N_h$ .  $\square$

## 5 Reduction error model

Using a ROM to evaluate the output of the forward PDE system greatly reduces the cost entailed by the solution of the whole Bayesian inverse problem. Indeed, by simply rewriting the additive error noise model (1), we get

$$\mathbf{s}^{(k)} = \mathbf{s}_n^{(k)}(\boldsymbol{\mu}^*) + \left( \mathbf{s}_h^{(k)}(\boldsymbol{\mu}^*) - \mathbf{s}_n^{(k)}(\boldsymbol{\mu}^*) \right) + \boldsymbol{\varepsilon}_{\text{noise}}, \quad \forall k = 1, \dots, N_\tau; \quad (30)$$

hence, if the reduction error  $\mathbf{s}_h^{(k)}(\boldsymbol{\mu}^*) - \mathbf{s}_n^{(k)}(\boldsymbol{\mu}^*)$  is not negligible with respect to  $\boldsymbol{\varepsilon}_{\text{noise}}$ , the RB-EnKF might yield biased estimates. Therefore, we wish to introduce a statistical model for the reduction error  $\boldsymbol{\varepsilon}_{\text{ROM}}^{(k)}(\boldsymbol{\mu})$ , over each window  $k = 1, \dots, N_\tau$ , such that (30) can be replaced by

$$\mathbf{s}^{(k)} = \mathbf{s}_n^{(k)}(\boldsymbol{\mu}^*) + \hat{\boldsymbol{\varepsilon}}_{\text{ROM}}^{(k)}(\boldsymbol{\mu}^*) + \boldsymbol{\varepsilon}_{\text{noise}} \quad \forall k = 1, \dots, N_\tau \quad (31)$$

and the evaluation of the deterministic quantity  $\mathbf{s}_h(\boldsymbol{\mu}) - \mathbf{s}_n(\boldsymbol{\mu})$ , which would depend on the FOM solution, can thus be avoided.

To compute an estimate of  $\boldsymbol{\varepsilon}_{\text{ROM}}^{(k)}(\boldsymbol{\mu})$ , which will be denoted hereon by  $\hat{\boldsymbol{\varepsilon}}_{\text{ROM}}^{(k)}(\boldsymbol{\mu})$  and to which we refer to as *reduction error model* (REM), we rely on the curve kriging method, a weighted interpolation technique for spatially-distributed functional data (see e.g. [13, 28]).

Let us denote by  $S_{\text{cal}} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{N_{\text{cal}}}\}$  a *calibration set* made by  $N_{\text{cal}}$  parameter vectors, for which we need to determine  $N_{\text{cal}}$  queries to both the ROM and the FOM. We underline that these evaluations have to be performed only once, after the ROM has been built, and before the inversion procedure takes place. In particular, we can choose  $S_{\text{cal}}$  such that  $S_{\text{train}} \subset S_{\text{cal}}$  so that we can take advantage of the snapshots already computed before running the POD, and ensure not to overestimate the reduction error in those training points.

We assume each component  $\{\chi_t^{(j)}(\boldsymbol{\mu})\}$ ,  $j = 1, \dots, s$ , of the reduction error

$$\chi_t(\boldsymbol{\mu}) = \mathbf{H}(\mathbf{u}_h(t; \boldsymbol{\mu}) - \mathbf{u}_n(t; \boldsymbol{\mu})) \in \mathbb{R}^s, \quad t \in (0, T)$$

to be a functional random field, that is, a set of functional random variables indexed by  $\boldsymbol{\mu} \in \mathcal{P}$ , taking values in  $L^2(a, b)$ , with  $(a, b) \subseteq (0, T)$ . For each  $j = 1, \dots, s$ , the curve kriging method provides an estimate of the error over  $(0, T)$ , for any new  $\boldsymbol{\mu}_0 \in \mathcal{P}$ , as a linear combination of the reduction errors computed for the elements in the calibration set  $S_{\text{cal}}$ , that is,

$$\hat{\chi}_t^{(j)}(\boldsymbol{\mu}_0) = \sum_{q=1}^{N_{\text{cal}}} \lambda_q^{(j)}(\boldsymbol{\mu}_0) \chi_t^{(j)}(\boldsymbol{\mu}_q), \quad \boldsymbol{\mu}_0 \in \mathcal{P}, \quad j = 1, \dots, s,$$

where the set of weights  $\{\lambda_q^{(j)}(\boldsymbol{\mu}_0)\}_{q=1}^{N_{\text{cal}}}$  are computed by requiring that  $\hat{\chi}_t^{(j)}$  is the best linear unbiased estimator of  $\chi_t^{(j)}$ , see Appendix B for a detailed construction.

Since we are interested to embed the REM into the Kalman formula for the sequential update of the ensemble on each window  $[\tau^{(k)}, \tau^{(k+1)})$ , we need to build a curve kriging predictor  $\hat{\chi}_t^{(j)}(\boldsymbol{\mu})$  for each  $j = 1, \dots, s$ , on each window  $[\tau^{(k-1)}, \tau^{(k)})$ . As a matter of fact, our REM is given, for any  $\boldsymbol{\mu} \in \mathcal{P}_c^{(k)}$  by integrating over  $[\tau^{(k-1)}, \tau^{(k)})$  the kriging predictor, that is,

$$\hat{\boldsymbol{\varepsilon}}_{\text{ROM}}^{(k)}(\boldsymbol{\mu}) : \hat{\boldsymbol{\varepsilon}}_{\text{ROM},j}^{(k)}(\boldsymbol{\mu}) = \int_{\tau^{(k-1)}}^{\tau^{(k)}} \hat{\chi}_t^{(j)}(\boldsymbol{\mu}) dt. \quad (32)$$

The corresponding trace-variances (see equation (49) and Appendix 2 for the definition of  $\gamma_t$  and  $\eta$ )

$$\hat{\sigma}_{\hat{\chi}_t^{(j)}}^2(\boldsymbol{\mu}) = \sum_{q=1}^{N_{\text{cal}}} \lambda_q^{(j)}(\boldsymbol{\mu}) \int_{\tau^{(k-1)}}^{\tau^{(k)}} \gamma_t(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}\|) - \eta$$

allows to define the (diagonal) covariance matrix

$$\hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k)}(\boldsymbol{\mu}) : (\hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k)})_{jj}(\boldsymbol{\mu}) = \hat{\sigma}_{\hat{\chi}_t^{(j)}}^2(\boldsymbol{\mu}). \quad (33)$$

The proposed REM thus yields an output correction  $\hat{\varepsilon}_{\text{ROM}}^{(k)}$  and an additional contribution  $\hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k)}$  to the Kalman gain – which have indeed to be evaluated for each  $k = 1, \dots, N_\tau$  and upon each ensemble particle – thus leading to the following *corrected Kalman formula* to update the ensemble  $\mathcal{P}_c^{(k)}$ :

$$\begin{bmatrix} \boldsymbol{\mu}_{c,q}^{(k+1)} \\ \mathbf{u}_c^{(k+1)}(\boldsymbol{\mu}_{c,q}^{(k+1)}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{c,q}^{(k)} \\ \mathbf{u}_c^{(k+1)}(\boldsymbol{\mu}_{c,q}^{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mu_c s_c}^{(k+1)} \\ \mathbf{C}_{u_c s_c}^{(k+1)} \end{bmatrix} (\mathbf{\Gamma} + \hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k+1)}(\boldsymbol{\mu}_{c,q}^{(k)}) + \mathbf{C}_{s_c s_c}^{(k+1)})^{-1} (\mathbf{s}^{(k+1)} - \mathbf{s}_c^{(k+1)}(\boldsymbol{\mu}_{c,q}^{(k)})) \quad (34)$$

where  $\mathbf{s}_c^{(k)}(\boldsymbol{\mu})$  represents the *corrected output*, i.e.

$$\mathbf{s}_c^{(k)}(\boldsymbol{\mu}) = \mathbf{s}_n^{(k)}(\boldsymbol{\mu}) + \hat{\varepsilon}_{\text{ROM}}^{(k)}(\boldsymbol{\mu}), \quad \boldsymbol{\mu} \in \mathcal{P};$$

the sample covariance  $\mathbf{C}_{s_c s_c}^{(k)}$  and cross-covariances  $\mathbf{C}_{\mu_c s_c}^{(k)}$ ,  $\mathbf{C}_{u_c s_c}^{(k)}$  are computed as in equations (12), (13) and (14) by substituting  $\mathbf{s}_h$ ,  $\mathbf{u}_h$  and  $\mathcal{P}_h^{(k-1)}$  with  $\mathbf{s}_c$ ,  $\mathbf{u}_c$  and  $\mathcal{P}_c^{(k-1)}$ , respectively.

The use of our REM during the RB-EnKF thus only requires, at each iteration of the filtering procedure, to solve  $s$  linear systems (see equation (45)) of small dimension  $(N_{cal} + 1) \times (N_{cal} + 1)$ , to get the weights  $\{\lambda_q^{(j)}\}_{q=1}^{N_{cal}}$  for each output component  $j = 1, \dots, s$ . A detailed description of the *corrected RB-EnKF* algorithm is finally reported in Algorithm 2. The components appearing in the matrix and the vector of system (45) are evaluated using the so-called semi-variogram function, which is fitted on the calibration set  $S_{cal}$  as shown in Appendix B.

---

#### Algorithm 2 Reduced basis Ensemble Kalman filter procedure

---

- 1: **procedure** RB-EnKF(+REM)
  - 2:   *Initialization*
  - 3:    $\{\mathcal{P}_c^{(0)}, \mathcal{U}_c^{(0)}\} \leftarrow$  sampling  $N_e$  vectors from  $\pi_{\text{prior}}$
  - 4:   **for**  $k = 0 : N_\tau - 1$  **do**
  - 5:     **for**  $j=1 : s$  **do**
  - 6:       sample the empirical semi-variogram  $\{(\delta_m, \hat{\gamma}(\delta_m))\}_{m=1}^M$ , using (46)
  - 7:       fit the parametric semi-variogram model (47) on the sample  $\{(\delta_m, \hat{\gamma}(\delta_m))\}_{m=1}^M$
  - 8:     *Prediction stage:*
  - 9:     **for**  $[\boldsymbol{\mu}, \mathbf{u}_h^{(k)}(\boldsymbol{\mu})]^T \in \{\mathcal{P}_c^{(k)}, \mathcal{U}_c^{(k)}\}$  **do**
  - 10:        $\mathbf{u}_n^{(k+1)}(\boldsymbol{\mu}) \leftarrow$  solve forward problem (16) with initial datum  $\mathbf{u}_n^{(k)}(\boldsymbol{\mu})$
  - 11:       **for**  $j=1 : s$  **do**
  - 12:           $\hat{\varepsilon}_{\text{ROM}}^{(k+1)}(\boldsymbol{\mu}) \leftarrow$  solve linear system (45)
  - 13:           $(\hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k+1)})_{jj} \leftarrow$  evaluate (49)
  - 14:           $\mathbf{s}_c^{(k+1)}(\boldsymbol{\mu}) = \mathbf{s}_n^{(k+1)}(\boldsymbol{\mu}) + \hat{\varepsilon}_{\text{ROM}}^{(k+1)}(\boldsymbol{\mu})$
  - 15:       compute means  $\bar{\mathbf{s}}_c^{(k+1)}$ ,  $\bar{\mathbf{u}}_c^{(k+1)}$ ,  $\bar{\boldsymbol{\mu}}_c^{(k+1)}$
  - 16:       compute covariance  $\mathbf{C}_{s_c s_c}^{(k+1)}$
  - 17:       compute cross-covariances  $\mathbf{C}_{\mu_c s_c}^{(k+1)}$ ,  $\mathbf{C}_{u_c s_c}^{(k+1)}$
  - 18:     *Update stage:*
  - 19:     **for**  $[\boldsymbol{\mu}, \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu})]^T \in \{\mathcal{P}_c^{(k)}\}$  **do**
  - 20:       update each state/parameters particle using (34)
- 

