A low-rank solver for parameter estimation and uncertainty quantification in linear time dependent systems of Partial Differential Equations

Sébastien Riffaud, Miguel Angel Fernández, Damiano Lombardi

A low-rank solver for parameter estimation and uncertainty quantification in time-dependent systems of Partial Differential Equations

Sébastien Riffaud[1], Miguel A. Fernández[1*] and Damiano Lombardi[1]

[1]Sorbonne Université, Inria & CNRS, UMR 7598 Laboratoire Jacques-Louis Lions, Paris, France.

*Corresponding author(s). E-mail(s): miguel.fernandez@inria.fr;
Contributing authors: sebastien.riffaud@inria.fr;
damiano.lombardi@inria.fr;

Abstract

In this work we propose a low-rank solver in view of performing parameter estimation and uncertainty quantification in systems of Partial Differential Equations. The solution approximation is sought in a space-parameter separated form. The discretisation in the parameter direction is made evolve in time through a Markov Chain Monte Carlo method. The resulting method is a Bayesian sequential estimation of the parameters. The computational burden is mitigated by the introduction of an efficient interpolator, based on a reduced basis built by exploiting the low-rank solves. The method is tested on four different applications.

Keywords: Low-rank approximation, Tensor methods, Parameter estimation, Uncertainty quantification

1 Introduction

This work deals with systems of time-dependent parametric Partial Differential Equations (PDEs). In numerous realistic applications, parameter estimation

is crucial in view of performing Data Assimilation and Uncertainty Quantification (UQ) tasks. There is an abundant literature on these topics. Far from being exhaustive, we propose hereafter a short review of the results which provide the context of the present work.

There are two major difficulties to be faced when trying to estimate the parameter values by exploiting solution observations. First, studying the identifiability (first formalised in [1]), which relates to the natural question: can the parameters be estimated given the available observations? Answering this question amounts to study the injectivity of the parameter-to-observable map, which is computationally hard or prohibitive in many realistic systems. Second, even if the system of PDEs is linear with respect to the unknown, the solution often depends in a non-linear way on the parameters. Performing parameter estimation in a classical variational formulation reduces henceforth to solve a non-linear non-convex optimisation problem. Bayesian methods may provide a robust framework to deal with parameter estimation, making available some information on the eventual non-identifiability of some of the parameters. We refer to [2–4] and the references therein. The major issue to be addressed is the computational cost. The computational complexity of the resolution of Bayesian problems with several Markov Chain Monte Carlo (MCMC) methods has been investigated in [5]. Several studies in the literature investigated parsimonious methods to perform Bayesian estimation when large systems are at hand. Among them, we can distinguish several classes. Polynomial chaos and dimensionality reduction were considered in [6]. A polynomial chaos expansion is proposed in [7], a spectral likelihood approximation is described in [8], sparse polynomials approximations were proposed in [9, 10] and an adaptive, multi-fidelity method is detailed in [11]. A sparse grid discretisation was used in [12]. The reduced basis method, consisting in building a parsimonious discretisation by projection on a small dimensional basis (built semi-empirically given samples of the parametric solution) was investigated in [13–17]. Recently, the work [18] considers a mean field approximation in view of performing Bayesian parameter estimation for large systems, and dimensionality reduction in Bayesian estimation was investigated in [19].

This work proposes a sequential estimation method. In this class of methods, the data are acquired progressively and make it possible to correct the estimation of the posterior distribution of the hidden quantities. An overview on Bayesian filtering can be found in [20, 21]. Exact filtering requires the solution of a PDE on the density function (see, e.g., [22]). Particle filters and Sequential Monte Carlo have been investigated in [23, 24] and numerous other works. One of the formulations which are the most used in practice is the ensemble Kalman filter (one of the non-linear generalisation of the Kalman filter). We refer to [25] for an extensive presentation of the method. Several studies in the litterature proposed methods to mitigate the computational burden when dealing with large systems: in [26] a polynomial chaos formulation is analysed, in [27, 28] the reduced basis method is used to accelerate the ensemble Kalman

filter. In the present work we propose a method to perform parameter estimation in a sequential way, for systems of time-dependent PDEs. Contrary to the works making use of the reduced basis method, we rely on parcimonius discretisations of the parametric PDEs, with no *offline* phase. In [29] an analysis is proposed, comparing sparse polynomial representations and low-rank approximations of parametric PDEs. In this work, we make the choice of relying on low-rank approximations. These were investigated in several studies. In [30, 31] the authors used the Proper Generalised Decomposition (PGD), a greedy algorithm to approximate a problem solution in a canonical tensor format. An analysis of the convergence of the Galerkin formulation for sums of rank-1 terms is proposed in [32]. In [33] uncertainty quantification has been considered combining ANOVA and the Tensor Train format. Hierarchical tensor approximation have been considered in [34], and in [35] the authors propose a method which combines Alternating Least Squares and TT-cross approximation in order to represent the solution of a parametric PDE in Tensor Train format. The discretisation of stochastic elliptic PDEs in quantics-Tensor Train format has been proposed in [36].

Another possibility in view of performing the integration of time-dependent parametric systems are the Dynamical Orthogonal decomposition, presented in [37–39] and the bi-orthogonal decomposition [40, 41]. The dynamical low-rank methods, introduced in [42] are usually used by fixing *a priori* the rank. In [43], a method to adapt the rank is investigated.

In the present work, the formulation considered is similar, in spirit, to the one proposed in [44–46]. When discretising the system, we will use a preconditioned Krylov-based method (we refer to [47]) to solve the discrete problem. The key contribution is the following: after having semi-discretised in time the PDEs, we will discretise in space by considering a standard Galerkin method and in parameters by using a collocation. We will make the parameter discretisation evolve with an MCMC strategy in such a way that the collocation points will distribute approximately as the posterior distribution, progressively. This have an important consequence. In many realistic systems, the parametric solution cannot be expected to be low-rank, meaning that it cannot always be represented with a small number of separated terms. This is especially true, roughly speaking, when considering large parameter intervals in which the solution may exhibit various regimes. When considering parameter estimation and making the parameter discretisation evolve, we can expect that the rank stays moderate. For example, in a structurally identifiable problem, when having zero measurement noise, we would end up with a rank-1 solution. We will introduce a method in which the low-rank tensor solver works at the same level of precision of a classical solver (the solver we would use to approximate the solution of one parametric instance). The likelihood evaluation involved in the MCMC algorithm is accelerated by introducing an efficient interpolation which leverages the low-rank structure of the solution. This is built by Galerkin projection on a time evolving basis computed thanks to the low-rank

solves. Overall, we have a sequential Bayesian parameter estimation based on a single parametric PDE solution.

The rest of the paper is organized as follows. In Section 2, we introduce the MCMC method employed to perform parameter estimation and uncertainty quantification. Then, Sections 3 and 4 respectively present the low-rank solver and interpolator used for the numerical solutions of parametric time-dependent problems. In Section 5, the performance of the proposed method is demonstrated for four different applications. Finally, Section 6 draws the conclusions of the present work.

2 Bayesian parameter estimation

In this section we present the method used in order to adapt the discretisation in the parameters direction. The goal is to solve a Bayesian problem and have the collocation points of the parameter discretisation distributed according to the posterior distribution. Let $\mathbf{z}_{\text{obs}}$ denote the observations. They are partial noisy observation of the system state. From now on we assume that the noise is additive, unbiased, normally distributed $\mathcal{N}(0, \Sigma)$ with covariance Σ. The prior probability density distribution π for the parameters is assumed to be uniform in an interval (the less informative prior in a bounded domain). The posterior distribution ρ is obtained by applying the infinite dimensional Bayes theorem (we refer to [4] for a comprehensive overview):

$$\rho(\boldsymbol{\theta}) \propto \exp(-\Phi(\mathbf{z}_{\text{obs}}, \boldsymbol{\theta}))\pi(\boldsymbol{\theta}),$$

where Φ is called the potential (it is minus the log-likelihood), encoding the effect of the discrepancy between the actual system measurements and the model observations. In this work we use a classical Metropolis-Hastings method to approximately distribute the collocation points according to the posterior.

2.1 Parameter estimation problem

Let the parameter domain $\mathcal{D}$ be a closed and bounded subset of the Euclidean space $\mathbb{R}^{N_p}$. In this work, the evolution of the state $\mathbf{x}(t_k;\ \boldsymbol{\theta}) \in \mathbb{R}^{N_x}$ and its observation $\mathbf{z}(t_k;\ \boldsymbol{\theta}) \in \mathbb{R}^{N_z}$ is related to the parameters $\boldsymbol{\theta} \in \mathcal{D}$, after discretisation in space and time (by means of standard numerical methods), according to

$$\begin{cases} \mathbf{x}(t_k;\ \boldsymbol{\theta}) = \mathbf{x}(t_{k-1};\ \boldsymbol{\theta}) + \Delta t \boldsymbol{f}(\mathbf{x}(t_k;\ \boldsymbol{\theta}), t_k;\ \boldsymbol{\theta}) \\ \mathbf{z}(t_k;\ \boldsymbol{\theta}) = \boldsymbol{h}(\mathbf{x}(t_k;\ \boldsymbol{\theta}), t_k;\ \boldsymbol{\theta}) + \boldsymbol{\eta}(t_k;\ \boldsymbol{\theta}), \end{cases}$$

where $\Delta t > 0$ is the time-step size, $\boldsymbol{f} : \mathbb{R}^{N_x} \times \mathbb{R} \times \mathcal{D} \to \mathbb{R}^{N_x}$ denotes the (known) state-transition model, $\boldsymbol{h} : \mathbb{R}^{N_x} \times \mathbb{R} \times \mathcal{D} \to \mathbb{R}^{N_z}$ denotes the (known) observation model, and $\boldsymbol{\eta}(t_k;\ \boldsymbol{\theta}) \in \mathbb{R}^{N_z}$ denotes the observation noise, which is assumed to be a zero mean Gaussian white noise.

2.2 Sequential Bayesian inference

Let $\{\mathbf{z}_{\text{obs}}(t_1), \dots, \mathbf{z}_{\text{obs}}(t_{N_t})\}$ be a temporal sequence of observations, a sequential MCMC method is employed to estimate the posterior probability distribution $\rho(\boldsymbol{\theta} \mid \mathbf{z}_{\text{obs}}(t_1), \dots, \mathbf{z}_{\text{obs}}(t_{N_t}))$. In particular, the Monte Carlo (MC) samples $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{N_\theta}\}$ are updated at each new observation $\mathbf{z}_{\text{obs}}(t_k)$ by the Metropolis-Hastings algorithm [48, 49]:

1. Uniformly sample the parameter domain $\mathcal{D}$ to initialize $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{N_\theta}\}$.
2. For $k \in \{1, \dots, N_t\}$:
   - For $i \in \{1, \dots, N_\theta\}$:
     - For $j \in \{1, \dots, N_{\text{mh}}\}$:
       (a) Generate candidates
       $\widehat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i + \delta \Delta\boldsymbol{\theta}_i$ where $\Delta\boldsymbol{\theta}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$.
       (b) Calculate the acceptance ratio

$$\gamma_i = \frac{\exp\left(-\sum_{m=1}^{k} \left\| \mathbf{z}_{\text{obs}}(t_m) - \boldsymbol{h}\big(\mathbf{x}(t_m;\ \widehat{\boldsymbol{\theta}}_i), t_m;\ \widehat{\boldsymbol{\theta}}_i\big) \right\|_2^2 \right)}{\exp\left(-\sum_{m=1}^{k} \left\| \mathbf{z}_{\text{obs}}(t_m) - \boldsymbol{h}\big(\mathbf{x}(t_m;\ \boldsymbol{\theta}_i), t_m;\ \boldsymbol{\theta}_i\big) \right\|_2^2 \right)}.$$

       (c) Accept or reject candidates
       if $\tau_i \leqslant \gamma_i$ with $\tau_i \sim \mathcal{U}(0, 1)$, then set $\boldsymbol{\theta}_i = \widehat{\boldsymbol{\theta}}_i$.

In the numerical experiments presented in Section 5, the number of Metropolis-Hastings steps is chosen as $N_{\text{mh}} = 50$, which was observed to be sufficient for the sampling to converge in the first moments, and the step size is taken as $\delta = 10^{-2}$, and the correlation matrix $\mathbf{C} \in \mathbb{R}^{N_p \times N_p}$ is defined as

$$\mathbf{C} = \text{diag}(\boldsymbol{\theta}_{\max} - \boldsymbol{\theta}_{\min})^2,$$

where $\boldsymbol{\theta}_{\min}$ and $\boldsymbol{\theta}_{\max}$ are respectively the lower and upper bounds of the parameter domain (i.e., $\boldsymbol{\theta}_{\min} \leqslant \boldsymbol{\theta} \leqslant \boldsymbol{\theta}_{\max}$ for any $\boldsymbol{\theta} \in \mathcal{D}$).

2.3 Uncertainty quantification

Let $\mathbf{q}(t_k;\ \boldsymbol{\theta}) \in \mathbb{R}^{N_q}$ be a selection of quantities of interest (QoIs). Once the posterior probability distribution is sampled, it becomes possible to estimate several statistical quantities, such as the empirical mean and (elementwise) variance, respectively, given by

$$\boldsymbol{\mu}_{\mathbf{q}}(t_k) = \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} \mathbf{q}(t_k;\ \boldsymbol{\theta}_i) \quad \text{and} \quad \boldsymbol{\sigma}^2_{\mathbf{q}}(t_k) = \frac{1}{N_\theta - 1} \sum_{i=1}^{N_\theta} \big(\mathbf{q}(t_k;\ \boldsymbol{\theta}_i) - \boldsymbol{\mu}_{\mathbf{q}}(t_k)\big)^2.$$

In particular, the mean provides access to unobserved QoIs of the solution, while the variance quantifies the accuracy of this estimation.

