HAL open science

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

Sébastien Riffaud, Miguel Angel Fernández, Damiano Lombardi

To cite this version:

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. Journal of Scientific Computing, In press. hal-03908682v1

HAL Id: hal-03908682

https://inria.hal.science/hal-03908682v1

Submitted on 20 Dec 2022 (v1), last revised 4 Feb 2024 (v2)

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

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

Distributed under a Creative Commons Attribution 4.0 International License

A low-rank solver for parameter estimation and uncertainty quantification in linear 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.

Contributing authors: sebastien.riffaud@inria.fr; miguel.fernandez@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 linear systems of Partial Differential Equations. The solution approximation is look for 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 three different applications.

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

Contents

1 Introduction **2**

2 Low-rank tensor solver **4**

2.1 Parametrized discrete system 5
2.2 Preconditioned GMRES algorithm 6
2.3 Low-rank approximation 7

3 Markov chain Monte Carlo method 9
3.1 Metropolis-Hastings algorithm 10
3.2 Interpolator 11

4 Numerical experiments. 13
4.1 Heat conduction in 3D 14
4.2 Pressure-wave propagation in 2D 18
4.3 Flow over an elastic wall in 3D 24

5 Conclusion 28

1 Introduction

This work deals with systems of linear 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. 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 [19, 20]. Particle filters and Sequential Monte Carlo have been investigated in [21, 22] 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 [23] 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 [24] a polynomial chaos formulation is analysed, in [25, 26] 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 linear 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 [27] 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 [28, 29] 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 [30]. In [31] uncertainty quantification has been considered combining ANOVA and the Tensor Train format. Hierarchical tensor approximation have been considered in [32], and in [33] 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 [34]. In the present work, the formulation considered is similar, in spirit, to the one proposed in [35–37]. When discretising the system, we will use a preconditioned Krylov-based method (we refer to [38]) 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 adaptation of the parameter discretisation will be performed by exploiting an efficient interpolator. 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 structure of the work is the following:

2 Low-rank tensor solver

In this section the low-rank solver for parametric PDEs is presented. Let the parameter domain be $\mathcal{D} \subset \mathbb{R}^p$, 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 linear parametric PDE can be written in abstract form as:

$$\partial_t u = \mathcal{A}(\boldsymbol{\theta})u,$$

where the function u is the solution and $\mathcal{A}$ is a linear operator acting on u. 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:

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

   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:

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

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.

2.1 Parametrized discrete system

Let the parameter domain $\mathcal{D}$ be a closed and bounded subset of the Euclidean space $\mathbb{R}^p$. In this work, we consider parametrized time-dependent PDEs that lead at each time-step t_k to a linear discrete system of the form

$$\mathbf{A}(\boldsymbol{\theta})\mathbf{x}(t_k;\ \boldsymbol{\theta}) = \mathbf{b}(t_k;\ \boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta} \in \mathcal{D}$ denotes the input model parameter and $\mathbf{x}(t_k;\ \boldsymbol{\theta}) \in \mathbb{R}^n$ denotes the output discrete solution (collecting the values of the degrees of freedom for the different parameter values). Given the input parameters $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$, the corresponding discrete systems (1) can be assembled into the global linear system

$$\begin{bmatrix} \mathbf{A}(\boldsymbol{\theta}_1) & & \\ & \ddots & \\ & & \mathbf{A}(\boldsymbol{\theta}_m) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t_k;\ \boldsymbol{\theta}_1) \\ \vdots \\ \mathbf{x}(t_k;\ \boldsymbol{\theta}_m) \end{bmatrix} = \begin{bmatrix} \mathbf{b}(t_k;\ \boldsymbol{\theta}_1) \\ \vdots \\ \mathbf{b}(t_k;\ \boldsymbol{\theta}_m) \end{bmatrix},$$

which can also be cast in matrix format as follows

$$\mathcal{A}(\mathbf{X}_k) = \mathbf{B}_k, \tag{2}$$

by defining $\mathcal{A}(\mathbf{X}_k) := [\mathbf{A}(\boldsymbol{\theta}_1)\mathbf{x}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{A}(\boldsymbol{\theta}_m)\mathbf{x}(t_k;\ \boldsymbol{\theta}_m)] \in \mathbb{R}^{n\times m}$, $\mathbf{X}_k := [\mathbf{x}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{x}(t_k;\ \boldsymbol{\theta}_m)] \in \mathbb{R}^{n\times m}$ and $\mathbf{B}_k := [\mathbf{b}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{b}(t_k;\ \boldsymbol{\theta}_m)] \in \mathbb{R}^{n\times m}$.