## 5.1 Effectivity of the proposed REM

We observe that the reduction error directly affects the quality of the likelihood function (9) from which we have derived the updating formula of the Kalman filter (see Appendix A). By defining the reduced likelihood as

$$\pi_n \left( \mathbf{s}^{(k+1)} \middle| \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{V}\mathbf{u}_n^{(k+1)} \end{array} \right] \right) = (2\pi)^{(-\frac{s}{2})} |\mathbf{\Gamma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} \|\mathbf{s}^{(k+1)} - \mathbf{s}_n^{(k+1)}\|_{\mathbf{\Gamma}}\right\},$$

and the corrected likelihood as

$$\pi_c \left( \mathbf{s}^{(k+1)} \middle| \left[ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{V}\mathbf{u}_n^{(k+1)} \end{array} \right] \right) = (2\pi)^{(-\frac{s}{2})} |\mathbf{\Gamma} + \hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} \|\mathbf{s}^{(k+1)} - \mathbf{s}_c^{(k+1)}\|_{\mathbf{\Gamma} + \hat{\mathbf{\Gamma}}_{\text{ROM}}^{(k)}}\right\},$$

we can rely on the analysis provided in [27, Section 6] on the Kullback-Leibler (KL) divergence between the likelihood function  $\pi$ ,  $\pi_n$  and  $\pi_c$  at each *prediction-analysis* step  $k = 1, \dots, N_\tau$ . To this end, let us recall the notion of Kullback-Leibler (KL) divergence, which is a measure of the difference between two probability distributions  $\pi_A$  and  $\pi_B$ :

$$D_{KL}(\pi_A||\pi_B) = \int \pi_A(z) \log \left( \frac{\pi_A(z)}{\pi_B(z)} \right) dz.$$

In this case, we would obtain

$$D_{KL}(\pi||\pi_n) = \frac{1}{2} \sum_{j=1}^s \left( \frac{(\mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}) - \mathbf{s}_n^{(k+1)}(\boldsymbol{\mu}))_j^2}{\boldsymbol{\Gamma}_{jj}} \right)$$

and

$$D_{KL}(\pi||\pi_c) = \frac{1}{2} \sum_{j=1}^s \left( \frac{(\mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}) - \mathbf{s}_n^{(k+1)}(\boldsymbol{\mu}) - \hat{\boldsymbol{\epsilon}}_{\text{ROM}}^{(k+1)}(\boldsymbol{\mu}))_j^2}{\boldsymbol{\Gamma}_{jj} + (\hat{\boldsymbol{\Gamma}}_{\text{ROM}}^{(k+1)})_{jj}(\boldsymbol{\mu})} + \frac{\boldsymbol{\Gamma}_{jj}}{\boldsymbol{\Gamma}_{jj} + (\hat{\boldsymbol{\Gamma}}_{\text{ROM}}^{(k+1)})_{jj}(\boldsymbol{\mu})} - 1 - \log \left( \frac{\boldsymbol{\Gamma}_{jj}}{\boldsymbol{\Gamma}_{jj} + (\hat{\boldsymbol{\Gamma}}_{\text{ROM}}^{(k+1)})_{jj}(\boldsymbol{\mu})} \right) \right).$$

Thus, in order to ensure that the KL divergence  $D_{KL}(\pi||\pi_c)$  is smaller than  $D_{KL}(\pi||\pi_n)$ , we require that:

1. the REM correction is effective, that is

$$\mathbb{E}[\|\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \mathbf{s}_c^{(k)}(\boldsymbol{\mu})\|] < \mathbb{E}[\|\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \mathbf{s}_n^{(k)}(\boldsymbol{\mu})\|] \quad \forall k = 1, \dots, N_\tau, \quad (35)$$

2.  $(\hat{\boldsymbol{\Gamma}}_{\text{ROM}}^{(k+1)})_{jj}$  is sufficiently small compared to  $\boldsymbol{\Gamma}_{jj}$ ,  $j = 1, \dots, s$ .

Note that by construction  $\mathbf{s}_h^{(k)}(\boldsymbol{\mu}) - \mathbf{s}_c^{(k)}(\boldsymbol{\mu}) = \mathbf{0}$  and  $\hat{\boldsymbol{\Gamma}}_{\text{ROM}}(\boldsymbol{\mu}) = \mathbf{0}$  for each  $\boldsymbol{\mu} \in S_{cal}$ . Since the noise is prescribed with a fixed covariance, the ROM and the REM construction can be suitably performed in order to ensure both the previous assumptions.

**Remark 4.** *Since the EnKF is based on a finite ensemble of particles, the distributions  $\pi$ ,  $\pi_n$  and  $\pi_c$  are only approximated in the EnKF updating formula. It is sufficient to consider a large ensemble in order to avoid the propagation of additional sources of error.*

Note that the updating formula (34) could be derived using the corrected likelihood distribution  $\pi_c$  instead of  $\pi$  following the approach proposed in Appendix A. As a consequence, the updating formula (34) is a good approximation of (15) if the REM is effective. Moreover, under the two previous assumptions the corrected particle ensemble  $\mathcal{P}_c^{(k)}$  (and respectively  $\mathcal{U}_c^{(k)}$ ) is considerably closer to  $\mathcal{P}_h^{(k)}$  ( $\mathcal{U}_h^{(k)}$ ) than  $\mathcal{P}_n^{(k)}$  ( $\mathcal{U}_n^{(k)}$ ). If we assume at each step  $k = 1, \dots, N_\tau$  to use as initial datum (a priori information) the full-order ensemble  $\mathcal{P}_h^{(k)}$  ( $\mathcal{U}_h^{(k)}$ ), it is possible to prove that

$$\mathbb{E}[D_{KL}(\pi^{(k)}||\pi_c^{(k)})] < \mathbb{E}[D_{KL}(\pi^{(k)}||\pi_n^{(k)})]$$

where  $\pi^{(k)}$ ,  $\pi_n^{(k)}$  and  $\pi_c^{(k)}$  denote respectively the full-order, the reduced-order and the corrected posterior distribution obtained by substituting the respective likelihood function  $\pi$ ,  $\pi_n$  and  $\pi_c$  in (10); for further details see [27, Section 6.2].

As a matter of fact  $\hat{\boldsymbol{\epsilon}}_{\text{ROM}}$  and  $\hat{\boldsymbol{\Gamma}}_{\text{ROM}}$  depend on either the number of basis functions and on the calibration set dimension. For this reason, in the numerical results we compare the errors  $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|$  and  $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_c\|$  between the sample means and the errors between the relative covariance matrices  $\|(\mathbf{C}_{\boldsymbol{\mu}_h, \boldsymbol{\mu}_h})^{1/2} - (\mathbf{C}_{\boldsymbol{\mu}_n, \boldsymbol{\mu}_n})^{1/2}\|$  and  $\|(\mathbf{C}_{\boldsymbol{\mu}_h, \boldsymbol{\mu}_h})^{1/2} - (\mathbf{C}_{\boldsymbol{\mu}_c, \boldsymbol{\mu}_c})^{1/2}\|$  over the ensemble obtained with the full-order Kalman filter ( $\mathcal{P}_h$ ), the RB-EnKF ( $\mathcal{P}_n$ ) and the corrected RB-EnKF ( $\mathcal{P}_c$ ) on varying the the number of basis functions and on the calibration set dimension.

## 6 Numerical results

We present some numerical results exploiting the proposed RB-EnKF procedure for the identification of unknown parameters/fields in a FitzHugh-Nagumo and a Fisher-Kolmogorov model<sup>1</sup>; the dynamics described by those two nonstationary nonlinear diffusion-reaction PDEs involve traveling front.

### 6.1 Test case 1

We consider the FitzHugh-Nagumo (FN) equations [10], which model the activation/deactivation dynamics of an excitable system, e.g., a neuron or a cardiac cell. In particular, we consider the test case proposed in [3]: given the parameter vector  $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3]^T$ ,  $\forall t \in (0, T)$ , find the couple  $[u(t; \boldsymbol{\mu}), w(t; \boldsymbol{\mu})]$ ,  $x \in \Omega = (0, 1)$ , such that:

$$\begin{cases} \nu u_t(t; \boldsymbol{\mu}) = \nu^2 u_{xx}(t; \boldsymbol{\mu}) + \mathcal{N}(u(t; \boldsymbol{\mu})) - w(t; \boldsymbol{\mu}) + \mu_2 & x \in \Omega, t \in (0, T) \\ w_t(t; \boldsymbol{\mu}) = \mu_1 u(x, t) - \gamma u(x, t) + \mu_2 & x \in \Omega, t \in (0, T) \\ u_x(t; \boldsymbol{\mu}) = -50000t^3 e^{-15t} & x = 0, t \in (0, T) \\ u_x(t; \boldsymbol{\mu}) = 0 & x = 1, t \in (0, T) \\ u(0; \boldsymbol{\mu}) = 0, \quad w(0, \boldsymbol{\mu}) = 0 & x \in \Omega; \end{cases} \quad (36)$$

we define a cubic nonlinear term  $\mathcal{N}(u) = u(u - 0.1)(1 - u)$ , and set  $\gamma = 2$ ,

$$\epsilon = 0.015(1 - \mu_3) \exp\left(-\frac{(x - 0.6)^2}{0.04}\right).$$

The semi-discretized FE approximation of problem (36) based on a partition of  $\Omega$  into 1024 elements and linear finite elements yields an ODE system which can be written under the form (2). By considering a time discretization based on  $N_t = 800$  time-steps and the implicit Euler method, we obtain the dynamical system under the form (3). Then, we consider as output the vector

$$\mathbf{s}_h^{(k+1)} = \begin{bmatrix} \int_{\tau^{(k)}}^{\tau^{(k+1)}} \mathbf{u}_h(t; \boldsymbol{\mu})|_{x=0} dt \\ \int_{\tau^{(k)}}^{\tau^{(k+1)}} \mathbf{u}_h(t; \boldsymbol{\mu})|_{x=1} dt \end{bmatrix} \in \mathbb{R}^2 \quad \forall k = 0, \dots, N_\tau - 1.$$

The goal of the inverse problem is to estimate  $\boldsymbol{\mu}^*$  from a noisy output measurement  $\mathbf{s} = \mathbf{s}_h(\boldsymbol{\mu}^*) + \boldsymbol{\varepsilon}_{noise}$ , with  $\boldsymbol{\varepsilon}_{noise} \sim \mathcal{N}(0, \sigma^2 I)$  exploiting the RB-EnKF procedure detailed in the paper and comparing the results obtained with the full-order EnKF. We take a Gaussian prior, so that  $\boldsymbol{\mu} \in \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}, \Sigma_{\text{prior}})$ , with:

$$\boldsymbol{\mu}_{\text{prior}} = \begin{bmatrix} 0.7 \\ 0.07 \\ 0.76 \end{bmatrix}, \quad \Sigma_{\text{prior}} = \begin{bmatrix} 0.0004 & 0 & 0 \\ 0 & 0.00009 & 0 \\ 0 & 0 & 0.004 \end{bmatrix}.$$

The discretized solution  $[\mathbf{u}_h, \mathbf{w}_h]^T$  of the forward problem for the optimal vector of parameters  $\boldsymbol{\mu}^* = [0.6331, 0.0985, 0.7197]$  is represented in Fig. 2. This is hereon considered as the *true* parameter vector value, which generate the data  $\mathbf{s}$ .