3 Low-rank solver

In this section the low-rank solver for parametric PDEs is presented. Let the parameter domain be $\mathcal{D} \subset \mathbb{R}^{N_p}$, a compact set. Let $\Omega \subset \mathbb{R}^d$ be the space domain. We present hereafter the context for a generic parametric time-dependent PDE. The extension to vector valued solutions and to systems of parametric PDEs is straightforward. A parametric PDE can be written in abstract form as:

$$\partial_t u = \mathcal{A}(\boldsymbol{\theta})u + \mathcal{G}(u, t;\ \boldsymbol{\theta}),$$

where the function u is the solution, $\mathcal{A}$ is a linear operator acting on u, and $\mathcal{G}$ is a nonlinear operator. In the present work we are interesting in studying the solution as function of the space-time variable and the parameters acting as extra variables, hence a function $u : \Omega \times \mathcal{D} \times [0, T] \to \mathbb{R}$. The operator $\mathcal{A}$ satisfies the following hypotheses:

1. $\mathcal{A}$ is linear in u;
2. $\mathcal{A}$ is a finite rank tensorised operator. Let $\mathcal{V}(\Omega)$, $\mathcal{V}(\mathcal{D})$ be two Hilbert spaces of functions on Ω and $\mathcal{D}$ respectively. Let $A^{(\Omega)} : \mathcal{V}(\Omega) \to \mathcal{V}(\Omega)$ and $A^{(\mathcal{D})} : \mathcal{V}(\mathcal{D}) \to \mathcal{V}(\mathcal{D})$ be two linear continuos operators. A rank-1 tensorised operator is defined as:

$$A^{(\Omega)} \otimes A^{(\mathcal{D})} : \begin{cases} \mathcal{V}(\Omega) \times \mathcal{V}(\mathcal{D}) \ \to \ \ \mathcal{V}(\Omega) \times \mathcal{V}(\mathcal{D}) \\ v \otimes w \ \mapsto A^{(\Omega)} v \otimes A^{(\mathcal{D})} w \end{cases}$$

   Asking for $\mathcal{A}$ to be a finite rank operator means that there exist a finite number $L \in \mathbb{N}^*$ of rank-1 operators $\left\{ A_l^{(\Omega)} \otimes A_l^{(\mathcal{D})} \right\}_{1 \le l \le L}$ such that:

$$\mathcal{A} = \sum_{l=1}^{L} A_l^{(\Omega)} \otimes A_l^{(\mathcal{D})}.$$

In this work, the PDE is first semi-discretised in time. Let $k \in \mathbb{N}^*$ denote the time iteration level. When considering the above stated hypotheses and using an implicit Euler scheme we would get the semi-discretised form:

$$\big(\mathcal{I} - \Delta t \mathcal{A}(\boldsymbol{\theta})\big) u^{(k+1)} = u^{(k)} + \Delta t \mathcal{G}(u^{(k+1)}, t_{k+1};\ \boldsymbol{\theta}),$$

where $\mathcal{I}$ denotes the identity operator.

We can now discretise this in space-parameters and obtain the full discrete formulation of the problem. When using a Galerkin method in space and a collocation method in parameters this lead to a parametrised linear system to be solved. The hypothesis on the tensorised structure of the operator would make it possible to look for a solution representation in a low rank format and to deal with the system resolution in an efficient way. This is presented in the following section.

3.1 Low-rank approximation

Let $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_\theta}\}$ be the MC samples, the corresponding solutions $\mathbf{x}(t_k;\ \boldsymbol{\theta})$ at time-step t_k can be assembled into the matrix

$$\mathbf{X}_k = \left[\mathbf{x}(t_k;\ \boldsymbol{\theta}_1)\ \ldots\ \mathbf{x}(t_k;\ \boldsymbol{\theta}_{N_\theta})\right] \in \mathbb{R}^{N_x \times N_\theta}.$$

In order to reduce the number of degrees of freedom (DOFs), the solution $\mathbf{X}_k$ is approximated by the low-rank matrix

$$\overline{\mathbf{X}}_k = \sum_{n=1}^{r_k} \sigma_n^k \mathbf{u}_n^k (\mathbf{v}_n^k)^T = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T, \tag{1}$$

where $\mathbf{U}_k = [\mathbf{u}_1^k, \ldots, \mathbf{u}_{r_k}^k] \in \mathbb{R}^{N_x \times r_k}$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^k, \ldots, \sigma_{r_k}^k) \in \mathbb{R}^{r_k \times r_k}$, $\mathbf{V}_k = [\mathbf{v}_1^k, \ldots, \mathbf{v}_{r_k}^k] \in \mathbb{R}^{N_\theta \times r_k}$, and $r_k \ll N_x, N_\theta$. In particular, this second-order tensor is truncated using the Eckart-Young theorem [50] to minimize the approximation error. Without loss of generality, we can assume that $\overline{\mathbf{X}}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$ is the singular value decomposition (SVD) of $\overline{\mathbf{X}}_k$. If $\mathbf{U}_k$ or $\mathbf{V}_k$ are not unitary matrices, Algorithm 2 can be used to efficiently orthonormalized them. The low-rank matrix $\overline{\mathbf{X}}_k$ is then obtained by keeping only the first r_k columns of $\mathbf{U}_k$ and $\mathbf{V}_k$. In practice, the rank $r_k \in \mathbb{N}$ is defined as the smallest integer such that the relative truncation error is less than a given tolerance $\eta_{\text{trunc}} \in [0, 1]$:

$$\left\|\overline{\mathbf{X}}_k - \boldsymbol{\mathcal{T}}(\overline{\mathbf{X}}_k)\right\|_F \leqslant \eta_{\text{trunc}} \left\|\overline{\mathbf{X}}_k\right\|_F, \tag{2}$$

where $\boldsymbol{\mathcal{T}} : \mathbb{R}^{N_x \times N_\theta} \to \mathbb{R}^{N_x \times N_\theta}$ denotes the resulting rounding operator, and $\|\cdot\|_F$ stands for the Frobenius norm.

3.2 Linear time-invariant system

Given the solution approximation (1) defined above, we now present how $\overline{\mathbf{X}}_k$ is determined. We first consider linear state-transition models of the form

$$\mathbf{A}(\boldsymbol{\theta})\mathbf{x}(t_k;\ \boldsymbol{\theta}) = \mathbf{b}(t_k;\ \boldsymbol{\theta}), \tag{3}$$

where $\mathbf{A}(\boldsymbol{\theta}) \in \mathbb{R}^{N_x \times N_x}$, and $\mathbf{b}(t_k;\ \boldsymbol{\theta}) \in \mathbb{R}^{N_x}$ may depend on the solution at the previous time-step $\mathbf{x}(t_{k-1};\ \boldsymbol{\theta})$. Given $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_\theta}\}$, the corresponding systems (3) can be assembled into the linear system

$$\boldsymbol{\mathcal{A}}(\mathbf{X}_k) = \mathbf{B}_k, \tag{4}$$

where $\boldsymbol{\mathcal{A}}(\mathbf{X}_k) = [\mathbf{A}(\boldsymbol{\theta}_1)\mathbf{x}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{A}(\boldsymbol{\theta}_{N_\theta})\mathbf{x}(t_k;\ \boldsymbol{\theta}_{N_\theta})] \in \mathbb{R}^{N_x \times N_\theta}$, and $\mathbf{B}_k = [\mathbf{b}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{b}(t_k;\ \boldsymbol{\theta}_{N_\theta})] \in \mathbb{R}^{N_x \times N_\theta}$. The system (3) is solved by a truncated version of the generalized minimal residual (GMRES) algorithm [51–53], which approximates the solution (and the other vectors) in a low-rank form similar to (1). This iterative method is initialized using the solution at

the previous time-step $\overline{\mathbf{X}}_{k-1}$ and terminates when the relative residual is less than a given tolerance $\varepsilon_{\text{gmres}} \in (0, 1]$:

$$\left\|\boldsymbol{\mathcal{A}}(\overline{\mathbf{X}}_k) - \overline{\mathbf{B}}_k\right\|_F \leqslant \varepsilon_{\text{gmres}} \left\|\overline{\mathbf{B}}_k\right\|_F . \tag{5}$$

The number of iterations $N_{\text{gmres}} \in \mathbb{N}$ is therefore defined as the smallest integer such that the stopping criterion (5) is satisfied. In addition, the algorithm is equipped with a left preconditioning matrix $\mathbf{P} \in \mathbb{R}^{N_x \times N_x}$ to speedup convergence. A common choice for the preconditioner is $\mathbf{P} \approx \mathbf{A}(\boldsymbol{\theta})$. However, since we have the matrices $\{\mathbf{A}(\boldsymbol{\theta}_1), \ldots, \mathbf{A}(\boldsymbol{\theta}_{N_\theta})\}$, this would result in N_θ preconditioning matrices, which would be prohibitive to store in memory. For this reason, we choose the preconditioner $\mathbf{P} = \mathbf{A}(\boldsymbol{\theta}_c)$, where $\boldsymbol{\theta}_c$ is the center of the parameter domain $\mathcal{D}$ (i.e., $\boldsymbol{\theta}_c = \frac{\boldsymbol{\theta}_{\min} + \boldsymbol{\theta}_{\max}}{2}$). A similar choice is discussed in [54]. The resulting linear solver is described in Algorithm 3. Notably, the rounding operator $\boldsymbol{\mathcal{T}}$ is used at lines 1, 5, 8 and 14 to balance the increase in rank occurring when tensors are added.

Remark 1 Since $\mathbf{A}(\boldsymbol{\theta})$ depends only on $\boldsymbol{\theta}$, this matrix can be decomposed as follows

$$\mathbf{A}(\boldsymbol{\theta}) = \sum_{n=1}^{N_\alpha} \alpha_n(\boldsymbol{\theta}) \mathbf{A}_n, \tag{6}$$

where $N_\alpha \ll N_\theta$. For example, the heat conduction problem described in Section 5.1 leads to a discrete system with a matrix of the form

$$\mathbf{A}(\boldsymbol{\theta}) = \mathbf{A}_1 + \kappa \mathbf{A}_2 + \nu \mathbf{A}_2,$$

where

$$(\mathbf{A}_1)_{i,j} := \frac{1}{\Delta t} \int_\Omega \phi_i \phi_j \, \mathrm{d}\boldsymbol{x}, \quad (\mathbf{A}_2)_{i,j} := \int_\Omega \nabla\phi_i \cdot \nabla\phi_j \, \mathrm{d}\boldsymbol{x}, \quad (\mathbf{A}_3)_{i,j} := \int_{\Gamma_{\text{int}}} \phi_i \phi_j \, \mathrm{d}\boldsymbol{s},$$

and $\{\phi_1, \ldots, \phi_{N_x}\}$ denotes the finite element basis. The operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{N_x \times N_\theta} \to \mathbb{R}^{N_x \times N_\theta}$ thus requires to store only the matrices $\{\mathbf{A}_1, \ldots, \mathbf{A}_{N_\alpha}\}$, and not all the matrices $\{\mathbf{A}(\boldsymbol{\theta}_1), \ldots, \mathbf{A}(\boldsymbol{\theta}_{N_\theta})\}$, which would be prohibitive in terms of storage complexity.

3.3 Extension to nonlinear systems

We now extend the method to nonlinear state-transition models:

$$\mathbf{A}(\boldsymbol{\theta})\mathbf{x}(t_k;\ \boldsymbol{\theta}) = \mathbf{b}(t_k;\ \boldsymbol{\theta}) + \mathbf{g}(\mathbf{x}(t_k;\ \boldsymbol{\theta}), t_k;\ \boldsymbol{\theta}), \tag{7}$$

where $\boldsymbol{g} : \mathbb{R}^{N_x} \times \mathbb{R} \times \mathcal{D} \to \mathbb{R}^{N_x}$ is nonlinear with respect to the solution $\mathbf{x}(t_k;\ \boldsymbol{\theta})$. Given $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_\theta}\}$, the corresponding systems (7) can be assembled into

$$\boldsymbol{\mathcal{A}}(\mathbf{X}_k) = \mathbf{B}_k + \boldsymbol{\mathcal{G}}(\mathbf{X}_k), \tag{8}$$

where $\boldsymbol{\mathcal{G}}(\mathbf{X}_k) = \left[\mathbf{g}(\mathbf{x}(t_k;\ \boldsymbol{\theta}_1), t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{g}(\mathbf{x}(t_k;\ \boldsymbol{\theta}_{N_\theta}), t_k;\ \boldsymbol{\theta}_{N_\theta})\right] \in \mathbb{R}^{N_x \times N_\theta}$. The system (8) is solved by a fixed-point iteration method, where the linear

(resp. nonlinear) terms are treated implicitly (resp. explicitly), and the resulting linear system is solved by Algorithm 3. The method is initialized using the solution at the previous time-step $\overline{\mathbf{X}}_{k-1}$ and terminates when the (absolute or relative) difference between two iterates is less than a given tolerance $\varepsilon_{\text{fp}} \in (0, 1]$:

$$\left\|\overline{\mathbf{X}}_{(k,l+1)} - \overline{\mathbf{X}}_{(k,l)}\right\|_F \leqslant \varepsilon_{\text{fp}} \max\left(1, \left\|\overline{\mathbf{X}}_{(k,l)}\right\|_F\right). \tag{9}$$

The number of iterations $N_{\text{fp}} \in \mathbb{N}$ is therefore defined as the smallest integer such that the stopping criterion (9) is satisfied. In addition, the method is relaxed, with a factor $\lambda \in (0, 1]$, so that the iteration defines a contraction function and converges toward a fixed-point. The resulting solver is described in Algorithm 1. In the following, the target accuracy $\varepsilon = \varepsilon_{\text{fp}} = \varepsilon_{\text{gmres}}$ of Algorithms 3 and 1 is set to $\varepsilon = 10^{-6}$ in order to provide sufficient accuracy in the numerical experiments presented in Section 5. Furthermore, the rounding accuracy (which must be less than the target accuracy) is taken as $\eta_{\text{trunc}} = 10^{-8}$ to be small compared to ε, and the relaxation factor is set to $\lambda = 0.5$.

Algorithm 1 Truncated relaxed fixed-point iteration method

Input: $\boldsymbol{\mathcal{A}} : \mathbb{R}^{N_x \times N_\theta} \to \mathbb{R}^{N_x \times N_\theta}$, $\overline{\mathbf{X}}_{k-1} \in \mathbb{R}^{N_x \times N_\theta}$, $\overline{\mathbf{B}}_k \in \mathbb{R}^{N_x \times N_\theta}$, $\boldsymbol{\mathcal{G}} : \mathbb{R}^{N_x \times N_\theta} \to \mathbb{R}^{N_x \times N_\theta}$, $\mathbf{P} \in \mathbb{R}^{N_x \times N_x}$, $\lambda \in (0, 1]$, $\varepsilon_{\text{fp}} \in (0, 1]$, $\varepsilon_{\text{gmres}} \in (0, 1]$, and $\eta_{\text{trunc}} \in [0, 1]$

Output: $\overline{\mathbf{X}}_k \in \mathbb{R}^{N_x \times N_\theta}$

$\overline{\mathbf{X}}_{(k,0)} = \overline{\mathbf{X}}_{k-1}$
for $l = 1, \ldots, N_{\text{fp}}$ **do**
  Find $\overline{\mathbf{X}}_* \in \mathbb{R}^{N_x \times N_\theta}$ solution to $\boldsymbol{\mathcal{A}}(\overline{\mathbf{X}}_*) = \overline{\mathbf{B}}_k + \boldsymbol{\mathcal{G}}(\overline{\mathbf{X}}_{(k,l-1)})$
  $\overline{\mathbf{X}}_{(k,l)} = \boldsymbol{\mathcal{T}}(\lambda \overline{\mathbf{X}}_{(k,l-1)} + (1 - \lambda)\overline{\mathbf{X}}_*)$
end
$\overline{\mathbf{X}}_k = \overline{\mathbf{X}}_{(k,N_{\text{fp}})}$

Remark 2 The convergence rate of Algorithm 1 can also be improved by using convergence acceleration techniques, such as Aitken's delta-squared process.

Remark 3 In this work, we will focus on quadratic polynomial nonlinearities (i.e., $\boldsymbol{\mathcal{G}}(\mathbf{X}_k) = \mathbf{E}(\boldsymbol{\theta})[\mathbf{X}_k \odot \mathbf{X}_k]$, where $\mathbf{E}(\boldsymbol{\theta}) \in \mathbb{R}^{N_x \times N_x}$, $\mathbf{X}_k \odot \mathbf{X}_k \in \mathbb{R}^{N_x \times N_\theta}$, and $\odot$ stands for the Hadamard product), for which computations remain tractable. Specifically, the product of two tensors, with rank r_1 and r_2, results in a tensor of rank $r_1 \times r_2$. The latter can therefore be evaluated at an affordable cost if the ranks r_1, r_2 are low. To this end, a rounding operator can be used to promote low-rank representations. However, this approach generally becomes computationally prohibitive when the nonlinearity is either a high-degree polynomial or nonpolynomial. In these cases, hyper-reduction methods [55, 56] could be employed, and their design should be tailored to the specific problem at hand. This will be the subject of further investigations.

4 Interpolator

In Section 3, we described a low-rank solver to evaluate the solution corresponding to different parameters $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_\theta}\}$ for a single time-step t_k. However, calculating the acceptance ratio in the Metropolis-Hastings algorithm requires the solution for all time-steps $\{t_1, \ldots, t_k\}$. For this reason, we extend now the method to time-dependent problems. To this end, we employ an interpolator, instead of the previous low-rank solver, to significantly reduce the solution time requirement. This interpolator is a reduced-oder model (ROM) based on the POD-Galerkin method, where the reduced basis is updated each time a new observation is available. The step in the sequential MCMC at which the observation $\mathbf{z}_{\text{obs}}(t_k)$ is received is denoted by $k \in \{1, \ldots, N_t\}$. The reason for this update is the following: as the parameter sampling is progressively adapted by the MCMC, the set of solutions to be approximated changes. This update therefore aims to provide a low-dimensional basis for accurately interpolating the solution defined on the new time parameters domain of interest.

4.1 Reduced-order approximation

At each step k, the solution $\mathbf{x}(t_m;\ \widehat{\boldsymbol{\theta}})$ is approximated for $m \in \{1, \ldots, k\}$ by

$$\widetilde{\mathbf{x}}(t_m;\ \widehat{\boldsymbol{\theta}}) = \sum_{n=1}^{s_k} \mathbf{w}_n^k y_n(t_m;\ \widehat{\boldsymbol{\theta}}) = \mathbf{W}_k \mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}),$$

where $\mathbf{W}_k = [\mathbf{w}_1^k, \ldots, \mathbf{w}_{s_k}^k] \in \mathbb{R}^{N_x \times s_k}$ denotes the reduced basis, $\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}) = [y_1(t_m;\ \widehat{\boldsymbol{\theta}}), \ldots, y_{s_k}(t_m;\ \widehat{\boldsymbol{\theta}})] \in \mathbb{R}^{s_k}$ denote the reduced coordinates of the solution in this basis, and $s_k \ll N_x$.

4.2 Proper orthogonal decomposition

At the beginning of step k, the reduced basis $\mathbf{W}_{k-1}$ is updated in order to approximate the solution corresponding to the new time-parameter domain $[t_1, t_k] \times \mathcal{D}_k$, where $\mathcal{D}_k$ is the convex hull of $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_\theta}\}$. For this purpose, the snapshot database is defined as

$$\bigcup_{m=1}^{k} \left\{\mathbf{x}(t_m;\ \boldsymbol{\theta}_1), \ldots, \mathbf{x}(t_m;\ \boldsymbol{\theta}_{N_\theta})\right\},$$

where the solutions $\{\mathbf{x}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{x}(t_k;\ \boldsymbol{\theta}_{N_\theta})\}$ are evaluated using the low-rank solver, and the solutions $\{\mathbf{x}(t_m;\ \boldsymbol{\theta}_1), \ldots, \mathbf{x}(t_m;\ \boldsymbol{\theta}_{N_\theta})\}_{m=1}^{k-1}$ are approximated by the interpolator using $\mathbf{W}_{k-1}$. The corresponding snapshot matrix is therefore given by

$$\mathbf{S} = \left[\mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T \ \ \mathbf{W}_{k-1} \mathbf{Y}_{k-1}\right] \in \mathbb{R}^{N_x \times k N_\theta},$$

where $\mathbf{Y}_{k-1} = [\mathbf{y}(t_1;\ \boldsymbol{\theta}_1), \ldots, \mathbf{y}(t_1;\ \boldsymbol{\theta}_{N_\theta}), \ldots, \mathbf{y}(t_{k-1};\ \boldsymbol{\theta}_1), \ldots, \mathbf{y}(t_{k-1};\ \boldsymbol{\theta}_{N_\theta})]$. The reduced basis $\mathbf{W}_k$ is then build by Proper Orthogonal Decomposition (POD) [57, 58] in order to obtain the approximation subspace minimizing, in the least-squares sense, the difference between the snapshots and their projections onto this subspace. In particular, the POD is efficiently computed using the SVD update of Brand described in Algorithm 4. Moreover, the dimension $s_k \in \mathbb{N}$ of the reduced basis is defined as the smallest integer such that the relative projection error is less than a given tolerance $\eta_{\text{proj}} \in [0, 1]$:

$$\left\|\mathbf{S} - \mathbf{W}_k \mathbf{W}_k^T \mathbf{S}\right\|_F \leqslant \eta_{\text{proj}} \left\|\mathbf{S}\right\|_F .$$

In the following, the projection accuracy is chosen as $\eta_{\text{proj}} = \varepsilon = 10^{-6}$ (which corresponds to 99.9999% of the singular value energy) to be of the same order of magnitude as the target accuracy ε. Furthermore, the approximation of the different variables is decorrelated since they can have different orders of magnitude. For example, in the FSI problem described in Section 5.2, the variables $\{\boldsymbol{u}, p, \boldsymbol{d}, \dot{\boldsymbol{d}}\}$ are respectively approximated by $\{\mathbf{W}_k^{\boldsymbol{u}}, \mathbf{W}_k^{p}, \mathbf{W}_k^{\boldsymbol{d}}, \mathbf{W}_k^{\dot{\boldsymbol{d}}}\}$, and the reduced basis is defined as

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{W}_k^{\boldsymbol{u}} & & & \\ & \mathbf{W}_k^{p} & & \\ & & \mathbf{W}_k^{\boldsymbol{d}} & \\ & & & \mathbf{W}_k^{\dot{\boldsymbol{d}}} \end{bmatrix}.$$

4.3 Galerkin method

Once the reduced basis is updated, the ROM can finally be exploited to accelerate the Metropolis-Hastings algorithm. To this end, the reduced coordinates $\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}})$ are determined for $m \in \{1, \ldots, k\}$ by the Galerkin method to provide robust physic-based interpolations:

$$\widetilde{\mathbf{A}}(\widehat{\boldsymbol{\theta}})\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}) = \widetilde{\mathbf{b}}(t_m;\ \widehat{\boldsymbol{\theta}}) + \widetilde{\mathbf{g}}(\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}), t_m;\ \widehat{\boldsymbol{\theta}}), \tag{10}$$

where $\widetilde{\mathbf{A}}(\widehat{\boldsymbol{\theta}}) = \mathbf{W}_k^T \mathbf{A}(\widehat{\boldsymbol{\theta}})\mathbf{W}_k \in \mathbb{R}^{s_k \times s_k}$, $\widetilde{\mathbf{b}}(t_m;\ \widehat{\boldsymbol{\theta}}) = \mathbf{W}_k^T \mathbf{b}(t_m;\ \widehat{\boldsymbol{\theta}}) \in \mathbb{R}^{s_k}$, and $\widetilde{\mathbf{g}}(\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}), t_m;\ \widehat{\boldsymbol{\theta}}) = \mathbf{W}_k^T \mathbf{g}(\mathbf{W}_k \mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}), t_m;\ \widehat{\boldsymbol{\theta}}) \in \mathbb{R}^{s_k}$.

Remark 4 As stated previously, we will focus in the following on quadratic polynomial nonlinearities (i.e., $\widetilde{\mathbf{g}}(\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}), t_m;\ \widehat{\boldsymbol{\theta}}) = \widetilde{\mathbf{E}}(\widehat{\boldsymbol{\theta}})[\mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}}) \otimes \mathbf{y}(t_m;\ \widehat{\boldsymbol{\theta}})]$, where $\widetilde{\mathbf{E}}(\widehat{\boldsymbol{\theta}}) = \mathbf{W}_k^T \mathbf{E}(\widehat{\boldsymbol{\theta}})[\mathbf{w}_1^k \odot \mathbf{W}_k, \ldots, \mathbf{w}_{s_k}^k \odot \mathbf{W}_k] \in \mathbb{R}^{s_k \times s_k^2}$, and $\otimes$ stands for the Kronecker product). The associated system (10) therefore scales with the dimension of the reduced basis $s_k \ll N_x$, which typically enables a speedup factor of several orders of magnitude compared to the resolution of system (7). Additionally, hyper-reduction techniques [55, 56, 59] can also be employed to further reduce the computational cost associated with the solution of equation (10). Lastly, it is important to notice that, according to (6), the matrix $\widetilde{\mathbf{A}}(\widehat{\boldsymbol{\theta}})$ (and similarly $\widetilde{\mathbf{E}}(\widehat{\boldsymbol{\theta}})$) can be efficiently computed

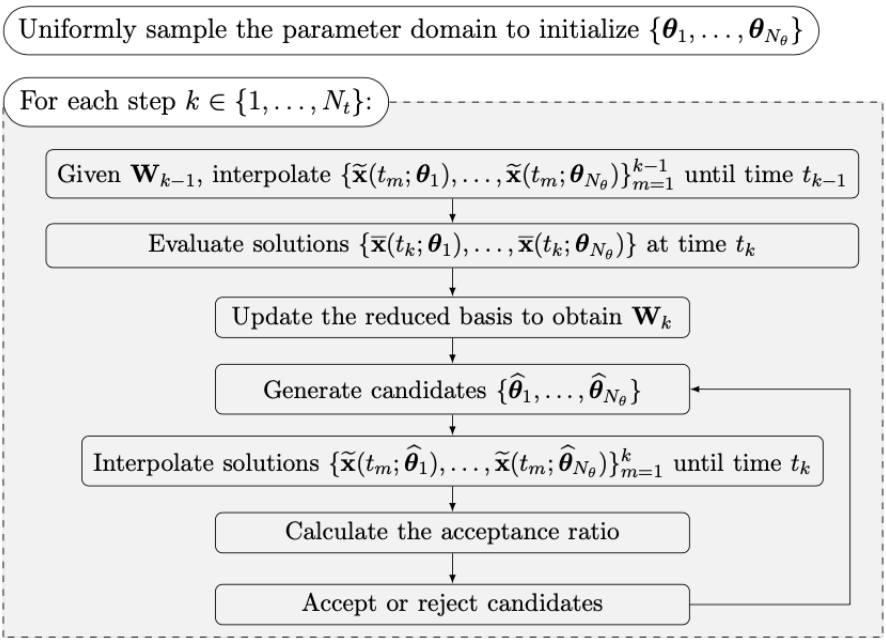


Figure 1: Workflow of the proposed method.

since

$$\widetilde{\mathbf{A}}(\widehat{\boldsymbol{\theta}}) = \sum_{n=1}^{N_\alpha} \alpha_n(\widehat{\boldsymbol{\theta}}) \widetilde{\mathbf{A}}_n,$$

where the matrices $\widetilde{\mathbf{A}}_n = \mathbf{W}_k^T \mathbf{A}_n \mathbf{W}_k$ are precomputed once-for-all during the update of the reduced basis.

Remark 5 In the numerical experiments presented in Section 5, the resulting ROM-based interpolator will inherit the stability property of the numerical methods used to obtain the discrete system (7).