We first solve the inverse problem with the *full-order* EnKF starting from data measurements with different noise levels, in particular by considering  $\sigma = 5\sigma_0$  and  $\sigma = \sigma_0$ , with  $\sigma_0 = 0.033$ . By looking at the behavior of the estimate  $\hat{\boldsymbol{\mu}}_h$  in Fig. 3 for each component of the parameter vector, we observe that a faster convergence of the estimate to the *true* parameter value during the inversion procedure is achieved in the case of a smaller standard deviation ( $\sigma_0$  with respect to  $5\sigma_0$ ) on the noise.

Next, we compare the solutions of the inverse problem obtained by varying the window length  $\Delta\tau$  and the noise standard deviation (see Fig. 4) taking  $m_D = 15$  DEIM elements and  $n = [7, 11, 15]$  basis function on the RB approximation. As expected, the estimates improve if both the noise

<sup>1</sup>Computations have been run on a laptop with a 2,2 GHz Intel Core i7 processor and 8 GB of RAM.

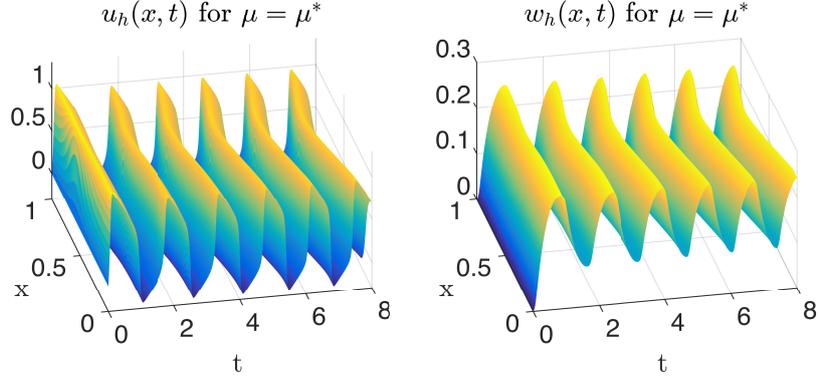


Figure 2: FE approximation of the forward problem for  $\mu = \mu^*$ .

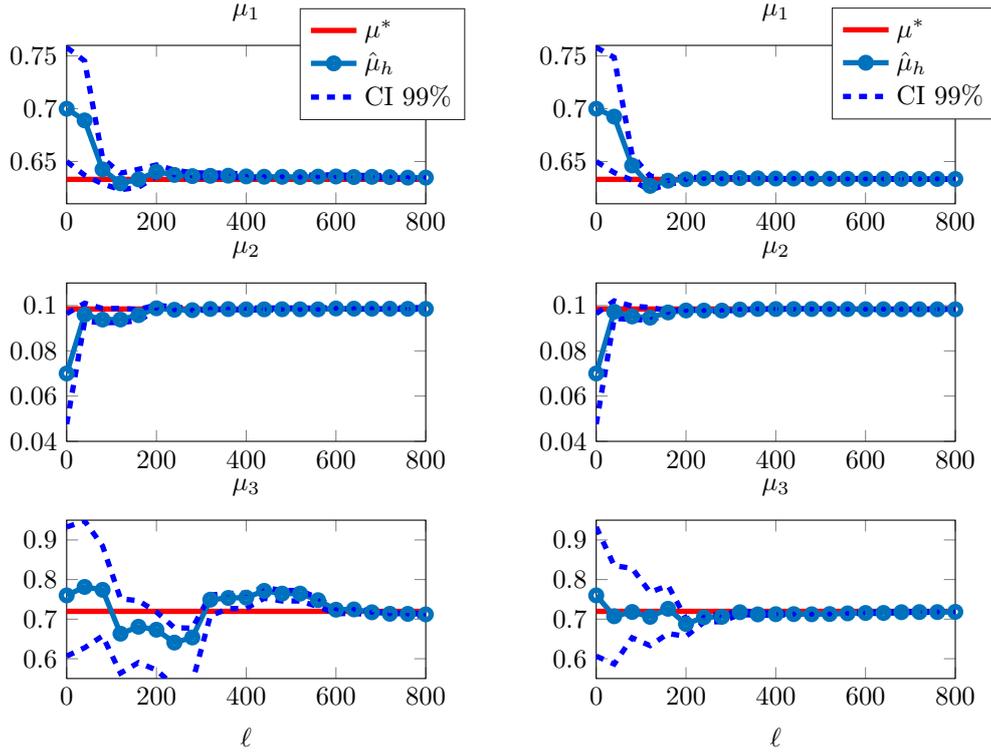


Figure 3: Comparison between  $\mu^*$  and  $\{\hat{\mu}_h^{(k)}\}_{k=1}^{N_\tau}$  for  $\sigma = 5\sigma_0$  (left)  $\sigma = \sigma_0$  (right), using  $N_{ens} = 500$  particles. The lower the noise level, the faster the convergence to  $\mu^*$  during the inverse procedure (see e.g. the figures on the third line, related to the component  $\mu^3$ ).

and  $\Delta\tau$  decrease. While the former is a datum of the problem, the latter can be properly tuned (and reduced) to improve the estimation of the quantities of interest by slightly increasing the computational costs of the inversion procedure.

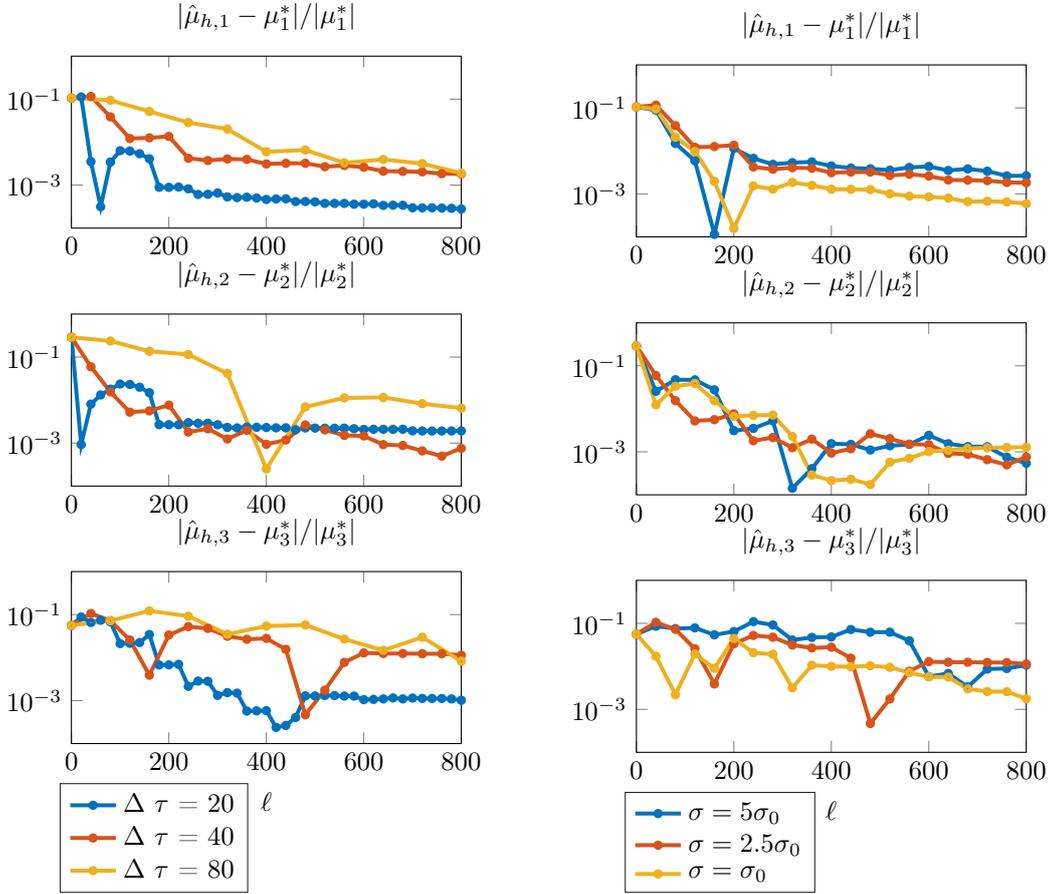


Figure 4: Comparison of the relative error on estimated parameter components  $|\mu^* - \hat{\mu}_h^{(k)}|/|\mu^*|$  on varying the updating time-discretization (left) and on varying the noise standard deviation (right). By considering different values of the updating time-step length  $\Delta\tau$  it is possible to improve the accuracy of the estimated parameter values.

Then, we consider the solution of the inverse problem by exploiting the RB-EnKF not including the REM correction. By looking at Fig. 5, we notice that the estimate  $\hat{\mu}_n$  is not as accurate as the FOM estimate  $\hat{\mu}_h$ , except for the case of  $n = 15$  and a noise variance of  $5\sigma_0$ , because of the propagation of the reduction error on the solution (see Fig. 5) and, consequently, on the measured outputs  $\mathbf{s}_n$ . These numerical results empirically confirm the theoretical findings of Proposition 2 and Corollary 3 we have proven in Sect. 4.3.

The use of a REM thus proves to be mandatory in order to improve the accuracy of the estimates when relying on the RB-EnKF: as a matter of facts the proposed REM based on curve kriging allows us to improve the accuracy of parameter identification in our RB-EnKF even by two orders of magnitude in some cases, as shown in Fig. 7.

More detailed results can be found in Table 1: for a noise level  $\sigma = \sigma_0$ , the ROM affects the accuracy of the identification for every choice of  $n$ , while for higher noise levels the estimation error can be much smaller at least for larger RB dimensions (see e.g. the results obtained for  $n = 15$ ). This is motivated by the fact that the reduction error for  $n = 15$  is considerably small, as we can observe in Fig. 7.

We recall that REM construction is performed only once after the ROM has been built. Given the reduction error  $\chi_t(\mu)$  for each  $\mu \in S_{cal}$ , we check the assumptions of the functional kriging interpolation (see Appendix B). As shown in Figure 8 for the case  $N_{train} = 80$  and  $n = 11$ , the

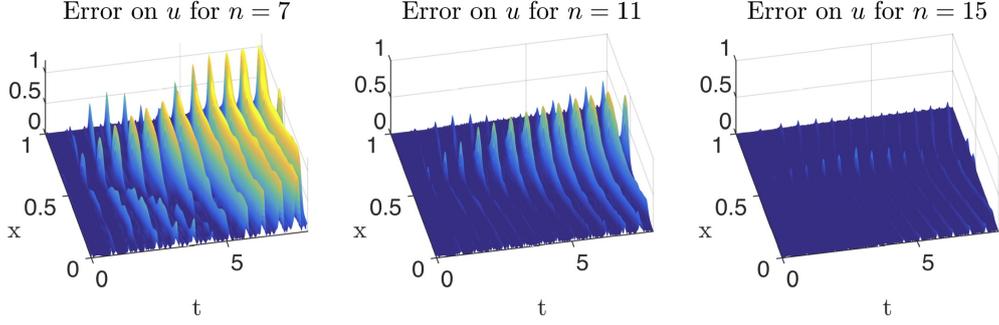


Figure 5: Error  $|\mathbf{u}_n(t; \boldsymbol{\mu}^*) - \mathbf{u}_n(t; \boldsymbol{\mu}^*)|$  between the FOM and the ROM for different choices of the RB dimension  $n = 7, 11, 15$ .