5 Numerical experiments

In this section, the performance of the proposed method is assessed for four applications. In each case, we perform parameter estimation and uncertainty quantification using noisy partial observations generated from the ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ (i.e., $\mathbf{z}_{\text{obs}}(t_k) = \mathbf{z}(t_k;\ \boldsymbol{\theta}_{\text{obs}})$). The accuracy of the estimate $\boldsymbol{\mu}_{\mathbf{q}}(t_k)$ with respect to the associated QoI, $\mathbf{q}(t_k;\ \boldsymbol{\theta}_{\text{obs}})$, is evaluated using the

relative space-time error:

$$\text{Error} = \sqrt{\frac{\sum_{k=1}^{N_t} \|\mathbf{q}(t_k;\ \boldsymbol{\theta}_{\text{obs}}) - \boldsymbol{\mu}_{\mathbf{q}}(t_k)\|_2^2}{\sum_{k=1}^{N_t} \|\mathbf{q}(t_k;\ \boldsymbol{\theta}_{\text{obs}})\|_2^2}}.$$

Furthermore, the computational speedup provided by the proposed method with respect to a sequential MCMC method using a standard finite element solver is evaluated in each case to quantify the reduction in computational cost for sampling the posterior distribution function.

5.1 Heat conduction in 3D

In the first application, we consider a three-dimensional linear heat conduction problem. Let $\Omega \subset \mathbb{R}^3$ be an open set representing the physical domain with Γ_{int} and Γ_{ext}, the interior and exterior boundaries, respectively. The temperature $T : \Omega \times \mathbb{R}_+ \to \mathbb{R}$ is governed by the heat equation:

$$\begin{cases} \partial_t T - \kappa \Delta T = 0 & \text{in} \quad \Omega \times \mathbb{R}_+, \\ T = 0 & \text{in} \quad \Omega \times \{0\}, \\ \kappa \dfrac{\partial T}{\partial n} + \nu (T - T_{\text{bc}}) = 0 & \text{on} \quad \Gamma_{\text{int}} \times \mathbb{R}_+, \\ \dfrac{\partial T}{\partial n} = 0 & \text{on} \quad \Gamma_{\text{ext}} \times \mathbb{R}_+, \end{cases} \tag{11}$$

where κ denotes the thermal diffusivity constant, ν denotes the Robin parameter, and the boundary condition on Γ_{int} is defined as

$$T_{\text{bc}}(\boldsymbol{x}, t) = c_1 + c_2 \cos(\varphi) + c_3 \sin(\varphi) + c_4 \cos(2\varphi) + c_5 \sin(2\varphi)$$

with $\varphi = \text{atan}(y/x)$. The physical domain is discretized using 35,328 vertices and 186,624 tetrahedral elements as shown in Figure 2. We consider the time-interval $[0, 10]$, and the time-step size is $\Delta t = 10^{-1}$ (i.e., $N_t = 100$). The heat equation (11) is then discretized by continuous $\mathbb{P}_1$ finite elements in space and the implicit Euler scheme in time, which leads to a linear discrete system (3) having $N_x = 35,328$ DOFs. The input parameters are $\boldsymbol{\theta} = [\kappa, \nu, c_1, c_2, c_3, c_4, c_5]$, and we consider the parameter domain given in Table 1.

κ	ν	c_1	c_2	c_3	c_4	c_5
$[1, 10]$	$[1, 10]$	$[5, 15]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$

Table 1: Parameter domain $\mathcal{D}$ for the heat conduction problem.

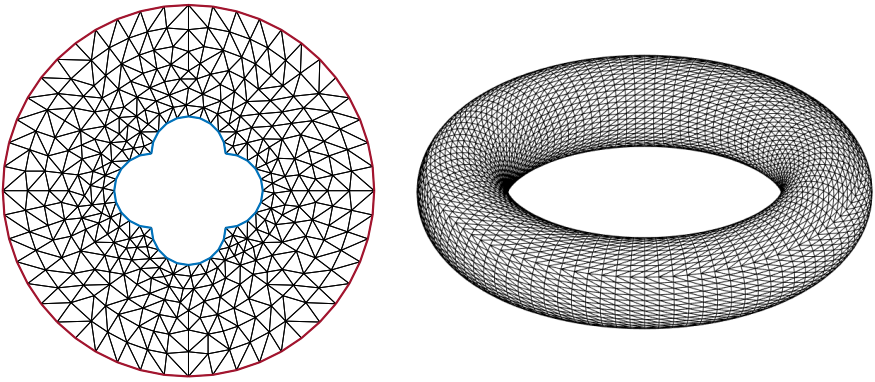
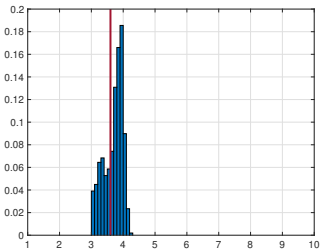
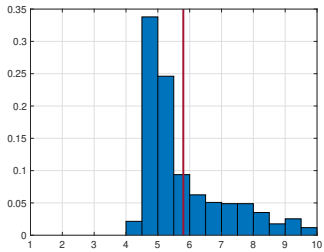


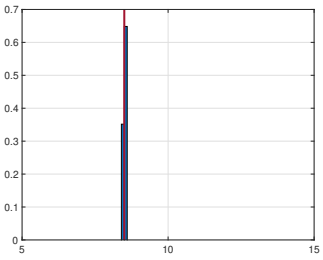
Figure 2: Left: cross section of the domain Ω with boundaries Γ_{int} and Γ_{ext} plotted in blue and red, respectively. Right: computational mesh for the heat conduction problem.



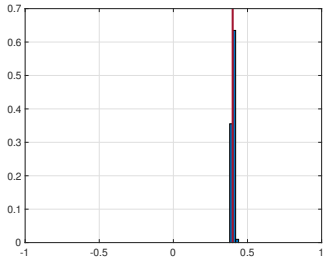
(a) κ



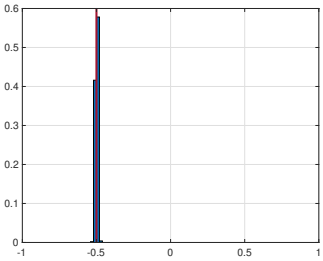
(b) ν



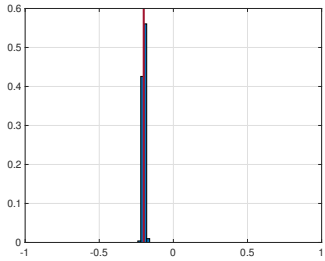
(c) c_1



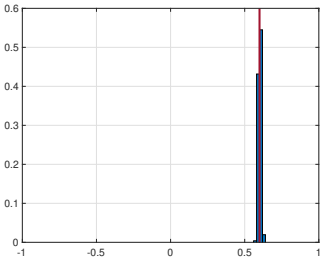
(d) c_2



(e) c_3



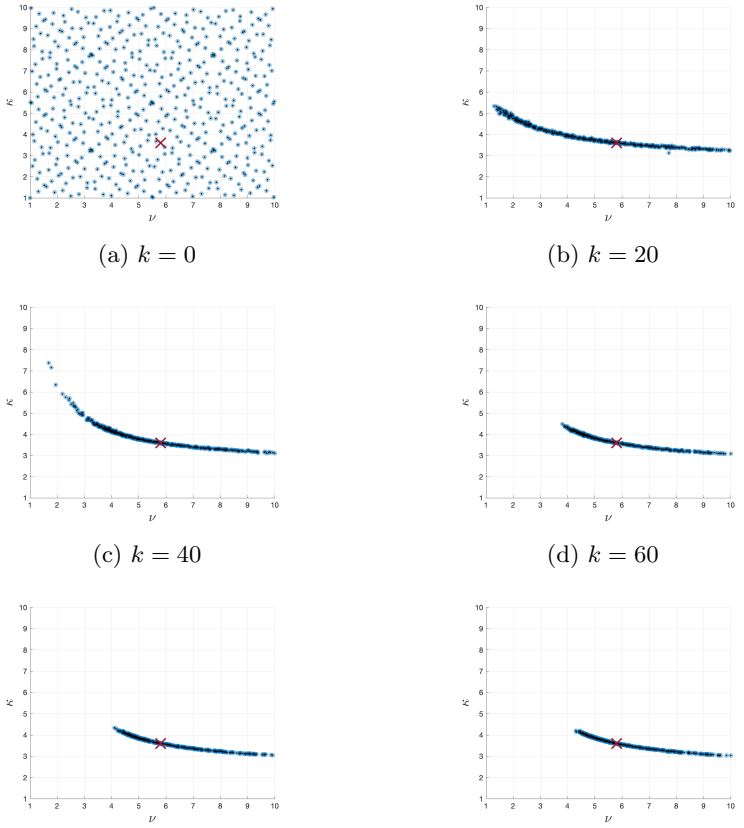
(f) c_4



(g) c_5

Figure 3: Parameter estimation resulting from the sampling of the posterior probability distribution at final step (i.e., $k = N_t$) for the heat conduction problem. The ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.

This application aims at estimating the model parameters by exploiting measurements taken on the exterior surface Γ_{ext}, which thereafter enables to reconstruct the temperature on the interior surface Γ_{int} (so that the QoI, $\mathbf{q}$, is here $T|_{\Gamma_{\text{int}}}$). These observations contain 5% of noise and were generated from the ground truth parameter $\boldsymbol{\theta}_{\text{obs}} = [3.6, 5.8, 8.5, 0.4, -0.5, -0.2, 0.6]$. Figure 3 shows the sampling of the posterior probability distribution obtained using $N_\theta = 512$ queried input parameters. The parameters $\{c_1, c_2, c_3, c_4, c_5\}$ are accurately estimated, while the values of $\{\kappa, \nu\}$ appear to be correlated. Specifically, only one of them is identifiable, the other being given by the hyperbolic-like relation that can be identified from the sampling of the posterior probability distribution displayed in Figure 4. This illustrates in particular the robustness of the method against the existence of local minima. In Figure 5, we present the estimation of the temperature on Γ_{int} at final time resulting from this sampling. The standard deviation $\boldsymbol{\sigma}_{\mathbf{q}}$ is two orders of magnitude smaller than the mean $\boldsymbol{\mu}_{\mathbf{q}}$, leading to robust confidence intervals. In particular, the relative space-time error between the ground truth temperature and the estimate $\boldsymbol{\mu}_{\mathbf{q}}$ delivered by the proposed method is 0.533%. Figure 6 reports the evolution of the solution rank and reduced basis dimension. At the beginning, the dimension of the reduced basis increases due to the new solution features to be approximated, contained in the snapshots provided by the low-rank solver. It then decreases as the sampling of the posterior probability distribution starts to converge and becomes less spread out. This demonstrates the ability of the interpolator to accurately approximate the solutions associated with the adaptive sampling of the parameter domain, while maintaining the reduced basis dimension as low as possible to speedup computations. In particular, the computational speedup factor delivered by the proposed method compared to a sequential MCMC method using a standard finite element solver for solving system (3) is 887. Of the time required for sampling the posterior probability distribution, 83.79% is spent in the low-rank solver, 0.34% in updating the reduced basis and associated precomputed quantities, 15.26% in the solution interpolation, and 0.61% in the other parts of the method.

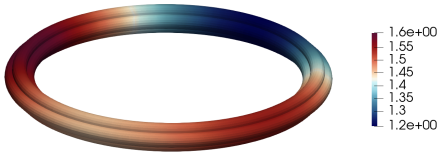


(a) $k = 0$ (b) $k = 20$

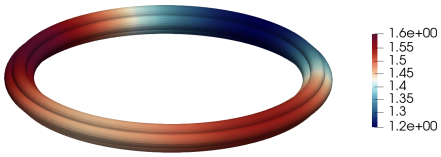
(c) $k = 40$ (d) $k = 60$

(e) $k = 80$ (f) $k = 100$

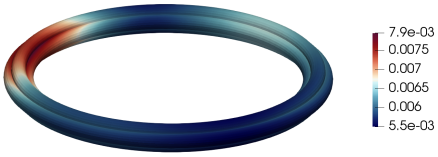
Figure 4: Sampling of the posterior probability distribution at different steps k for the heat conduction problem. The ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.



(a) Ground truth temperature T on Γ_{int}



(b) Mean $\boldsymbol{\mu_{\mathbf{q}}}$



(c) Standard deviation $\boldsymbol{\sigma_{\mathbf{q}}}$

Figure 5: Estimation of the temperature on the interior surface at final time $t_k = 10$ for the heat conduction problem.

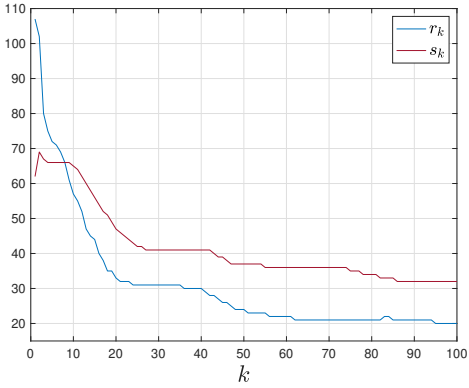


Figure 6: Solution rank and reduced basis dimension at different steps $k \in \{1, \ldots, N_t\}$ for the heat conduction problem.

5.2 Pressure-wave propagation in 2D

In the second application, we consider the well-known linear fluid-structure interaction (FSI) benchmark of a two-dimensional pressure-wave propagation

(see, e.g., [60, 61]). For a given length L, radius R and thickness ϵ, let $\Omega_\mathrm{f} := (0, L) \times (0, R) \subset \mathbb{R}^2$ and $\Omega_\mathrm{s} := (0, L) \times (R, R + \epsilon) \subset \mathbb{R}^2$ be, respectively, the open sets representing the fluid and solid domains, and $\Sigma = (0, L) \times \{R\}$ be the interface between Ω_f and Ω_s. The fluid and solid boundaries are partitioned as follows $\partial\Omega_\mathrm{f} = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Sigma$ and $\partial\Omega_\mathrm{s} = \Gamma_\mathrm{d} \cup \Gamma_\mathrm{n} \cup \Sigma$ (see Figure 7).

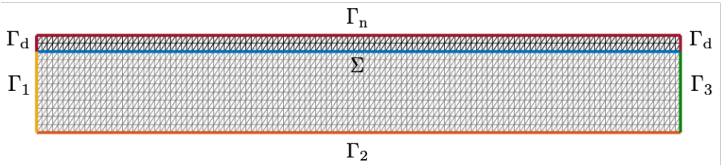


Figure 7: Computational mesh for the pressure-wave propagation problem: the fluid domain Ω_f is plotted in gray, the solid domain Ω_s in black, and the interface Σ in blue.