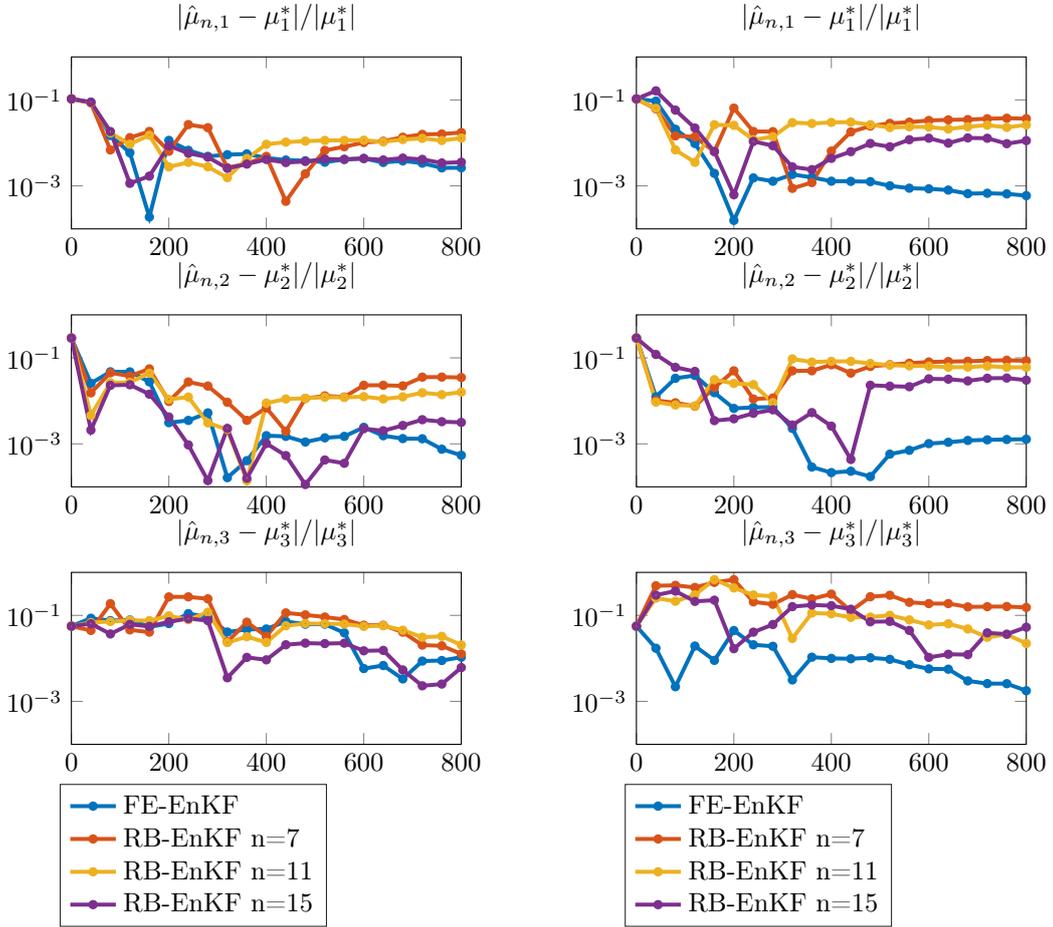


Figure 6: Comparison of the relative error on estimated parameter components  $|\mu^* - \hat{\mu}_n^{(k)}|/|\mu^*|$  on varying the RB dimension  $n$  with  $|S_{train}| = 80$ . The performance of the filters clearly depends on the relation between the ROM error  $\mathbf{s}_h(\boldsymbol{\mu}) - \mathbf{s}_n(\boldsymbol{\mu})$  and the noise standard deviation  $\sigma$ : the error in the estimated parameter component is around 10% in the case where  $\sigma = \sigma_0$ .

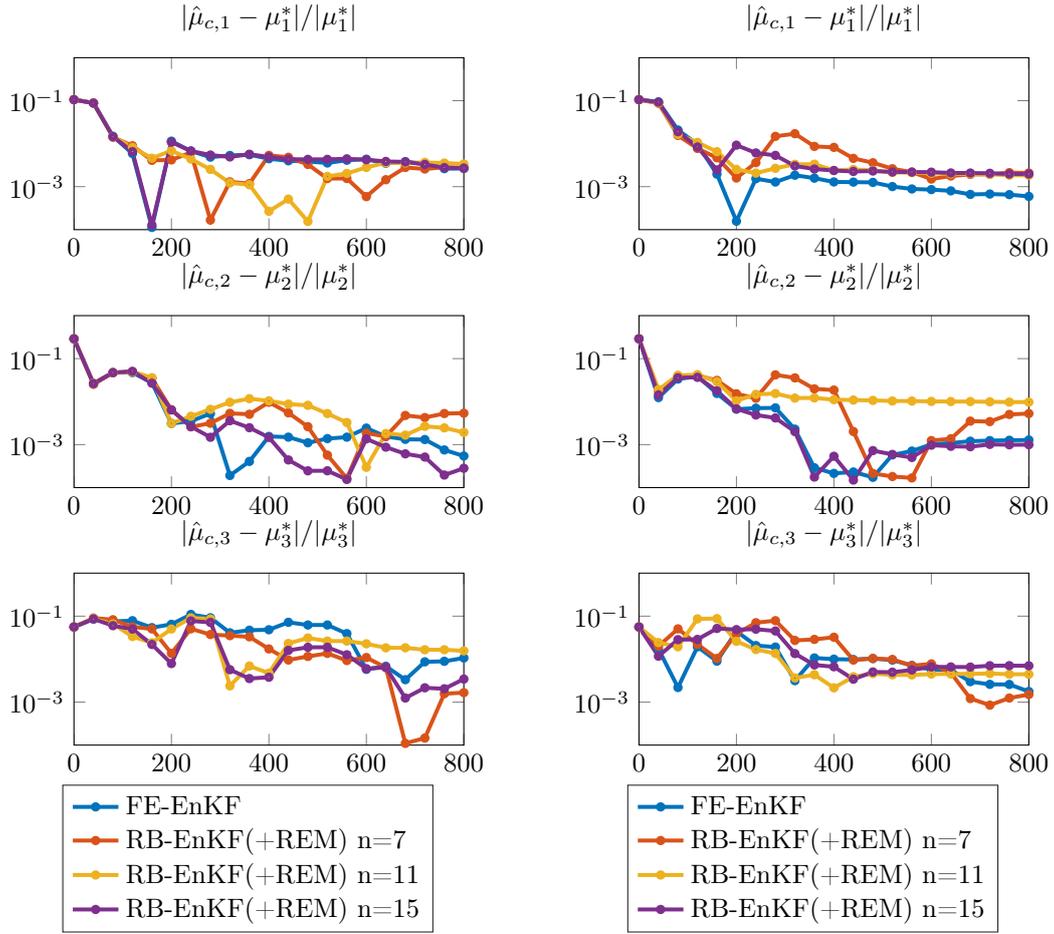


Figure 7: Comparison of the relative error on estimated parameter components  $|\mu^* - \hat{\mu}_c^{(k)}|/|\mu^*|$  obtained by varying the dimension of the RB space for the corrected RB-EnKF with  $S_{cal} = S_{train}$  and  $|S_{train}| = 80$ . A noise variance of  $5\sigma_0$  (left) or  $\sigma_0$  (right) is considered,  $\sigma_0 = 0.033$ . Relying on the proposed REM we obtain a relevant improvement of the estimation in both cases.

correlation between errors shows a dependency with respect to the parameter location: parameters with small lag between each other present a smaller variability with respect to parameters with a larger lag. Then we estimated the empirical semi-variogram  $\{\hat{\gamma}(\delta_1), \dots, \hat{\gamma}(\delta_8)\}$  using (46) at 8 discrete lags  $\{\delta_1, \dots, \delta_8\}$  for each component of the output and on each window  $(\tau^{(k)}, \tau^{(k+1)})$ . Through these estimated values the spherical semi-variogram model (47) is fitted and then used to compute the corresponding matrix of the linear system (45). An example of empirical semi-variograms and relative semi-variograms model is presented in Figure 9.

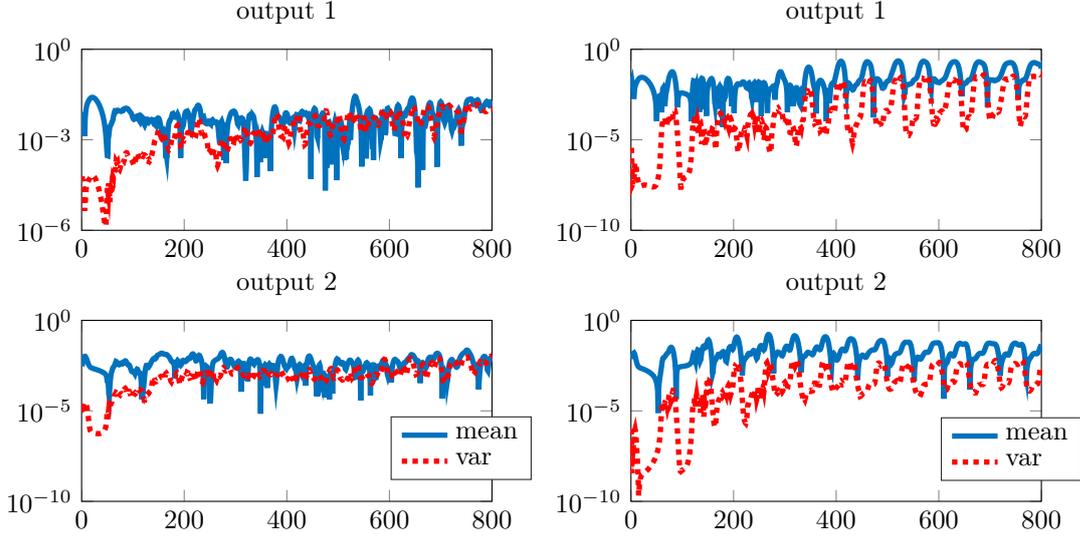


Figure 8: Sample means and covariances of the errors on both output components evaluated over samples  $N(\delta_1)$  (left),  $N(\delta_2)$  (right),  $\delta_1 \ll \delta_2$ .

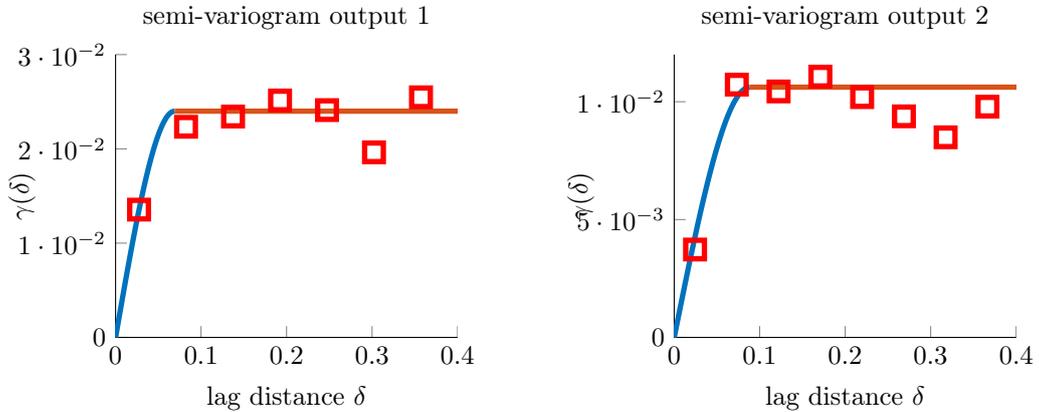


Figure 9: Estimated semi-variogram  $\{\hat{\gamma}(\delta_1), \dots, \hat{\gamma}(\delta_8)\}$  (red squares) and fitted spherical model  $\gamma$  for the two output components on the time interval  $(a, b) = (0, T)$

The use of REM allows to improve the accuracy of  $\hat{\mu}_c$  in any case, when both the RB dimension  $n$  and the noise level  $\sigma$  vary. In Fig. 7 we show the results obtained constructing a REM with  $N_{train} = 80$  and  $N_{cal} = 240$  samples.

The quality of the REM yields significant improvements not only on the estimated means  $\hat{\mu}_c$ , but also on the covariance matrix of the parameter ensemble  $\mathbf{C}_{\mu\mu}^{(N_\tau)}$ . If we compare the errors  $\|\hat{\mu}_h - \hat{\mu}_n\|$  and  $\|\hat{\mu}_h - \hat{\mu}_c\|$  between the estimates obtained with the full-order Kalman filter ( $\hat{\mu}_h$ ), the RB-EnKF ( $\hat{\mu}_n$ ) and the corrected RB-EnKF ( $\hat{\mu}_c$ ), we find that  $\|\hat{\mu}_h - \hat{\mu}_c\|$  is smaller than  $\|\hat{\mu}_h - \hat{\mu}_n\|$  in all the considered cases, differing in some cases by more than two orders of magnitude. Even more, also the error  $\|(\mathbf{C}_{\mu_h, \mu_h}^{(N_\tau)})^{1/2} - (\mathbf{C}_{\mu_c, \mu_c}^{(N_\tau)})^{1/2}\|$  between the square roots of covariance matrices

is considerably smaller than the error on the mean  $\|\hat{\boldsymbol{\mu}}_h^{(N_\tau)} - \hat{\boldsymbol{\mu}}_c^{(N_\tau)}\|$ , as we can observe in Table 1. In other words, the correction introduced by the REM is able to correct the bias yielded by the propagation of the reduction error, without modifying substantially the distribution of the ensemble particles. This means that the ensemble  $\mathcal{P}_c^{(N_\tau)}$  resulting from the application of the corrected RB-EnKF is closer to  $\mathcal{P}_h^{(N_\tau)}$ , the ensemble given by the full-order EnKF, than  $\mathcal{P}_n^{(N_\tau)}$ , the ensemble given by the RB-EnKF, as we have proven in Sect. 5.1.

	$\sigma$	$n = 7$	$n = 11$	$n = 15$
	$5\sigma_0$	0.0442 ( $1.49 \cdot 10^{-4}$ )	0.0365 ( $2.01 \cdot 10^{-4}$ )	0.0054 ( $3.11 \cdot 10^{-5}$ )
RB-Enkf	$2.5\sigma_0$	0.1594 ( $6.11 \cdot 10^{-5}$ )	0.0488 ( $2.37 \cdot 10^{-4}$ )	0.0070 ( $3.47 \cdot 10^{-5}$ )
	$\sigma_0$	0.1779 ( $4.09 \cdot 10^{-5}$ )	0.0708 ( $2.59 \cdot 10^{-5}$ )	0.0613 ( $2.04 \cdot 10^{-5}$ )
	$5\sigma_0$	0.0148 ( $3.37 \cdot 10^{-4}$ )	0.0265 ( $1.01 \cdot 10^{-3}$ )	0.0073 ( $2.25 \cdot 10^{-4}$ )
RB-Enkf(+REM)	$2.5\sigma_0$	0.0175 ( $1.24 \cdot 10^{-4}$ )	0.0226 ( $9.63 \cdot 10^{-4}$ )	0.0117 ( $2.84 \cdot 10^{-4}$ )
	$\sigma_0$	0.0058 ( $1.95 \cdot 10^{-4}$ )	0.0108 ( $5.85 \cdot 10^{-4}$ )	0.0059 ( $2.82 \cdot 10^{-4}$ )

Table 1: Comparison of the error  $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|$  ( $\|(\mathbf{C}_{\boldsymbol{\mu}_h, \boldsymbol{\mu}_h}^{(N_\tau)})^{1/2} - (\mathbf{C}_{\boldsymbol{\mu}_n, \boldsymbol{\mu}_n}^{(N_\tau)})^{1/2}\|$ ) and  $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_c\|$  ( $\|(\mathbf{C}_{\boldsymbol{\mu}_h, \boldsymbol{\mu}_h}^{(N_\tau)})^{1/2} - (\mathbf{C}_{\boldsymbol{\mu}_c, \boldsymbol{\mu}_c}^{(N_\tau)})^{1/2}\|$ ) on varying the dimension  $n$  of the RB space and noise variance  $\sigma$ . Our REM considerably improves the accuracy of the estimated parameter:  $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|$  decreases by an order of magnitude for  $n = 7$  whereas the error on the covariance matrices  $\|(\mathbf{C}_{\boldsymbol{\mu}_h, \boldsymbol{\mu}_h}^{(N_\tau)})^{1/2} - (\mathbf{C}_{\boldsymbol{\mu}_n, \boldsymbol{\mu}_n}^{(N_\tau)})^{1/2}\|$  is still negligible with respect to the error  $\|\hat{\boldsymbol{\mu}}_h - \hat{\boldsymbol{\mu}}_n\|$ .

		$N_{cal} = 0$	$N_{cal} = 24$	$N_{cal} = 80$	$N_{cal} = 240$
$N_{train} = 24$	$n = 7$	0.6669	0.1490	0.0250	0.0361
	$n = 11$	0.0465	0.0337	0.0152	0.0140
	$n = 15$	0.0385	0.0118	0.0161	0.0082
$N_{train} = 80$	$n = 7$	0.1594	–	0.0218	0.0175
	$n = 11$	0.0488	–	0.0398	0.0226
	$n = 15$	0.0070	–	0.0075	0.0117

Table 2: Comparison of the error  $\|\boldsymbol{\mu}^* - \hat{\boldsymbol{\mu}}_c\|/\|\hat{\boldsymbol{\mu}}\|$  on varying the dimension  $N_{train} = |S_{train}|$ ,  $N_{cal} = |S_{cal}|$  of the training set and of the calibration set. The error decreases as soon as the calibration and the training sample have a large dimension. The case with  $N_{cal} < N_{train}$  is meaningless, since we would ignore part of the already computed data within the training set.

As we can observe in Table 3, building a RB approximation of small dimension  $n$  over a training set with dimension  $N_{cal} = 24$  (resp.  $N_{cal} = 80$ ) requires an *offline* CPU time of 16 *min* (resp. 53 *min*), which is small compare to the 387 *min* requested by the full-order EnKF procedure. In this setting, the calibration of the REM over sets of comparable dimension ( $N_{cal} = 24$  and  $N_{cal} = 80$ ) can be performed in few seconds. On the other hand, considering a calibration sample of large dimension  $N_{cal} = 240$  yields better results in terms of estimation accuracy, however entailing a remarkable increase of the calibration costs. The solution of the inverse problem using the corrected RB-EnKF entails only 11 *min*: by comparing the whole procedures, in the worst case scenario we are saving 219 *min*, i.e. more than the 55% of the total computational cost. We also pointed out that the computational saving is even larger if more than one inverse problem has to be solved, for instance on varying the noisy data  $\mathbf{s}$ : in this case the basis computation and the calibration must not be run again, and the only additional costs are given by the execution of the filtering procedure.

## 6.2 Test case 2

As second test case, we consider the Fisher-Kolmogorov-Petrovski-Piskunov (FKPP) reaction-diffusion equation, which models the dynamics of patterns in reactive media e.g. arising in combustion, spreading of epidemics and transport of chemicals in cells (see, e.g. [39, 29] for further details). Here we consider a two dimensional domain  $\Omega \subset \mathbb{R}^2$  and formulate the forward problem

	RB			FE
	$n = 7$	$n = 11$	$n = 15$	$N_h = 1024$
dof reduction	99.3%	98.9%	98.5%	0%
Forward solution	0.6 s	0.7 s	0.8 s	40 s
$ S_{train}  = 24$				
ROM construction	16 min	16 min	16 min	
REM cal. $N_{cal} = 24$	15.1 s	17.4 s	18.73 s	
REM cal. $N_{cal} = 80$	37 min	37.1 min	37.3 min	
REM cal. $N_{cal} = 240$		139.2 min	139.3 min	139.4 min
$ S_{train}  = 80$				
ROM construction	53 min	53 min	53 min	
REM cal. $N_{cal} = 24$	15.1 s	17.4 s	18.73 s	
REM cal. $N_{cal} = 80$	49.8 s	53.8 s	59.7 s	
REM cal. $N_{cal} = 240$		103.3 min	103.4 min	103.5 min
Inverse problem				
EnKF time	6 min	6.3 min	6.9 min	387 min
EnKF REM $N_{cal} = 24$	7.7 min	8.2 min	8.6 min	
EnKF REM $N_{cal} = 80$	8.9 min	9 min	9.4 min	
EnKF REM $N_{cal} = 240$	10.9 min	11.1 min	11.4 min	

Table 3: Test case 1: computational performances of the proposed framework (offline construction and online inversion stages) and comparison with the FOM case.

as: find  $u = u(t; \boldsymbol{\mu})$  s.t.

$$\begin{cases} \frac{\partial u}{\partial t} - \operatorname{div}(\nu(\mathbf{x}; \boldsymbol{\mu}) \nabla u) + \mathcal{N}(u) = 0 & \mathbf{x} \in \Omega, t \in (0, T) \\ \nabla u(t; \boldsymbol{\mu}) \cdot \mathbf{n} = 0 & \mathbf{x} \in \partial\Omega, t \in (0, T) \\ u(0; \boldsymbol{\mu}) = e^{-((x_1 - 1.5)^2 + 50x_2^2)} & \mathbf{x} \in \Omega, \end{cases} \quad (37)$$

where  $\mathcal{N}(u) = 75u(1 - u)$  and the spatial domain  $\Omega$  is given

$$\Omega = \left\{ (\rho, \vartheta) \in (1, 1.5) \times \left(0, \frac{\pi}{2}\right) : \rho = \sqrt{x_1^2 + x_2^2}, \vartheta = \arctan\left(\frac{x_2}{x_1}\right) \right\}.$$

This particular shape of  $\Omega$  imposes a preferential propagation of the front modeled by (37) along the tangential component of the arc. The semi-discretized FE approximation of problem (37) based on a partition of  $\Omega$  using  $N_h = 2768$  mesh nodes and linear finite elements yields a system of ODEs like the one in (2). By considering a partition of the time interval  $(0, T)$  into  $N_t = 140$  time-steps ( $\Delta t = 1.1 \cdot 10^{-3}$ ) and the implicit Euler method, we obtain the dynamical system (3). Then, we consider eight outputs of the form:

$$\mathbf{s}_j(t) = \int_{\Omega} \frac{1}{0.05\pi} \exp\left(-\frac{(x_1 - x_{cj})^2 + (x_2 - y_{cj})^2}{2(0.025)^2}\right) u(t; \boldsymbol{\mu}) d\Omega \quad j = 1, \dots, 8,$$

being  $(x_{cj}, y_{cj}) = (\rho \cos \theta, \rho \sin \theta)$  given by all the possible combinations between  $\rho = \{1, 1.5\}$  and  $\theta = \{\pi/6, \pi/4, \pi/3, \pi/2\}$ . These eight components of the output vector represent approximated pointwise measurements of the quantities of interest  $u(t; \boldsymbol{\mu})$ .