We assume that the solid undergoes infinitesimal displacements, so that the fluid velocity $\boldsymbol{u} : \Omega_\mathrm{f} \times \mathbb{R}_+ \to \mathbb{R}^2$ and pressure $p : \Omega_\mathrm{f} \times \mathbb{R}_+ \to \mathbb{R}$ are governed by the Stokes equations

$$\begin{cases} \rho_\mathrm{f} \partial_t \boldsymbol{u} - \operatorname{div} \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p) = 0 \quad \text{in} \quad \Omega_\mathrm{f} \times \mathbb{R}_+, \\ \operatorname{div} \boldsymbol{u} = 0 \quad \text{in} \quad \Omega_\mathrm{f} \times \mathbb{R}_+, \end{cases} \tag{12a}$$

and the solid displacement $\boldsymbol{d} : \Omega_\mathrm{s} \times \mathbb{R}_+ \to \mathbb{R}^2$ and velocity $\dot{\boldsymbol{d}} : \Omega_\mathrm{s} \times \mathbb{R}_+ \to \mathbb{R}^2$ by the linear elastodynamics equations

$$\begin{cases} \rho_\mathrm{s} \partial_t \dot{\boldsymbol{d}} - \operatorname{div} \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d}) + \beta \boldsymbol{d} = 0 \quad \text{in} \quad \Omega_s \times \mathbb{R}_+, \\ \dot{\boldsymbol{d}} = \partial_t \boldsymbol{d} \quad \text{in} \quad \Omega_s \times \mathbb{R}_+. \end{cases} \tag{12b}$$

In the above relations, the term $\beta \boldsymbol{d}$ represents the transverse membrane effects that appear in axisymmetric formulations, ρ_f, ρ_s, $\boldsymbol{\sigma}_\mathrm{f}$ and $\boldsymbol{\sigma}_\mathrm{s}$ denote respectively the fluid and solid densities and the fluid and solid stress tensors. The latter are given by the constitutive relations

$$\boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p) := 2\mu_\mathrm{f} \boldsymbol{\epsilon}(\boldsymbol{u}) - p\mathbf{I}, \quad \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d}) := 2\mu_\mathrm{s} \boldsymbol{\epsilon}(\boldsymbol{d}) + \lambda_\mathrm{s} (\operatorname{div} \boldsymbol{d}) \mathbf{I}.$$

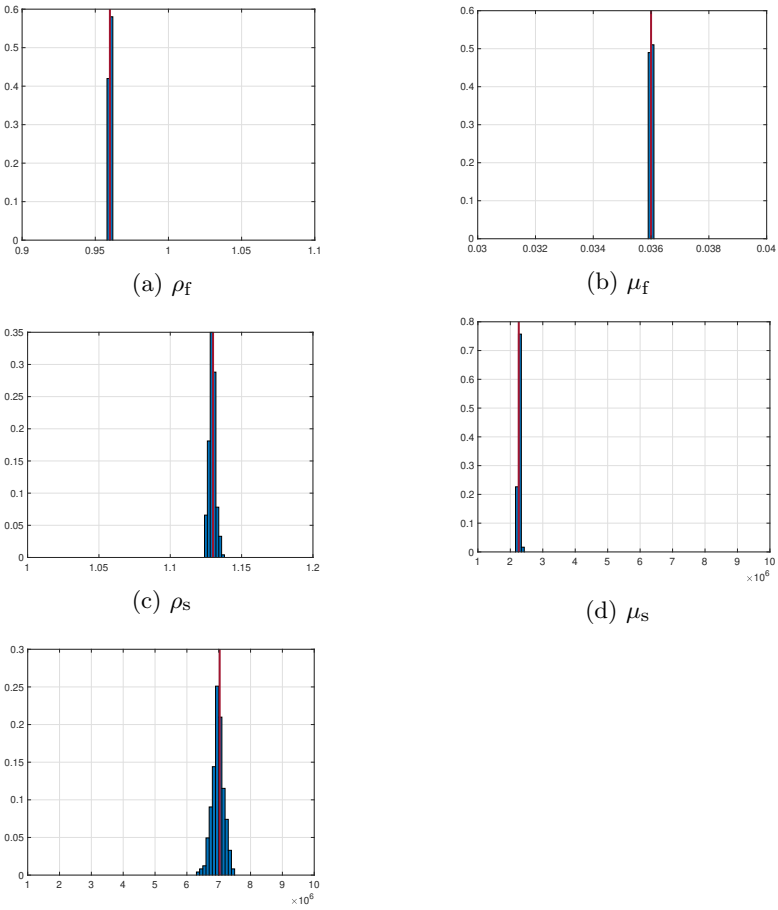
Here, μ_f stands for the fluid dynamics viscosity, μ_s and λ_s for the Lamé parameters of the solid, and $\boldsymbol{\epsilon}(\boldsymbol{u}) := \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\mathrm{T})$ denotes the symmetric part of the gradient. Finally, the interface coupling conditions between the fluid and solid subproblems (12a) and (12b) are given by

$$\begin{cases} \boldsymbol{u} = \dot{\boldsymbol{d}} \quad \text{on} \quad \Sigma \times \mathbb{R}_+, \\ \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p) \boldsymbol{n} = \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d}) \boldsymbol{n} \quad \text{on} \quad \Sigma \times \mathbb{R}_+, \end{cases} \tag{12c}$$

ρ_{f}	μ_{f}	ρ_{s}	μ_{s}	λ_{s}
$[0.9, 1.1]$	$[0.03, 0.04]$	$[1, 1.2]$	$[10^6, 10^7]$	$[10^6, 10^7]$

Table 2: Parameter domain $\mathcal{D}$ considered for the pressure-wave propagation problem.

where $\boldsymbol{n}$ denotes the outward unit normal to $\partial\Omega_{\mathrm{f}}$. The coupled problem (12) is completed with zero initial conditions on $\boldsymbol{u}$, $\boldsymbol{d}$ and $\dot{\boldsymbol{d}}$. A sinusoidal pressure profile $p_{\mathrm{in}}(t) = p_{\max}(1 - \cos(2t\pi/T^*))/2$, with maximum pressure $p_{\max} = 2 \times 10^4$, is prescribed on the inlet boundary Γ_1 during $T^* = 0.005$ seconds. Zero traction is imposed on Γ_3, and a slip (symmetry) condition is enforced on Γ_2. For the solid, we set $\boldsymbol{d} = \boldsymbol{0}$ on the extremities Γ_{d} and we impose zero traction on Γ_{n}.



(a) ρ_{f} (b) μ_{f}

(c) ρ_{s} (d) μ_{s}

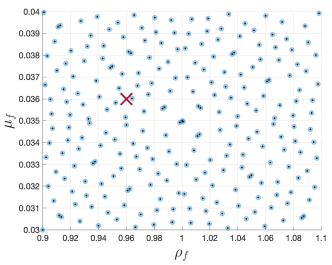
(e) λ_{s}

Figure 8: Parameter estimation resulting from the sampling of the posterior probability distribution at final step (i.e., $k = N_t$) for the pressure-wave propagation problem. The ground truth parameter $\boldsymbol{\theta}_{\mathrm{obs}}$ is plotted in red.

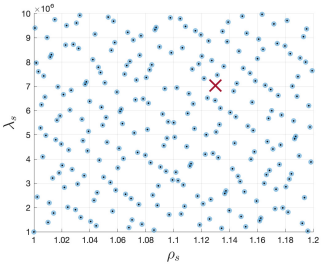
For the time discretization of (12), we consider a fully implicit method (strongly coupled scheme), using a backward Euler scheme in the fluid (12a) and a neutrally stable mid-point scheme for the solid (12b). For the discretization in space, we consider continuous $\mathbb{P}_1$ finite elements, with a Nitsche based treatment of the interface conditions (see [62, Section 4.1]), which yields a simple block structure in the resulting system matrix. To cope with the lack of inf-sup stability in the fluid approximation, the Brezzi-Pitkäranta pressure stabilization is considered (see [63]), with a stabilization parameter $\delta = 10^{-3}$. We consider an example with $L = 6$, $R = 0.5$, and $\epsilon = 0.1$. All the units are given in the CGS sytem. The fluid (resp. solid) domain is then discretized using

1,331 (resp. 363) vertices and 2,400 (resp. 480) triangular elements as shown in Figure 7. We consider the time-interval $[0, 0.015]$, and the time-step size is $\Delta t = 10^{-4}$ (i.e., $N_t = 150$). The Nitsche penalty parameter is set to $\gamma = 100$, and the solid parameter β is set to 4×10^6. The resulting linear discrete system (3) has $N_x = 5,445$ DOFs (that is, $2 \times 1,331$ for $\boldsymbol{u}$ + $1,331$ for p + 2×363 for $\boldsymbol{d}$ + 2×363 for $\dot{\boldsymbol{d}}$). The input parameters are $\boldsymbol{\theta} = [\rho_\mathrm{f}, \mu_\mathrm{f}, \rho_s, \mu_\mathrm{s}, \lambda_\mathrm{s}]$, and we consider the parameter domain given in Table 2.

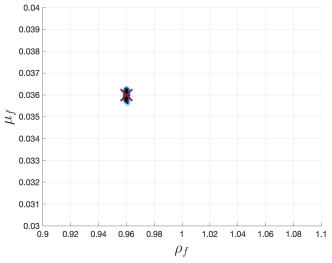
This application focuses on the estimation of the fluid pressure p from measurements of the fluid velocity $\boldsymbol{u}$ and solid velocity $\dot{\boldsymbol{d}}$ in the whole physical domain. This is type of data typically available in 4D-flow MRI data. These observations were generated from the ground truth parameter $\boldsymbol{\theta}_\mathrm{obs} = [0.96, 0.036, 1.13, 2.26 \cdot 10^6, 7.03 \cdot 10^6]$ and contain 5% of noise. Figure 8 shows the sampling of the posterior probability distribution obtained using $N_\theta = 243$ queried input parameters. The parameters $\{\rho_\mathrm{f}, \mu_\mathrm{f}, \rho_s, \mu_\mathrm{s}\}$ are accurately estimated, while the value of λ_s is more uncertain. In particular, the estimation converges faster for $\{\rho_\mathrm{f}, \mu_\mathrm{f}, \mu_\mathrm{s}\}$ than in the directions $\{\rho_\mathrm{s}, \lambda_\mathrm{s}\}$ as shown in Figure 9. Figure 10 presents the pressure estimate at final time resulting from this sampling. The standard deviation $\boldsymbol{\sigma_\mathbf{q}}$ is three orders of magnitude smaller than the mean $\boldsymbol{\mu_\mathbf{q}}$, leading to robust confidence intervals. Notably, the relative space-time error between the ground truth pressure and the estimate $\boldsymbol{\mu_\mathbf{q}}$ delivered by the proposed method is 0.008%. Figure 11 reports the evolution of the solution rank and reduced basis dimension. In particular, the computational speedup factor delivered by the proposed method compared to a sequential MCMC method using a standard finite element solver for solving system (3) is 91. Of the time required for sampling the posterior probability distribution, 77.91% is spent in the low-rank solver, 0.34% in updating the reduced basis and associated precomputed quantities, 15.26% in the solution interpolation, and 0.61% in the other parts of the method.



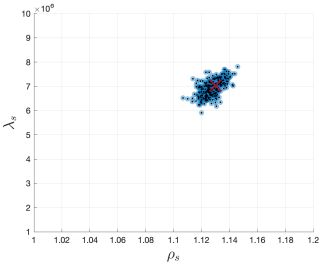
(a) $k = 0$



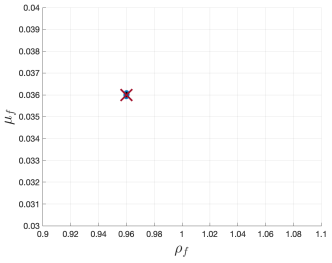
(b) $k = 0$



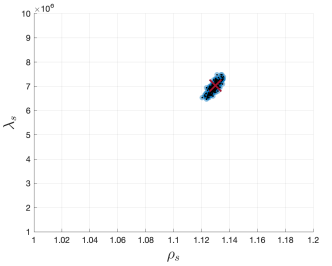
(c) $k = 50$



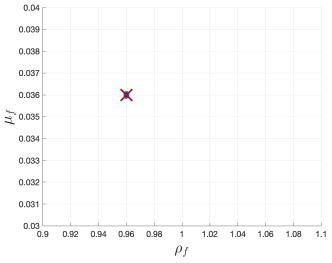
(d) $k = 50$



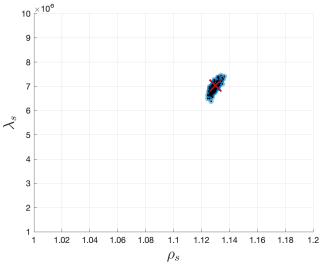
(e) $k = 100$



(f) $k = 100$

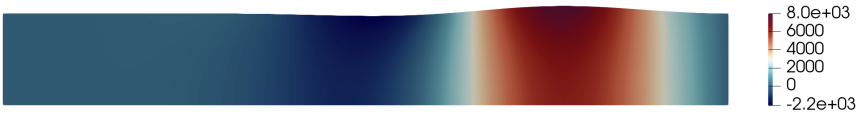


(g) $k = 150$

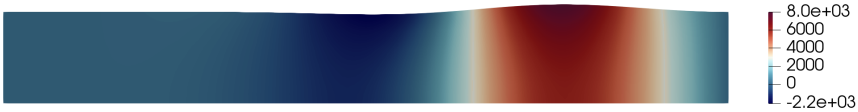


(h) $k = 150$

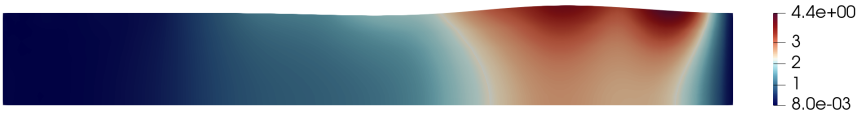
Figure 9: Sampling of the posterior probability distribution at different steps k for the pressure-wave propagation problem. The ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.