To show the reliability of our procedure in the solution of large-scale parameter identification problems, we consider a parametric field description of the diffusion coefficient  $\nu$  appearing in (37) of the form

$$\nu(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{1} + \sum_{i=1}^d \mu_i \sqrt{\lambda_i} \boldsymbol{\xi}_i \quad (38)$$

where  $\mu_i$ ,  $1 \leq i \leq d$  play the role of identifiable parameters, and  $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_d$  are the  $d$  most relevant (independent) eigenmodes corresponding to the largest eigenvalues of the covariance matrix

$C$ , whose components are chosen as

$$C_{ij} = a \exp\left(-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|}{2b^2}\right) + c\delta_{ij} \quad \forall i, j = 1, \dots, N_h,$$

being  $a, b, c > 0$  and  $\{\mathbf{x}^{(i)}\}_{i=1}^{N_h}$  the nodes of the computational mesh. The parametric field (38) is nothing but the Karhunen-Loève expansion (up to term  $d$ ) of the Gaussian random field  $\nu \sim \mathcal{N}(\mathbf{0}, C)$ , where the prescribed covariance matrix  $C$  describes the spatial smoothness of the random field  $\nu$ .

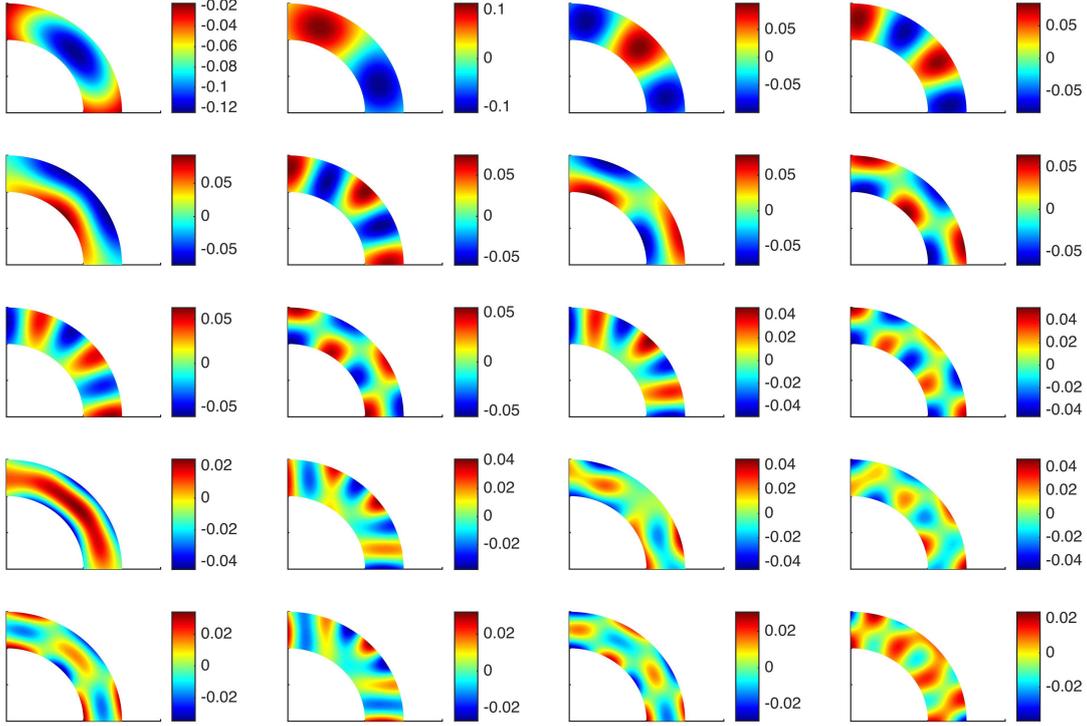


Figure 10: First  $d = 20$  most relevant field modes of the Karhunen-Loève expansion of  $\nu(\mathbf{x}, \boldsymbol{\mu})$ .

The goal of the inverse problem is to estimate  $\boldsymbol{\mu}^*$  from noisy data with  $\varepsilon_{\text{noise}} \sim \mathcal{N}(0, \sigma^2 I)$ . The prior distribution of  $\boldsymbol{\mu}$  is assumed to be Gaussian,  $\boldsymbol{\mu} \in \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}, \Sigma_{\text{prior}})$ , with  $\boldsymbol{\mu}_{\text{prior}} = \mathbf{0}$  and  $\Sigma_{\text{prior}} = \mathbf{I}$ . The reference field  $\nu(\mathbf{x}; \boldsymbol{\mu}^*)$  and the solution of the inverse problem  $\nu(\mathbf{x}; \hat{\boldsymbol{\mu}}_h)$  obtained with the full-order EnKF, relying on the FOM, are represented in Fig. 11.

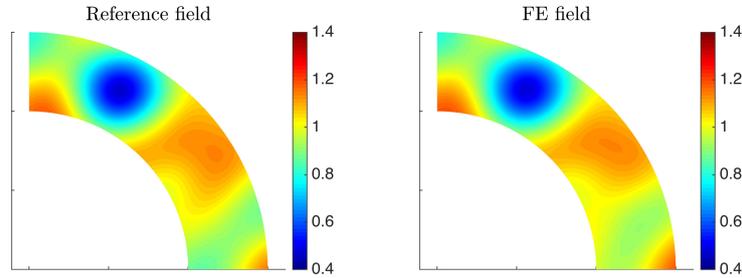


Figure 11: Reference (left) and estimated (right) field using the full-order EnKF.

In this case, the identification problem is even more challenging than in the test case 1: the inverse problem solution with the full-order EnKF requires more than 38 hours of CPU time and constructing the ROM is very expensive due to the higher dimension of the parameter space and

the complex nonlinear dynamics. For these reasons, the REM will play a key role in balancing performance and accuracy of the proposed framework. Therefore, we compare the results obtained by solving the inverse problem when relying on a reduced EnKF taking  $m_D = 300$  DEIM elements,  $n = [115, 150, 185]$  basis functions and by considering a REM built upon a calibration sample  $S_{cal} = S_{train}$  of dimension  $N_{train} = 420$ . As shown in Fig. 12, the uncorrected RB-EnKF fails in the identification of the unknown field for  $n = [115, 150]$ , that is, if the RB approximation is too coarse. For each choice of  $n$ , we recover more accurate reconstructions of the field using the corrected RB-EnKF with respect to the uncorrected procedure (the error  $|\nu(\mathbf{x}; \hat{\boldsymbol{\mu}}_c) - \nu(\mathbf{x}; \hat{\boldsymbol{\mu}}_h)|$  is reduced by at least an order of magnitude with respect to  $|\nu(\mathbf{x}; \hat{\boldsymbol{\mu}}_n) - \nu(\mathbf{x}; \hat{\boldsymbol{\mu}}_h)|$ , without extra-costs due to the calibration procedure. As shown in Table 4, the ROM construction requires several hours to be performed, but the online costs of the reduced EnKF are considerably smaller with respect to ones obtained using the full-order EnKF. The extra costs of the corrected RB-EnKF are negligible with respect to the entire procedure: this fact, as well as the improved accuracy in the estimation, clearly motivate the introduction of a REM in the solution of complex inverse problems. We finally point out that developing efficient ROMs for problems involving traveling fronts is still an open issue; we remark that the proposed framework to solve Bayesian inverse problems is rather general and can be integrated within any choice of surrogate or reduced-order models.

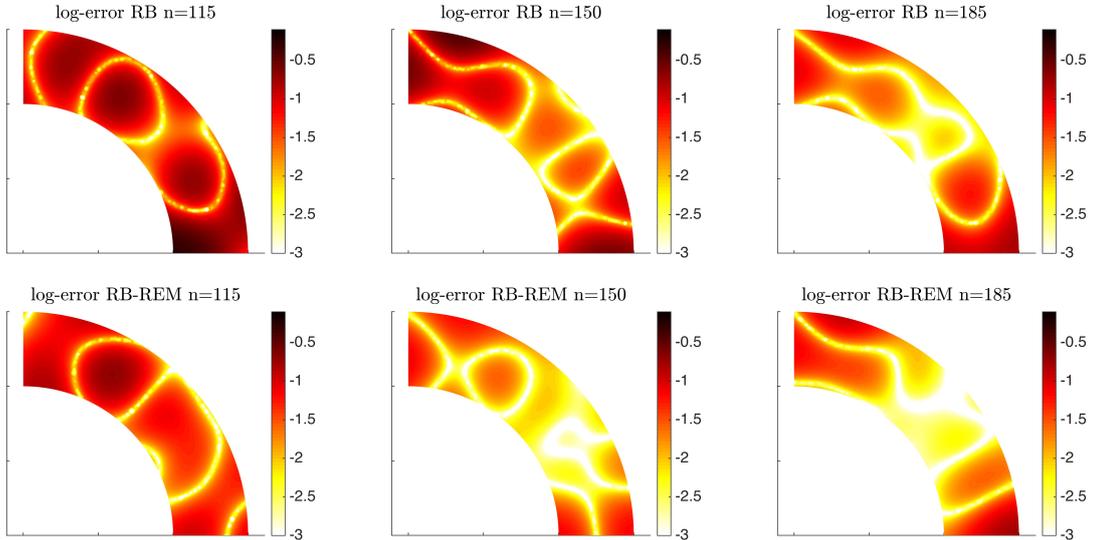


Figure 12: Errors between reference and identified fields for different ROM dimension  $n = 115, 150, 185$ , without (top) and with (bottom) REM correction. This latter clearly yields a relevant improvement in the estimated field.

	RB			FE
	$n = 115$	$n = 150$	$n = 185$	$N_h = 2768$
dof reduction	95.8%	94.5%	93.3%	0%
Forward solution	2.7 s	3.9 s	5 s	279 s
$ S_{train}  = 420$				
ROM construction	31 h	31 h	31 h	
Inverse problem				
EnKF time	25 m	34 m	46 m	38 h 50 m
EnKF REM $N_{cal} = 420$	54 m	62 m	76 m	

Table 4: Test case 2: computational performances of the proposed framework (offline construction and online inversion stages) and comparison with the FOM case.

## Conclusions

The RB-EnKF approach proposed in this paper allows to speed up the solution of state/parameter identification problems without affecting the accuracy of the computed estimates. Since the EnKF requires repetitive evaluations of solutions (and outputs) of the nonlinear dynamical system, in order to reduce the cost of the inversion procedure we can replace the forward map  $\boldsymbol{\mu} \rightarrow \mathbf{s}_h(t; \boldsymbol{\mu})$  with the less expensive one  $\boldsymbol{\mu} \rightarrow \mathbf{s}_n(t; \boldsymbol{\mu})$  obtained using a RB method instead of a full-order model. Our numerical results confirm that a considerable speedup is achieved when using the RB-EnKF instead of the full-order EnKF: the computational speedup in performing the filter goes from  $55\times$  to  $64.5\times$  for the first test case and from  $50\times$  to  $93\times$  in the second one. Nevertheless, the propagation of the reduction errors during the inversion procedure leads to biased estimates of the unknown state/parameter, as underlined also by our error analysis. The RB-EnKF has to be equipped with a statistical REM in order to recover unbiased output evaluation and consequently to guarantee the accuracy of the overall filtering technique. Moreover, the additional costs introduced by the REM are negligible: in the worst case scenario the computational time required by the corrected RB-EnKF is twice the uncorrected RB-EnKF, ensuring anyway a considerable speed up with respect to the full-order EnKF.

## References

- [1] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.
- [2] C. Brett, K. F. Lam, K. Law, D. McCormick, M. Scott, and A. Stuart. Accuracy and stability of filters for dissipative pdes. *Physica D: Nonlinear Phenomena*, 245(1):34–45, 2013.
- [3] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comp.*, 32(5):2737–2764, 2010.
- [4] T. Cui, Y. M. Marzouk, and K. E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.
- [5] M. Dihlmann and B. Haasdonk. A reduced basis Kalman filter for parametrized partial differential equations. *ESAIM: Control, Optimisation and Calculus of Variations*, 2015.
- [6] O. G. Ernst, B. Sprungk, and H.-J. Starkloff. Bayesian inverse problems and Kalman filters. In *Extraction of Quantifiable Information from Complex Systems*, volume 102 of *Lecture Notes in Computational Science and Engineering*, pages 133–159. Springer International Publishing, 2014.
- [7] O. G. Ernst, B. Sprungk, and H.-J. Starkloff. Analysis of the ensemble and polynomial chaos Kalman filters in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):823–851, 2015.
- [8] G. Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, 125(6):1342–1354, 1997.
- [9] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.
- [10] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445, 1961.
- [11] M. Frangos, Y. Marzouk, K. Willcox, and B. van Bloemen Waanders. Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems. *Large-Scale Inverse Problems and Quantification of Uncertainty*, 123149, 2010.

- [12] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International journal for numerical methods in engineering*, 81(12):1581–1608, 2010.
- [13] R. Giraldo, P. Delicado, and J. Mateu. Ordinary kriging for function-valued spatial data. *Environmental and Ecological Statistics*, 18(3):411–426, 2011.
- [14] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [15] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [16] C. Himpe and M. Ohlberger. Data-driven combined state and parameter reduction for inverse problems. *Advances in Computational Mathematics*, 41(5):1343–1364, 2015.
- [17] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.
- [18] J. Huttunen and J. Kaipio. Approximation error analysis in nonlinear state estimation with an application to state-space identification. *Inverse Problems*, 23(5):2141, 2007.
- [19] J. Huttunen and J. Kaipio. Model reduction in state identification problems with an application to determination of thermal parameters. *Applied Numerical Mathematics*, 59(5):877–890, 2009.
- [20] M. A. Iglesias, K. J. Law, and A. M. Stuart. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.
- [21] J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.
- [22] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [23] D. Kelly, K. Law, and A. Stuart. Well-posedness and accuracy of the ensemble Kalman filter in discrete and continuous time. *Nonlinearity*, 27(10):2579, 2014.
- [24] K. Law, A. Stuart, and K. Zygalakis. *Data Assimilation: A Mathematical Introduction*. Texts in Applied Mathematics. Springer International Publishing, 2015.
- [25] B. Lin and D. McLaughlin. Efficient characterization of uncertain model parameters with a reduced-order ensemble Kalman filter. *SIAM Journal on Scientific Computing*, 36(2):B198–B224, 2014.
- [26] Y. Maday, N. C. Nguyen, A. T. Patera, and S. H. Pau. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis*, 8(1):383–404, 2009.
- [27] A. Manzoni, S. Pagani, and T. Lassila. Accurate solution of Bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models. *SIAM/ASA J. Uncertainty Quantification*, 4(1):380–412, 2016.
- [28] A. Menafoglio, P. Secchi, and M. Dalla Rosa. A universal kriging predictor for spatially dependent functional data of a Hilbert space. *Electronic Journal of Statistics*, 7:2209–2240, 2013.
- [29] J. D. Murray. *Mathematical Biology I: An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer-Verlag, Berlin Heidelberg, 3rd edition, 2002.
- [30] E. Nadal, F. Chinesta, P. Díez, F. Fuenmayor, and F. Denia. Real time parameter identification and solution reconstruction from experimental data using the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 296:113–128, 2015.

- [31] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.
- [32] O. Pajonk, B. V. Rosić, A. Litvinenko, and H. G. Matthies. A deterministic filter for non-gaussian Bayesian estimation—applications to dynamical system estimation with noisy measurements. *Physica D: Nonlinear Phenomena*, 241(7):775–788, 2012.
- [33] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*, volume 92 of *Unitext Series*. Springer International Publishing, 2016.
- [34] M. Rochoux, C. Emery, S. Ricci, B. Cuenot, and A. Trouvé. Towards predictive data-driven simulations of wildfire spread—part ii: Ensemble Kalman filter for the state estimation of a front-tracking simulator of wildfire spread. *Natural Hazards and Earth System Science*, 15(8):1721–1739, 2015.
- [35] A. Solonen, T. Cui, J. Hakkarainen, and Y. Marzouk. On dimension reduction in gaussian filters. *arXiv preprint arXiv:1508.06452*, 2015.
- [36] R. Ștefănescu and A. Sandu. Efficient approximation of sparse jacobians for time-implicit reduced order models. *arXiv:1409.5506v3*, 2015.
- [37] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.
- [38] A. Tarantola. *Inverse problem theory and methods for model parameter estimation*: SIAM, 2005.
- [39] V. Tikhomirov. A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem. In *Selected works of AN Kolmogorov*. Springer, 1991.

## A Derivation of the Kalman update formula

In this section we provide the detailed derivation of the Kalman update formula (15) for the *analysis stage*. We will extend to our state/parameter case the derivation proposed for the state filter in [24, Chapter 4]. For simplicity we show the derivation procedure for the output

$$\mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}) = \mathbf{H}\mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}),$$

the more general case of (5) being essentially very similar to deal with. By assuming that the *prior* and the *likelihood* are multivariate Gaussian distributions, we can rewrite (10) for each element of the ensemble as

$$\begin{aligned} & \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{u} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\mu}_q^{(k+1)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k+1)}) \end{bmatrix}\right\|_{\mathcal{C}^h}^2\right) \\ & \propto \exp\left(-\frac{1}{2}\left\|\mathbf{s}^{(k+1)} - \mathbf{s}_h^{(k+1)}(\boldsymbol{\mu})\right\|_{\Gamma}^2 - \frac{1}{2}\left\|\begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{u} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\mu}_q^{(k)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k)}) \end{bmatrix}\right\|_{\mathcal{C}_{\text{prior}}^h}^2\right) \end{aligned} \quad (39)$$

where  $\mathcal{C}_{\text{prior}}^h$  is the covariance matrix of the parameter-state vector, given by

$$\mathcal{C}_{\text{prior}}^h = \begin{bmatrix} \mathbf{C}_{\boldsymbol{\mu}\boldsymbol{\mu}}^{(k+1)} & \mathbf{C}_{\boldsymbol{\mu}\mathbf{u}_h}^{(k+1)} \\ \mathbf{C}_{\mathbf{u}_h\boldsymbol{\mu}}^{(k+1)} & \mathbf{C}_{\mathbf{u}_h\mathbf{u}_h}^{(k+1)} \end{bmatrix}.$$

We recall that, in order to solve the inverse problem, we need to identify the unknown posterior mean  $[\boldsymbol{\mu}_q^{(k+1)}, \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k+1)})]^T$  and the unknown covariance  $\mathcal{C}^h$  of the posterior pdf. By requiring

that the means and the covariances of the quantities appearing in (39) are equal, we derive two equations to be satisfied by the unknown mean and covariance:

$$(\mathcal{C}^h)^{-1} = (\mathcal{C}_{\text{prior}}^h)^{-1} + \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix}^T, \quad (40)$$

$$(\mathcal{C}^h)^{-1} \begin{bmatrix} \boldsymbol{\mu}_q^{(k+1)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k+1)}) \end{bmatrix} = (\mathcal{C}_{\text{prior}}^h)^{-1} \begin{bmatrix} \boldsymbol{\mu}_q^{(k)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \Gamma^{-1} \mathbf{s}^{(k+1)}. \quad (41)$$

From (40), by applying the Woodbury matrix identity (see e.g. [24, Chapter 4]), we get

$$\mathcal{C}^h = \left( \mathcal{I} - \mathcal{C}_{\text{prior}}^h \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} (\Gamma + \mathbf{C}_{s_h s_h}^{(k+1)})^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix}^T \right) \mathcal{C}_{\text{prior}}^h;$$

substituting this last expression in (41) we obtain:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\mu}_q^{(k+1)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k+1)}) \end{bmatrix} &= \mathcal{C}^h \left( (\mathcal{C}_{\text{prior}}^h)^{-1} \begin{bmatrix} \boldsymbol{\mu}_q^{(k)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \Gamma^{-1} \mathbf{s}^{(k+1)} \right) \\ &= \left( \mathcal{I} - \mathcal{C}_{\text{prior}}^h \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} (\Gamma + \mathbf{C}_{s_h s_h}^{(k+1)})^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix}^T \right) \begin{bmatrix} \boldsymbol{\mu}_q^{(k)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k)}) \end{bmatrix} + \mathcal{C}^h \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \Gamma^{-1} \mathbf{s}^{(k+1)}. \end{aligned}$$

By observing that

$$\mathcal{C}^h \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix}^T = \mathcal{I} - \mathcal{C}^h (\mathcal{C}_{\text{prior}}^h)^{-1} = \mathcal{C}_{\text{prior}}^h \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix} (\Gamma + \mathbf{C}_{s_h s_h}^{(k+1)})^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix}^T$$

and that

$$\mathbf{s}^{(k+1)} - \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\mu}_q^{(k)} \\ \mathbf{u}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k)}) \end{bmatrix} = \mathbf{s}^{(k+1)} - \mathbf{s}_h^{(k+1)}(\boldsymbol{\mu}_q^{(k)}),$$

we easily obtain the system (15) for the Kalman update. The generalization of the updating formula to the case of a different output, like the one expressed by (5), can be obtained in a very similar way by considering as prior mean

$$\begin{bmatrix} \boldsymbol{\mu}_q^{(k)} & \mathbf{u}_h(\tau^{(k+1)}; \boldsymbol{\mu}_q^{(k)}) & \cdots & \mathbf{u}_h(t^{(\ell)}; \boldsymbol{\mu}_q^{(k)}) & \cdots & \mathbf{u}_h(\tau^{(k)}; \boldsymbol{\mu}_q^{(k)}) \end{bmatrix}^T,$$

and as output operator  $\begin{bmatrix} \mathbf{0} & \Delta t \omega_K \mathbf{H} & \cdots & \Delta t \omega_{K-1} \mathbf{H} & \cdots & \Delta t \omega_1 \mathbf{H} \end{bmatrix}^T$ , and following the same approach of the proof reported above.