(a) Ground truth pressure p



(b) Mean $\boldsymbol{\mu}_{\mathbf{q}}$



(c) Standard deviation $\boldsymbol{\sigma}_{\mathbf{q}}$

Figure 10: Estimation of the pressure at final time $t_k = 0.015$ for the pressure-wave propagation problem.

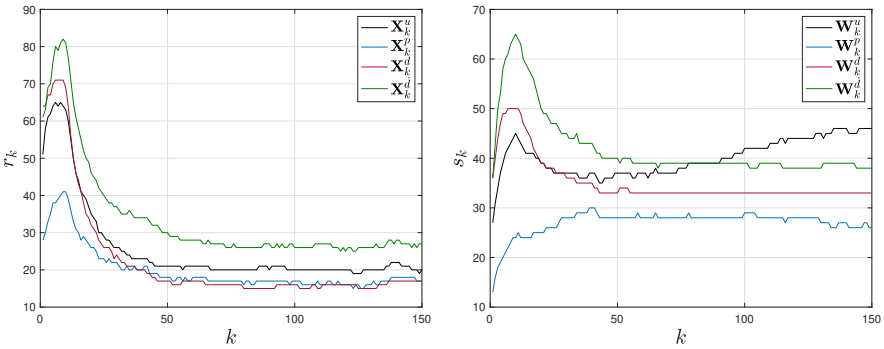


Figure 11: Solution rank and reduced basis dimension at different steps $k \in \{1, \ldots, N_t\}$ for the 2D pressure-wave propagation problem.

5.3 Flow over an elastic wall in 3D

In the third application, we consider the three-dimensional version of the previous linear FSI problem (12), with $\Omega_{\mathrm{f}}, \Omega_{\mathrm{s}} \subset \mathbb{R}^3$ given in Figure 12. The resulting FSI problem is completed with zero initial conditions. A sinusoidal pressure profile $p_{\mathrm{in}}(t) = -p_{\max} \sin(8t\pi/T^*)$, with $p_{\max} = 10^4$ and $T^* = 0.015$, is prescribed on the inlet boundary Γ_4. Zero traction is imposed on Γ_2 and a slip

ρ_{f}	μ_{f}	ρ_{s}	μ_{s}	λ_{s}
$[0.9, 1.1]$	$[0.03, 0.04]$	$[1, 1.2]$	$[10^5, 10^6]$	$[10^5, 10^6]$

Table 3: Parameter domain $\mathcal{D}$ for the 3D FSI example.

condition is enforced on $\Gamma_1 \cup \Gamma_3$. For the solid, we set $\boldsymbol{d} = 0$ on Γ_{d}. Symmetry conditions are enforced on the lateral walls.

The FSI problem (12) is discretized using the same numerical method that in Section 5.2. The fluid (resp. solid) domain is then discretized using 5,071 (resp. 330) vertices and 24,750 (resp. 1,080) tetrahedral elements as shown in Figure 12. We consider the time-interval $[0, 1]$, and the time-step size is $\Delta t = 10^{-3}$ (i.e., $N_t = 1000$). The resulting linear discrete system (3) has $N_x = 22,264$ DOFs. The input parameters are $\boldsymbol{\theta} = [\rho_{\mathrm{f}}, \mu_{\mathrm{f}}, \rho_{\mathrm{s}}, \mu_{\mathrm{s}}, \lambda_{\mathrm{s}}]$, and we consider the parameter domain given in Table 3.

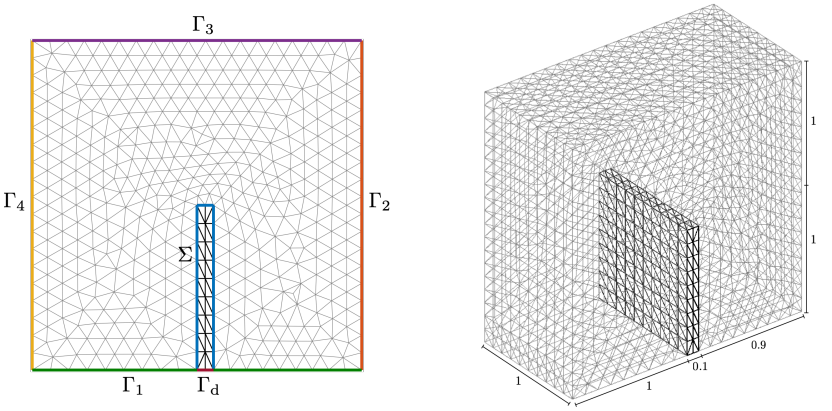
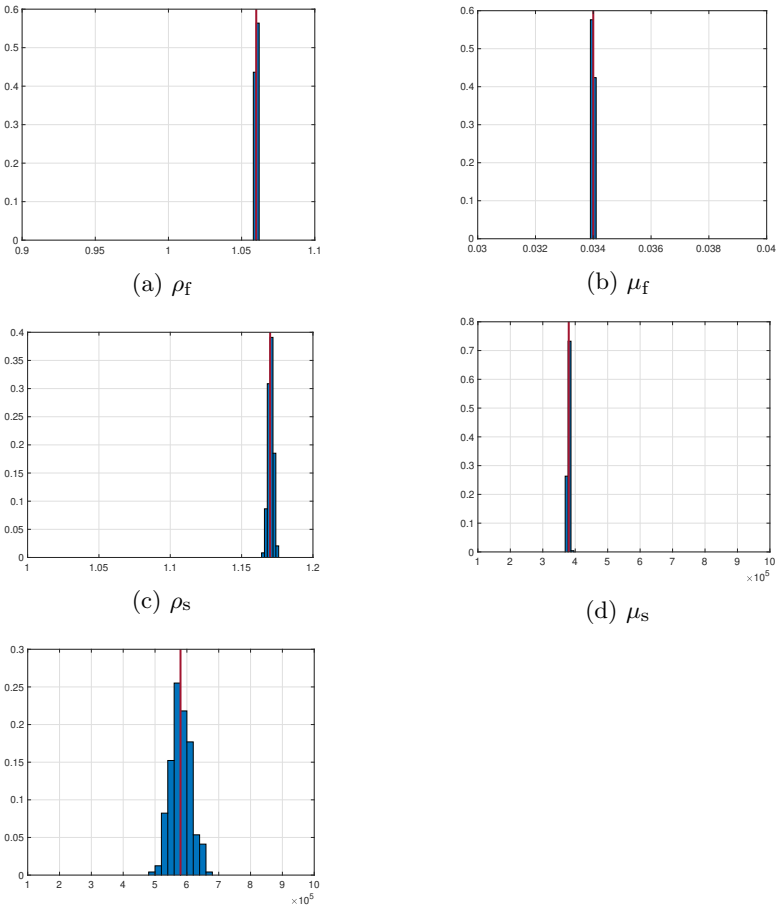


Figure 12: Left: cross section of the computational mesh with the fluid domain Ω_{f} plotted in gray, the solid domain Ω_{s} in black and, the interface Σ in blue. Right: computational mesh for the 3D FSI example.



(a) ρ_{f} (b) μ_{f}

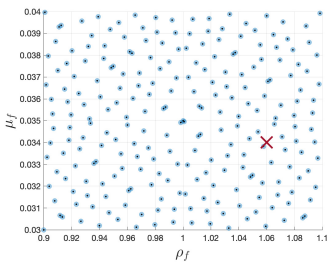
(c) ρ_{s} (d) μ_{s}

(e) λ_{s}

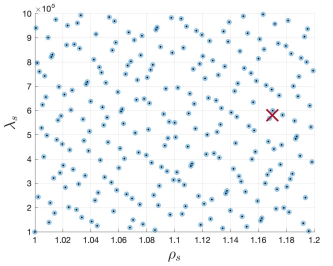
Figure 13: Parameter estimation resulting from the sampling of the posterior probability distribution at final step (i.e., $k = N_t$) for the 3D FSI example. The ground truth parameter $\boldsymbol{\theta}_{\mathrm{obs}}$ is plotted in red.

As previously, we focus on the estimation of the fluid pressure p from fluid and solid velocity measurements. These observations were generated from the ground truth parameter $\boldsymbol{\theta}_{\mathrm{obs}} = [1.06, 0.034, 1.17, 3.8 \times 10^5, 5.8 \times 10^5]$ and contain 5% of noise. Figure 13 shows the sampling of the posterior probability distribution obtained using $N_\theta = 243$ queried input parameters. The parameters $\{\rho_{\mathrm{f}}, \mu_{\mathrm{f}}, \rho_{\mathrm{s}}, \mu_{\mathrm{s}}\}$ are accurately estimated, while the value of λ_{s} is more uncertain. In particular, the estimation converges faster for $\{\rho_{\mathrm{f}}, \mu_{\mathrm{f}}, \mu_{\mathrm{s}}\}$ than in the directions $\{\rho_{\mathrm{s}}, \lambda_{\mathrm{s}}\}$ as shown in Figure 14. Figure 15 presents the pressure estimate at final time resulting from this sampling. The standard deviation $\boldsymbol{\sigma}_{\mathbf{q}}$ is two orders of magnitude smaller than the mean $\boldsymbol{\mu}_{\mathbf{q}}$, leading to

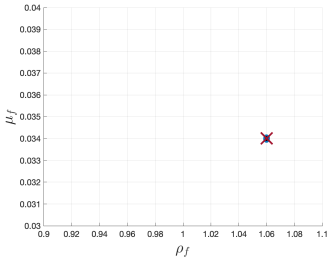
robust confidence intervals. Notably, the relative space-time error between the ground truth pressure and the estimate $\boldsymbol{\mu_q}$ delivered by the proposed method is 0.003%. Figure 16 reports the evolution of the solution rank and reduced basis dimension. In particular, the computational speedup factor delivered by the proposed method compared to a sequential MCMC method using a standard finite element solver for solving system (3) is 3,056. Of the time required for sampling the posterior probability distribution, 90.56% is spent in the low-rank solver, 1.03% in updating the reduced basis and associated precomputed quantities, 7.41% in the solution interpolation, and 1.00% in the other parts of the method.



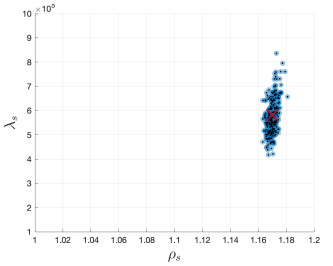
(a) $k = 0$



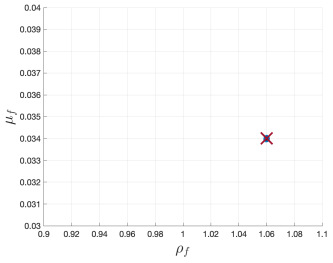
(b) $k = 0$



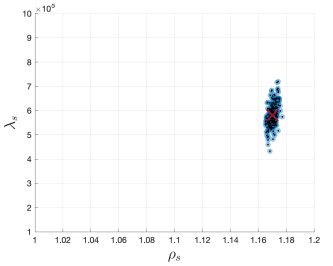
(c) $k = 333$



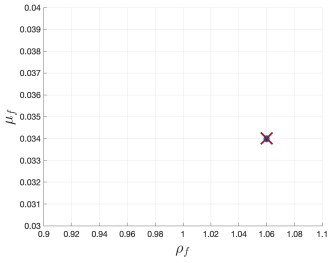
(d) $k = 333$



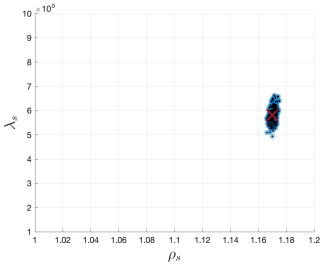
(e) $k = 667$



(f) $k = 667$

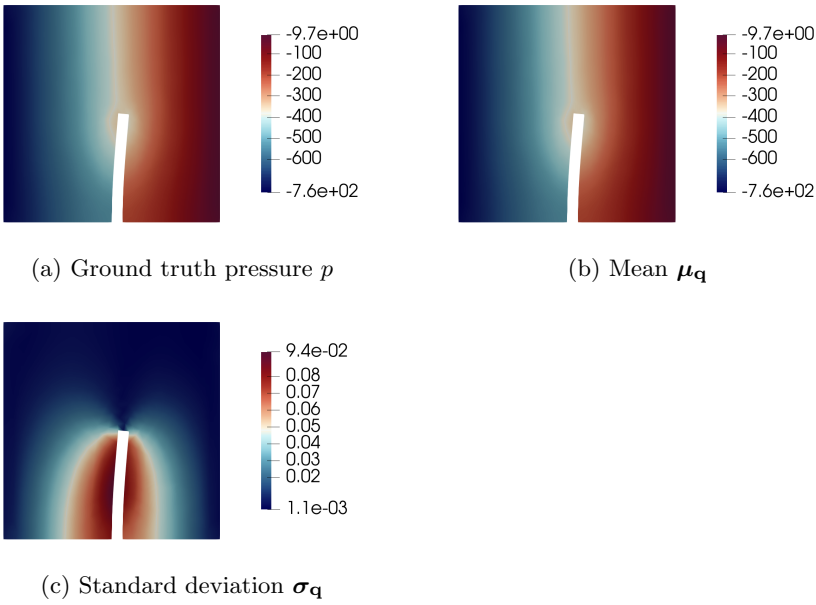


(g) $k = 1000$



(h) $k = 1000$

Figure 14: Sampling of the posterior probability distribution at different steps k for the 3D FSI example. The ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.



(a) Ground truth pressure p

(b) Mean $\boldsymbol{\mu_q}$

(c) Standard deviation $\boldsymbol{\sigma_q}$

Figure 15: Cross section of the pressure estimate at final time $t_k = 1$ for the 3D FSI example.

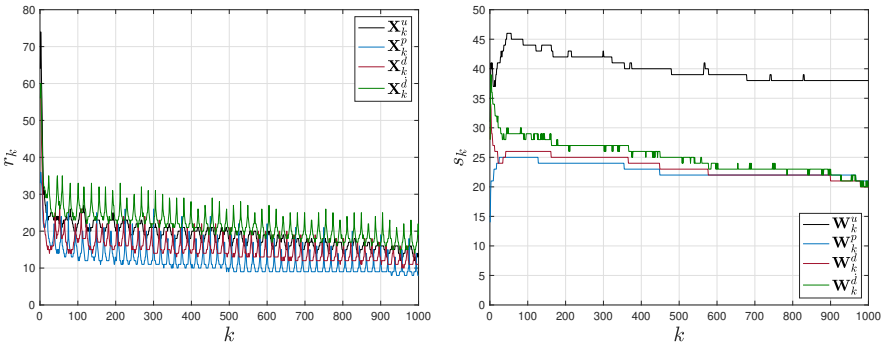


Figure 16: Solution rank and reduced basis dimension at different steps $k \in \{1, \ldots, N_t\}$ for the 3D FSI example.

5.4 Flow past a cylinder in 2D

In the last application, we consider the popular nonlinear benchmark of a two-dimensional flow past a cylinder. Let $\Omega \subset \mathbb{R}^2$ be an open set representing the physical domain with $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{cyl}} \cup \Gamma_{\text{out}}$. The fluid velocity $\boldsymbol{u} : \Omega \times \mathbb{R}_+ \to \mathbb{R}^2$

and pressure $p : \Omega \times \mathbb{R}_+ \to \mathbb{R}$ are governed by the incompressible Navier-Stokes equations

$$\begin{cases} \partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{0} & \text{in} \quad \Omega \times \mathbb{R}_+ \\ \nabla \cdot \boldsymbol{u} = \boldsymbol{0} & \text{in} \quad \Omega \times \mathbb{R}_+, \\ \boldsymbol{u} = \boldsymbol{0} & \text{in} \quad \Omega \times \{0\}, \\ \boldsymbol{u} = [u_\infty, 0] & \text{on} \quad \Gamma_{\text{in}} \times \mathbb{R}_+, \\ \boldsymbol{u} = \boldsymbol{0} & \text{on} \quad \Gamma_{\text{cyl}} \times \mathbb{R}_+, \\ \nu \partial_{\boldsymbol{n}} \boldsymbol{u} - p\boldsymbol{n} = \boldsymbol{0} & \text{on} \quad \Gamma_{\text{out}} \times \mathbb{R}_+, \end{cases} \tag{13}$$

where $u_\infty(\boldsymbol{x}, t) = 6\nu Re(H - y)y/(DH^2)$, $H = 0.41$, $D = 0.1$, ν denotes the kinematic viscosity, and Re denotes the Reynolds number. The physical domain is discretized using 1,718 vertices and 3,256 triangular elements as shown in Figure 17. We consider the time-interval $[0, 1]$, and the time-step size is $\Delta t = 10^{-3}$ (i.e., $N_t = 1000$). The Navier-Stokes equation (13) is then discretized by a continuous $\mathbb{P}_2 - \mathbb{P}_1$ finite element method in space and the implicit Euler scheme in time, which leads to a nonlinear discrete system (7) having $N_x = 15,102$ DOFs. The input parameters are $\boldsymbol{\theta} = [\nu, Re]$, and we consider the parameter domain given in Table 4.

This application focuses on the estimation of the fluid pressure p from measurements of the fluid velocity $\boldsymbol{u}$. These observations contain 5% of noise and were generated from the ground truth parameter $\boldsymbol{\theta}_{\text{obs}} = [2 \times 10^{-3}, 70]$. Figure 18 shows the sampling of the posterior probability distribution obtained using $N_\theta = 128$ queried input parameters. In particular, the two parameters $\{\nu, Re\}$ are accurately estimated. Figure 20 presents the pressure estimate at final time resulting from this sampling. The standard deviation $\boldsymbol{\sigma}_{\mathbf{q}}$ is two orders of magnitude smaller than the mean $\boldsymbol{\mu}_{\mathbf{q}}$, leading to robust confidence intervals. In particular, the relative space-time error between the ground truth pressure and the estimate $\boldsymbol{\mu}_{\mathbf{q}}$ delivered by the proposed method is 0.688%. Figure 21 reports the evolution of the solution rank and reduced basis dimension. The dimension of the reduced basis tends to increase for all variables, which is symptomatic of convection-dominated flow problems, but this increase is mitigated as the MC sampling starts to converge toward the posterior probability distribution. In particular, the computational speedup factor delivered by the proposed method compared to a sequential MCMC method using a standard finite element solver for solving system (7) is 173. Of the time required for sampling the posterior probability distribution, 95.25% is spent in the low-rank solver, 0.57% in updating the reduced basis and associated precomputed quantities, 4.17% in the solution interpolation, and 0.01% in the other parts of the method.

ν	Re
$[10^{-3}, 10^{-2}]$	$[1, 200]$

Table 4: Parameter domain $\mathcal{D}$ for the cylinder testcase.

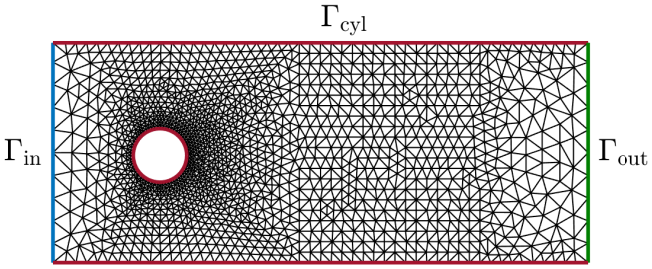
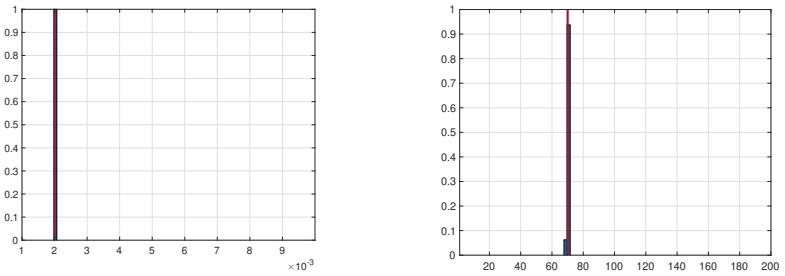
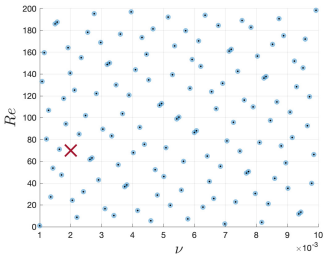


Figure 17: Computational mesh for the cylinder testcase.

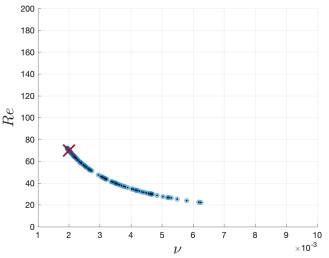


(a) ν (b) Re

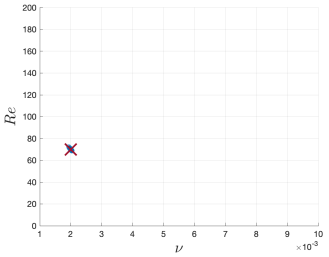
Figure 18: Parameter estimation resulting from the sampling of the posterior probability distribution at final step (i.e., $k = N_t$) for the cylinder testcase. The ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.



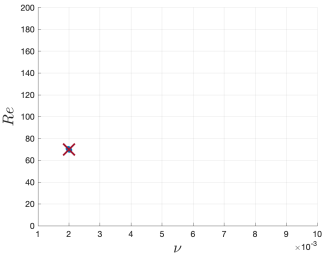
(a) $k = 0$



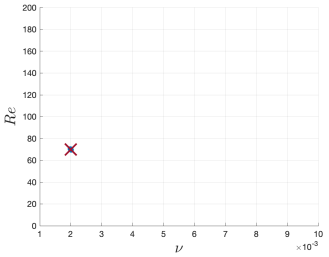
(b) $k = 200$



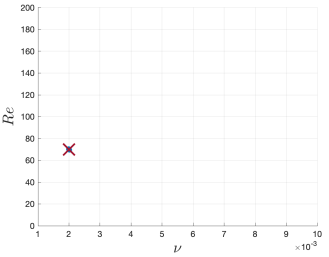
(c) $k = 400$



(d) $k = 600$



(e) $k = 800$

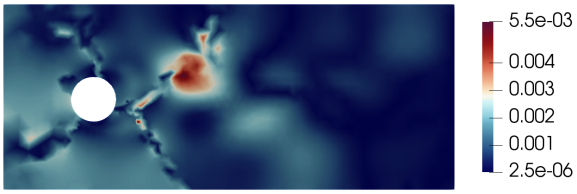


(f) $k = 1000$

Figure 19: Sampling of the posterior probability distribution at different steps k for the cylinder testcase. The ground truth parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.

(a) Ground truth pressure p

(b) Mean $\boldsymbol{\mu}_{\mathbf{q}}$



(c) Standard deviation $\boldsymbol{\sigma}_{\mathbf{q}}$

Figure 20: Estimation of the pressure at final time $t_k = 1$ for the cylinder testcase.

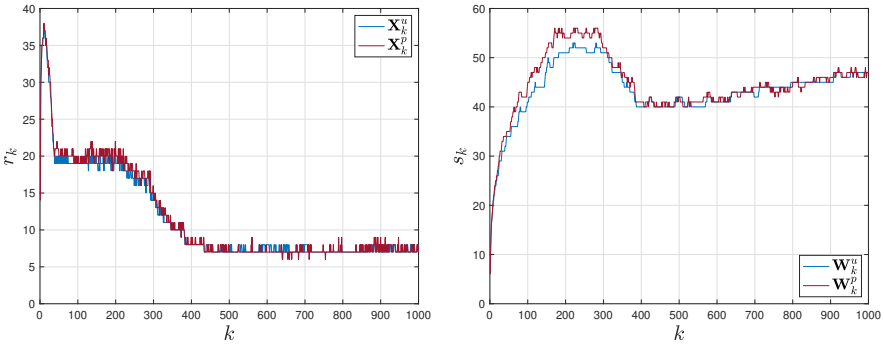


Figure 21: Solution rank and reduced basis dimension at different steps $k \in \{1, \ldots, N_t\}$ for the cylinder testcase.

6 Conclusion

In this work, we have presented a low-rank framework for performing parameter estimation and uncertainty quantification. In this approach, a low-rank solver, based on the fixed-point iteration method and GMRES algorithm, is developed to significantly reduce the storage requirement associated with numerical solutions of parametrized time-dependent problems. The discretisation in the parameters, performed by a collocation method is evolved by exploiting a MCMC method, in order to sample progressively the posterior density distribution. The efficient sampling in the Metropolis-Hastings is enabled by an interpolator. This is constructed by Galerkin projection on a low-dimensional basis built by exploiting the low-rank solution of the system. The interaction between the low-rank solver and the ROM-based interpolator is beneficial for two reasons. First, the interpolator makes it possible to sample the posterior density distribution and hence to adapt the discretisation to be used by the low-rank solver in an inexpensive way. Second, the low-rank solver, as it provides a low-rank solution, makes the ROM basis construction and update more efficient both in terms of memory and number of operations.

The proposed method have been evaluated on four applications. We first have investigated the estimation of input model parameters from noisy partial observations. The results shows the robustness of the method against the existence of local minima. Then, we have investigated the prediction of different quantities of interest. The results demonstrated the accuracy of the proposed method, capable of delivering less than 1% of error, while enabling a computational speedup factor of several orders of magnitude.

A SVD of a second-order tensor

Algorithm 2 SVD of a second-order tensor

Input: $\mathbf{U}_1 \in \mathbb{R}^{N_x \times r}$, $\mathbf{\Sigma}_1 \in \mathbb{R}^{r \times r}$, and $\mathbf{V}_1 \in \mathbb{R}^{N_\theta \times r}$, such that $\mathbf{X} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T$

Output: $\mathbf{U}_2 \in \mathbb{R}^{N_x \times r}$, $\mathbf{\Sigma}_2 \in \mathbb{R}^{r \times r}$, and $\mathbf{V}_2 \in \mathbb{R}^{N_\theta \times r}$, such that $\mathbf{X} = \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^T$ is the SVD of $\mathbf{X}$

1: Compute the QR decompositions of $\mathbf{U}_1 = \mathbf{Q}_u \mathbf{R}_u$ and $\mathbf{V}_1 = \mathbf{Q}_v \mathbf{R}_v$

2: Define $\mathbf{K} = \mathbf{R}_u \mathbf{\Sigma}_1 \mathbf{R}_v^T \in \mathbb{R}^{r \times r}$

3: Compute the SVD of $\mathbf{K} = \mathbf{U}_3 \mathbf{\Sigma}_3 \mathbf{V}_3^T$

4: Set $\mathbf{U}_2 = \mathbf{Q}_u \mathbf{U}_3$, $\mathbf{\Sigma}_2 = \mathbf{\Sigma}_3$ and $\mathbf{V}_2 = \mathbf{Q}_v \mathbf{V}_3$

B Truncated preconditioned GMRES algorithm

Algorithm 3 Truncated preconditioned GMRES algorithm

Input: $\boldsymbol{\mathcal{A}} : \mathbb{R}^{N_x \times N_\theta} \to \mathbb{R}^{N_x \times N_\theta}$, $\overline{\mathbf{X}}_{k-1} \in \mathbb{R}^{N_x \times N_\theta}$, $\overline{\mathbf{B}}_k \in \mathbb{R}^{N_x \times N_\theta}$, $\mathbf{P} \in \mathbb{R}^{N_x \times N_x}$ and $\varepsilon_{\text{gmres}} \in (0, 1]$, $\eta_{\text{trunc}} \in [0, 1]$ (involved in (2) and (5))

Output: $\overline{\mathbf{X}}_k \in \mathbb{R}^{N_x \times N_\theta}$

$\overline{\mathbf{R}} = \mathbf{P}^{-1}\boldsymbol{\mathcal{T}}(\overline{\mathbf{B}}_k - \boldsymbol{\mathcal{A}}(\overline{\mathbf{X}}_{k-1}))$
$\overline{\mathbf{Q}}_1 = \overline{\mathbf{R}}/\|\overline{\mathbf{R}}\|_F$
$\boldsymbol{\beta} = [\|\overline{\mathbf{R}}\|_F, 0, \ldots, 0] \in \mathbb{R}^{N_{\text{gmres}}}$
for $j = 1, \ldots, N_{\text{gmres}}$ **do**
    $\overline{\mathbf{R}} = \mathbf{P}^{-1}\boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{A}}(\overline{\mathbf{Q}}_j))$
    for $i = 1, \ldots, j$ **do**
        $H_{i,j} = \langle \overline{\mathbf{Q}}_i, \overline{\mathbf{R}} \rangle_F$
        $\overline{\mathbf{R}} = \boldsymbol{\mathcal{T}}(\overline{\mathbf{R}} - H_{i,j}\overline{\mathbf{Q}}_i)$
    end
    $H_{j+1,j} = \|\overline{\mathbf{R}}\|_F$
    $\overline{\mathbf{Q}}_{j+1} = \overline{\mathbf{R}}/H_{j+1,j}$
end
Find $\boldsymbol{\omega} \in \mathbb{R}^{N_{\text{gmres}}}$ minimizing $\|\mathbf{H}\boldsymbol{\omega} - \boldsymbol{\beta}\|_2$
$\overline{\mathbf{X}}_k = \boldsymbol{\mathcal{T}}(\overline{\mathbf{X}}_{k-1} + \sum_{j=1}^{N_{\text{gmres}}} \omega_j \overline{\mathbf{Q}}_j)$

In order to limit memory footprint, Algorithm 3 can also be restarted using the final iterate as initial guess if the maximum number of iterations is reached without converging.