## B Construction of a functional kriging interpolant

Assume that the error  $\{\chi_t(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{P}\}$  is a (functional) random field, that is, a set of (functional) random variables indexed by  $\boldsymbol{\mu} \in \mathcal{P}$ , taking values in  $L^2(a, b)$ , with  $(a, b) \subseteq (0, T)$ . We assume that  $\chi_t(\boldsymbol{\mu})$  is a second-order stationary and isotropic random process, that is:

1. the mean and the variance are constant with respect to  $\boldsymbol{\mu} \in \mathcal{P}$ ,

$$\mathbb{E}[\chi_t(\boldsymbol{\mu})] = m_\chi(t), \quad \text{Var}[\chi_t(\boldsymbol{\mu})] = \sigma_\chi^2(t), \quad t \in (a, b);$$

2. the covariance depends only on the lag  $\delta = \|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\|$ ,

$$\text{Cov}(\chi_{t_1}(\boldsymbol{\mu}_\alpha), \chi_{t_2}(\boldsymbol{\mu}_\beta)) = c_{t_1, t_2}(\boldsymbol{\mu}_\alpha, \boldsymbol{\mu}_\beta) = c_{t_1, t_2}(\delta), \quad \boldsymbol{\mu}_\alpha, \boldsymbol{\mu}_\beta \in \mathcal{P}, \quad t_1, t_2 \in (a, b);$$

moreover, if  $t_1 = t_2 = t$ , we denote by  $c_t(\delta) = c_{t_1, t_2}(\delta)$ .

Given a sample of known functions of the random process  $\{\chi_t(\boldsymbol{\mu}_q)\}_{q=1}^{N_{cal}}$ , the best linear unbiased predictor (BLUP) of  $\{\chi_t(\boldsymbol{\mu}_0)\}$ , for each new  $\boldsymbol{\mu}_0 \in \mathcal{P}$ , is given by the linear combination

$$\hat{\chi}_t(\boldsymbol{\mu}) = \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}) \chi_t(\boldsymbol{\mu}_q) \quad \forall \boldsymbol{\mu} \in \mathcal{P}$$

whose weights are obtained by imposing that the mean square error of  $\hat{\chi}_t(\boldsymbol{\mu}_0)$  is minimized, i.e.,

$$[\lambda_1(\boldsymbol{\mu}_0), \dots, \lambda_{N_{cal}}(\boldsymbol{\mu}_0)]^T = \arg \min_{\lambda_1, \dots, \lambda_{N_{cal}}} \int_a^b \text{Var}[\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)] dt \quad (42)$$

under the constraint that  $\hat{\chi}_t(\boldsymbol{\mu}_0)$  is unbiased, that is

$$\mathbb{E}[\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)] = 0 \quad (43)$$

Finding the BLUP thus yields a constrained quadratic program to be solved (for each  $\boldsymbol{\mu}_0$ ): indeed,

$$\begin{aligned} \text{Var}(\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)) &= \sum_{q,p=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) \lambda_p(\boldsymbol{\mu}_0) c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_p) + c_t(\boldsymbol{\mu}_0, \boldsymbol{\mu}_0) - 2 \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) c_t(\boldsymbol{\mu}_i, \boldsymbol{\mu}_0) \quad (44) \\ \mathbb{E}[\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)] &= \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) \mathbb{E}[\hat{\chi}_t(\boldsymbol{\mu}_q)] - \mathbb{E}[\hat{\chi}_t(\boldsymbol{\mu}_0)] = \mathbb{E}[\hat{\chi}_t(\boldsymbol{\mu}_0)] \left( \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) - 1 \right) \end{aligned}$$

thanks to assumption 1. Denoting by

$$\mathcal{L}(\lambda_1, \dots, \lambda_{N_{cal}}, \eta) = \int_a^b \text{Var}(\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)) dt + \eta \int_a^b \mathbb{E}[\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)] dt$$

the Lagrangian functional associated to problem (42)–(43), we get the following linear system to be solved, for each  $\boldsymbol{\mu}_0$ :

$$\begin{bmatrix} \int_a^b c_t(\boldsymbol{\mu}_1, \boldsymbol{\mu}_1) dt & \dots & \int_a^b c_t(\boldsymbol{\mu}_1, \boldsymbol{\mu}_{N_{cal}}) dt & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \int_a^b c_t(\boldsymbol{\mu}_{N_{cal}}, \boldsymbol{\mu}_1) dt & \dots & \int_a^b c_t(\boldsymbol{\mu}_{N_{cal}}, \boldsymbol{\mu}_{N_{cal}}) dt & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{N_{cal}} \\ \eta \end{bmatrix} = \begin{bmatrix} \int_a^b c_t(\boldsymbol{\mu}_i, \boldsymbol{\mu}_0) dt \\ \vdots \\ \int_a^b c_t(\boldsymbol{\mu}_{N_{cal}}, \boldsymbol{\mu}_0) dt \\ 1 \end{bmatrix}.$$

Thanks to assumption 2,  $c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_p) = \text{Cov}(\chi_t(\boldsymbol{\mu}_q), \chi_t(\boldsymbol{\mu}_p)) = c_t(\|\boldsymbol{\mu}_q - \boldsymbol{\mu}_p\|)$  so that, by denoting

$$\gamma_t(\delta) = c_t(0) - c_t(\delta),$$

the previous linear system can be equivalently rewritten as:

$$\begin{bmatrix} \int_a^b \gamma_t(0) dt & \dots & \int_a^b \gamma_t(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_{N_{cal}}\|) dt & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \int_a^b \gamma_t(\|\boldsymbol{\mu}_{N_{cal}} - \boldsymbol{\mu}_1\|) dt & \dots & \int_a^b \gamma_t(\|\boldsymbol{\mu}_{N_{cal}} - \boldsymbol{\mu}_{N_{cal}}\|) dt & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{N_{cal}} \\ \eta \end{bmatrix} = \begin{bmatrix} \int_a^b \gamma_t(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|) dt \\ \vdots \\ \int_a^b \gamma_t(\|\boldsymbol{\mu}_{N_{cal}} - \boldsymbol{\mu}_0\|) dt \\ 1 \end{bmatrix}. \quad (45)$$

To compute the components appearing in the matrix and the vector of system (45), we estimate the so-called (*theoretical*) *semi-variogram*

$$\gamma(\delta) = \int_a^b \gamma_t(\delta) dt$$

with an empirical semi-variogram

$$\hat{\gamma}(\delta) = \frac{1}{2|N(\delta)|} \sum_{i,j \in N(\delta)} \int_a^b (\chi_t(\boldsymbol{\mu}_i) - \chi_t(\boldsymbol{\mu}_j))^2 dt \quad (46)$$

being  $N(\delta) = \{(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) : \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| = \delta\}$ . In practice, the empirical semi-variogram is estimated at  $M$  discrete lags  $\{\delta_1, \dots, \delta_M\}$ . Through these estimated values  $\{\hat{\gamma}(\delta_1), \dots, \hat{\gamma}(\delta_M)\}$ , a parametric semi-variogram model (e.g. spherical, exponential or gaussian) is fitted using a least squares approach. In this paper we consider the spherical parametric semi-variogram model, that is

$$\gamma(\delta) = \begin{cases} d \left(1.5 \frac{\delta}{c} - 0.5 \left(\frac{\delta}{c}\right)^3\right) & \delta \leq c \\ d & \delta > c \end{cases} \quad (47)$$

where  $c, d$  are the two parameters identified by fitting this model on the finite set of points  $\{(\delta_m, \hat{\gamma}(\delta_m))\}_{m=1}^M$ .

Finally, we define the prediction trace-variance  $\hat{\sigma}_\chi^2(\boldsymbol{\mu}_0)$  of the functional ordinary kriging as

$$\hat{\sigma}_\chi^2(\boldsymbol{\mu}_0) = \int_a^b \text{Var}(\hat{\chi}_t(\boldsymbol{\mu}_0) - \chi_t(\boldsymbol{\mu}_0)) dt.$$

Using (44), we get

$$\hat{\sigma}_\chi^2(\boldsymbol{\mu}_0) = \int_a^b \left( \sum_{q,p=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) \lambda_p(\boldsymbol{\mu}_0) c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_p) + c_t(\boldsymbol{\mu}_0, \boldsymbol{\mu}_0) - 2 \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_0) \right) dt. \quad (48)$$

Since  $\{\lambda_p(\boldsymbol{\mu}_0)\}_{p=1}^{N_{cal}}$  and  $\eta$  are the solution of the linear system (45), it holds

$$\int_a^b \left( \sum_{p=1}^{N_{cal}} \lambda_p(\boldsymbol{\mu}_0) c_t(\boldsymbol{\mu}_p, \boldsymbol{\mu}_q) \right) dt = \int_a^b c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_0) dt - \eta \quad \forall q = 1, \dots, N_{cal},$$

so that, by substituting this latter relation in (44), we finally get

$$\begin{aligned} \hat{\sigma}_\chi^2(\boldsymbol{\mu}_0) &= \int_a^b \left( \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) (c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_0) + c_t(\boldsymbol{\mu}_0, \boldsymbol{\mu}_0)) - 2 \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_0) \right) dt - \underbrace{\sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) \eta}_{=1} \\ &= \int_a^b \left( \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) \underbrace{(c_t(\boldsymbol{\mu}_0, \boldsymbol{\mu}_0) - c_t(\boldsymbol{\mu}_q, \boldsymbol{\mu}_0))}_{=\gamma_t(\|\boldsymbol{\mu}_q - \boldsymbol{\mu}_0\|)} \right) dt - \eta = \sum_{q=1}^{N_{cal}} \lambda_q(\boldsymbol{\mu}_0) \gamma(\|\boldsymbol{\mu}_q - \boldsymbol{\mu}_0\|) - \eta. \end{aligned} \quad (49)$$

## MOX Technical Reports, last issues

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

- 23/2016** Fedele, M.; Faggiano, E.; Dedè, L.; Quarteroni, A.  
*A Patient-Specific Aortic Valve Model based on Moving Resistive Immersed Implicit Surfaces*
- 22/2016** Antonietti, P.F.; Facciola', C.; Russo, A.; Verani, M.  
*Discontinuous Galerkin approximation of flows in fractured porous media*
- 21/2016** Ambrosi, D.; Zanzottera, A.  
*Mechanics and polarity in cell motility*
- 19/2016** Guerciotti, B.; Vergara, C.  
*Computational comparison between Newtonian and non-Newtonian blood rheologies in stenotic vessels*
- 20/2016** Wilhelm, M.; Sangalli, L.M.  
*Generalized Spatial Regression with Differential Regularization*
- Guerciotti, B.; Vergara, C.  
*Computational comparison between Newtonian and non-Newtonian blood rheologies in stenotic vessels*
- 18/2016** Ferroni, A.; Antonietti, P.F.; Mazzieri, I.; Quarteroni, A.  
*Dispersion-dissipation analysis of 3D continuous and discontinuous spectral element methods for the elastodynamics equation*
- 17/2016** Penati, M.; Miglio, E.  
*A new mixed method for the Stokes equations based on stress-velocity-vorticity formulation*
- 15/2016** Ieva, F.; Paganoni, A.M.  
*A taxonomy of outlier detection methods for robust classification in multivariate functional data*
- 16/2016** Agosti, A.; Antonietti, P.F.; Ciarletta, P.; Grasselli, M.; Verani, M.  
*A Cahn-Hilliard type equation with degenerate mobility and single-well potential. Part I: convergence analysis of a continuous Galerkin finite element discretization.*