C Brand's algorithm

Algorithm 4 Brand's algorithm [64]

Input: $\mathbf{W}_{k-1} \in \mathbb{R}^{N_x \times s_{k-1}}$, $\mathbf{Y}_{k-1} \in \mathbb{R}^{s_{k-1} \times N_\theta(k-1)}$, $\mathbf{U}_k \in \mathbb{R}^{N_x \times r_k}$, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{r_k \times r_k}$, and $\eta_{\text{proj}} \in [0, 1]$

Output: $\mathbf{W}_k \in \mathbb{R}^{N_x \times s_k}$

Define $\mathbf{M} = \mathbf{W}_{k-1} - \mathbf{U}_k\mathbf{U}_k^T\mathbf{W}_{k-1} \in \mathbb{R}^{N_x \times s_{k-1}}$
Compute the QR decomposition of $\mathbf{M} = \mathbf{QR}$
Assemble $\mathbf{K} = \begin{bmatrix} \boldsymbol{\Sigma}_k & \mathbf{U}_k^T\mathbf{W}_{k-1}\mathbf{Y}_{k-1} \\ \mathbf{0} & \mathbf{RY}_{k-1} \end{bmatrix} \in \mathbb{R}^{(r_k+s_{k-1}) \times N_\theta(k-1)}$
Compute the SVD (or eigendecomposition) of $\mathbf{KK}^T = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$
Set $\mathbf{W}_k = [\mathbf{W}_{k-1}\ \mathbf{Q}][\mathbf{v}_1, \ldots, \mathbf{v}_{s_k}]$

Declarations

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Acknowledgement

The first author acknowledges support by Inria through the PEPR "Santé Numérique". The third author acknowledges support from ANR grant ADAPT 18-CE46-0001.

References

[1] Bellman, R., Åström, K.J.: On structural identifiability. Mathematical biosciences **7**(3-4), 329–339 (1970)

[2] Stuart, A.M.: Inverse problems: a bayesian perspective. Acta numerica **19**, 451–559 (2010)

[3] Reich, S., Cotter, C.: Probabilistic Forecasting and Bayesian Data Assimilation. Cambridge University Press, Cambridge (2015)

[4] Dashti, M., Stuart, A.M.: The bayesian approach to inverse problems. In: Handbook of Uncertainty Quantification, pp. 311–428. Springer, Cham (2017)

[5] Hoang, V.H., Schwab, C., Stuart, A.M.: Complexity analysis of accelerated mcmc methods for bayesian inversion. Inverse Problems **29**(8), 085010 (2013)

[6] Marzouk, Y.M., Najm, H.N.: Dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems. Journal of Computational Physics **228**(6), 1862–1902 (2009)

[7] Giraldi, L., Le Maître, O.P., Mandli, K.T., Dawson, C.N., Hoteit, I., Knio, O.M.: Bayesian inference of earthquake parameters from buoy data using a polynomial chaos-based surrogate. Computational Geosciences **21**(4), 683–699 (2017)

[8] Nagel, J.B., Sudret, B.: Spectral likelihood expansions for bayesian inference. Journal of Computational Physics **309**, 267–294 (2016)

[9] Schwab, C., Stuart, A.M.: Sparse deterministic approximation of bayesian inverse problems. Inverse Problems **28**(4), 045003 (2012)

[10] Shao, Q., Younes, A., Fahs, M., Mara, T.A.: Bayesian sparse polynomial chaos expansion for global sensitivity analysis. Computer Methods in Applied Mechanics and Engineering **318**, 474–496 (2017)

[11] Yan, L., Zhou, T.: Adaptive multi-fidelity polynomial chaos approach to bayesian inference in inverse problems. Journal of Computational Physics **381**, 110–128 (2019)

[12] Ma, X., Zabaras, N.: An efficient bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method. Inverse Problems **25**(3), 035013 (2009)

[13] Nguyen, N., Rozza, G., Huynh, D.P., Patera, A.T.: Reduced basis approximation and a posteriori error estimation for parametrized parabolic pdes: Application to real-time bayesian parameter estimation. Large-Scale Inverse Problems and Quantification of Uncertainty, 151–177 (2010)

[14] Boyaval, S.: A fast monte–carlo method with a reduced basis of control variates applied to uncertainty propagation and bayesian estimation. Computer Methods in Applied Mechanics and Engineering **241**, 190–205 (2012)

[15] Cui, T., Marzouk, Y.M., Willcox, K.E.: Data-driven model reduction for the bayesian solution of inverse problems. International Journal for Numerical Methods in Engineering **102**(5), 966–990 (2015)

[16] Chen, P., Schwab, C.: Sparse-grid, reduced-basis bayesian inversion. Computer Methods in Applied Mechanics and Engineering **297**, 84–115 (2015)

[17] Chen, P., Ghattas, O.: Stein variational reduced basis bayesian inversion. SIAM Journal on Scientific Computing **43**(2), 1163–1193 (2021)

[18] Huang, D.Z., Huang, J., Reich, S., Stuart, A.M.: Efficient derivative-free bayesian inference for large-scale inverse problems. arXiv preprint arXiv:2204.04386 (2022)

[19] Cui, T., Dolgov, S., Zahm, O.: Scalable conditional deep inverse Rosenblatt transports using tensor-trains and gradient-based dimension reduction. arXiv (2023)

[20] Del Moral, P.: Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications vol. 88. Springer, New York (2004)

[21] Särkkä, S.: Bayesian Filtering and Smoothing vol. 3. Cambridge university press, New York (2013)

[22] Fox, C., Norton, R.A., Morrison, M.E.K., Molteno, T.C.A.: In: de Gier, J., Praeger, C.E., Tao, T. (eds.) Sequential Bayesian Inference for Dynamical Systems Using the Finite Volume Method, pp. 13–23. Springer, Cham (2019)

[23] Doucet, A., Godsill, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and computing **10**(3), 197–208 (2000)

[24] Chen, Z., *et al.*: Bayesian filtering: From kalman filters to particle filters, and beyond. Statistics **182**(1), 1–69 (2003)

[25] Evensen, G.: The ensemble kalman filter: Theoretical formulation and practical implementation. Ocean dynamics **53**(4), 343–367 (2003)

[26] Ernst, O.G., Sprungk, B., Starkloff, H.-J.: Analysis of the ensemble and polynomial chaos kalman filters in bayesian inverse problems. SIAM/ASA Journal on Uncertainty Quantification **3**(1), 823–851 (2015)

[27] Pagani, S., Manzoni, A., Quarteroni, A.: Efficient state/parameter estimation in nonlinear unsteady pdes by a reduced basis ensemble kalman filter. SIAM/ASA Journal on Uncertainty Quantification **5**(1), 890–921 (2017)

[28] Silva, F.A., Pagliantini, C., Grepl, M., Veroy, K.: A reduced basis ensemble kalman method. arXiv preprint arXiv:2210.02279 (2022)

[29] Bachmayr, M., Cohen, A., Dahmen, W.: Parametric pdes: sparse or low-rank approximations? IMA Journal of Numerical Analysis **38**(4), 1661–1708 (2018)

[30] Nouy, A.: A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Computer Methods in Applied Mechanics and Engineering **199**(23-24), 1603–1626 (2010)

[31] Chinesta, F., Ladevèze, P.: Separated representations and pgd-based model reduction. Fundamentals and Applications, International Centre for Mechanical Siences, Courses and Lectures **554**, 24 (2014)

[32] Khoromskij, B.N., Schwab, C.: Tensor-structured galerkin approximation of parametric and stochastic elliptic pdes. SIAM Journal on Scientific Computing **33**(1), 364–385 (2011)

[33] Zhang, Z., Yang, X., Oseledets, I.V., Karniadakis, G.E., Daniel, L.: Enabling high-dimensional hierarchical uncertainty quantification by anova and tensor-train decomposition. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **34**(1), 63–76 (2014)

[34] Ballani, J., Grasedyck, L.: Hierarchical tensor approximation of output quantities of parameter-dependent pdes. SIAM/ASA Journal on Uncertainty Quantification **3**(1), 852–872 (2015)

[35] Dolgov, S., Scheichl, R.: A hybrid alternating least squares–tt-cross algorithm for parametric pdes. SIAM/ASA Journal on Uncertainty Quantification **7**(1), 260–291 (2019)

[36] Khoromskij, B.N., Oseledets, I.: Quantics-tt collocation approximation of parameter-dependent and stochastic elliptic pdes. Computational methods in applied mathematics **10**(4), 376–394 (2010)

[37] Sapsis, T.P., Lermusiaux, P.F.: Dynamically orthogonal field equations for continuous stochastic dynamical systems. Physica D: Nonlinear Phenomena **238**(23-24), 2347–2360 (2009)

[38] Sapsis, T.P., Lermusiaux, P.F.J.: Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty. Physica D: Nonlinear Phenomena **241**(1), 60–76 (2012)

[39] Musharbash, E., Nobile, F., Zhou, T.: Error analysis of the dynamically orthogonal approximation of time dependent random pdes. SIAM Journal on Scientific Computing **37**(2), 776–810 (2015)

[40] Cheng, M., Hou, T.Y., Zhang, Z.: A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations i: Derivation and algorithms. Journal of Computational Physics **242**, 843–868 (2013)

[41] Choi, M., Sapsis, T.P., Karniadakis, G.E.: On the equivalence of dynamically orthogonal and bi-orthogonal methods: Theory and numerical simulations. Journal of Computational Physics **270**, 1–20 (2014)

[42] Koch, O., Lubich, C.: Dynamical low-rank approximation. SIAM Journal on Matrix Analysis and Applications **29**(2), 434–454 (2007)

[43] Ceruti, G., Kusch, J., Lubich, C.: A rank-adaptive robust integrator for dynamical low-rank approximation. BIT Numerical Mathematics, 1–26 (2022)

[44] Benner, P., Onwunta, A., Stoll, M.: Low-rank solution of unsteady diffusion equations with stochastic coefficients. SIAM/ASA Journal on Uncertainty Quantification **3**(1), 622–649 (2015)

[45] Benner, P., Dolgov, S., Onwunta, A., Stoll, M.: Low-rank solvers for unsteady stokes–brinkman optimal control problem with random data. Computer Methods in Applied Mechanics and Engineering **304**, 26–54 (2016)

[46] Stoll, M., Breiten, T.: A low-rank in time approach to pde-constrained optimization. SIAM Journal on Scientific Computing **37**(1), 1–29 (2015)

[47] Kressner, D., Tobler, C.: Low-rank tensor krylov subspace methods for parametrized linear systems. SIAM Journal on Matrix Analysis and Applications **32**(4), 1288–1316 (2011)

[48] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. The journal of chemical physics **21**(6), 1087–1092 (1953)

[49] Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**(1), 97–109 (1970)

[50] Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**(3), 211–218 (1936)

[51] Saad, Y., Schultz, M.H.: Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on scientific and statistical computing **7**(3), 856–869 (1986)

[52] Dolgov, S.V.: Tt-gmres: solution to a linear system in the structured tensor format. Russian Journal of Numerical Analysis and Mathematical Modelling **28**(2), 149–172 (2013)

[53] Ballani, J., Grasedyck, L.: A projection method to solve linear systems in tensor format. Numerical linear algebra with applications **20**(1), 27–43 (2013)

[54] Weinhandl, R., Benner, P., Richter, T.: Low-rank linear fluid-structure interaction discretizations. Wiley Online Library (2020)

[55] Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique **339**(9), 667–672 (2004)

[56] Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM Journal on Scientific Computing **32**(5), 2737–2764 (2010)

[57] Sirovich, L.: Turbulence and the dynamics of coherent structures. i.

coherent structures. Quarterly of applied mathematics **45**(3), 561–571 (1987)

[58] Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annual review of fluid mechanics **25**(1), 539–575 (1993)

[59] Farhat, C., Chapman, T., Avery, P.: Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. International journal for numerical methods in engineering **102**(5), 1077–1110 (2015)

[60] Badia, S., Nobile, F., Vergara, C.: Fluid-structure partitioned procedures based on Robin transmission conditions. J. Comp. Phys. **227**, 7027–7051 (2008)

[61] Fernández, M.A., Mullaert, J., Vidrascu, M.: Generalized Robin-Neumann explicit coupling schemes for incompressible fluid-structure interaction: stability analysis and numerics. Internat. J. Numer. Methods Engrg. **101**(3), 199–229 (2015)

[62] Burman, E., Fernández, M.A.: Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility. Comput. Methods Appl. Mech. Engrg. **198**(5-8), 766–784 (2009)

[63] Brezzi, F., Pitkäranta, J.: On the stabilization of finite element approximations of the Stokes equations. Notes Numer. Fluid Mech., vol. 10, pp. 11–19. Vieweg (1984)

[64] Brand, M.: Fast low-rank modifications of the thin singular value decomposition. Linear algebra and its applications **415**(1), 20–30 (2006)