Note that since the matrix $\mathbf{A}(\boldsymbol{\theta})$ depends only on $\boldsymbol{\theta}$, it can be decomposed as

$$\mathbf{A}(\boldsymbol{\theta}) = \sum_{i=1}^{n_a} a_i(\boldsymbol{\theta})\mathbf{A}_i, \tag{3}$$

where $n_a \ll m$. For example, the heat conduction problem described in Section 4.1 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

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

and $\{\phi_1, \ldots, \phi_n\}$ denotes the finite element basis. The operator $\mathcal{A}$ thus requires to store only the matrices $\{\mathbf{A}_1, \ldots, \mathbf{A}_{n_a}\}$, and not all the matrices $\{\mathbf{A}(\boldsymbol{\theta}_1), \ldots, \mathbf{A}(\boldsymbol{\theta}_m)\}$, which would be prohibitive in terms of storage complexity.

2.2 Preconditioned GMRES algorithm

The parametrized discrete system (2) is solved by the generalized minimal residual (GMRES) algorithm [39]. This iterative method is initialized using the solution at the previous time-step $\mathbf{X}_{k-1}$ and terminates when the relative residual is less than a given tolerance $\varepsilon \in [0, 1]$:

$$\|\mathcal{A}(\mathbf{X}_k) - \mathbf{B}_k\|_F \leqslant \varepsilon \|\mathbf{B}_k\|_F \,, \tag{4}$$

where $\|\cdot\|_F$ stands for the Frobenius norm. The number of iterations $q \in \mathbb{N}$ is therefore defined as the smallest integer such that the stopping criterion (4) is satisfied. In addition, the algorithm is equipped with a left preconditioning matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ to speedup convergence. A common choice for the preconditioner is $\mathbf{P} \approx \mathbf{A}(\boldsymbol{\theta})$. However, since we have the matrices $\{\mathbf{A}(\boldsymbol{\theta}_1), \ldots, \mathbf{A}(\boldsymbol{\theta}_m)\}$, this would result in m 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 $\mathcal{D}$. A similar choice is discussed in [40]. The resulting linear solver is described in Algorithm 1.

Remark 1 In order to limit memory footprint, Algorithm 1 can also be restarted using the final iterate as initial guess if the maximum number of iterations is reached without converging.

Algorithm 1 Preconditioned GMRES algorithm [39]

Require: $\mathcal{A} \in \mathbb{R}^{n\times n\times m}$, $\mathbf{X}_{k-1} \in \mathbb{R}^{n\times m}$, $\mathbf{B}_k \in \mathbb{R}^{n\times m}$, $\mathbf{P} \in \mathbb{R}^{n\times n}$ and $\varepsilon \in [0,1]$
Ensure: $\mathbf{X}_k \in \mathbb{R}^{n\times m}$

```
R = P^{-1} (B_k - A(X_{k-1}))
Q_1 = R / ||R||_F
β = [||R||_F, 0, ..., 0] ∈ R^q
for j = 1, ..., q do
    R = P^{-1} A(Q_j)
    for i = 1, ..., j do
        H_{i,j} = ⟨Q_i, R⟩_F
        R = R - H_{i,j} Q_i
    end for
    H_{j+1,j} = ||R||_F
    Q_{j+1} = R / H_{j+1,j}
end for
Find ω ∈ R^q minimizing ||Hω - β||_2
X_k = X_{k-1} + Σ_{j=1}^{q} ω_j Q_j
```

2.3 Low-rank approximation

In order to reduce the number of degrees of freedom of the large-scale system (1), the solution $\mathbf{X}_k$ is represented as a low-rank tensor

$$\overline{\mathbf{X}}_k = \sum_{i=1}^{r_k} \sigma_i^k \mathbf{u}_i^k (\mathbf{v}_i^k)^T = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T,$$

where $\mathbf{U}_k = [\mathbf{u}_1^k, \ldots, \mathbf{u}_{r_k}^k] \in \mathbb{R}^{n\times r_k}$, $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1^k, \ldots, \sigma_{r_k}^k) \in \mathbb{R}^{r_k\times r_k}$ and $\mathbf{V}_k = [\mathbf{v}_1^k, \ldots, \mathbf{v}_{r_k}^k] \in \mathbb{R}^{m\times r_k}$. In the same way, the other matrices involved in Algorithm 1 are approximated by low-rank tensors in order to limit storage requirement. The algebraic operations used in Algorithm 1 then need to be expressed in tensor format. For instance, the addition of two matrices becomes

$$\overline{\mathbf{X}}_1 + \overline{\mathbf{X}}_2 = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix} \begin{bmatrix}\mathbf{\Sigma}_1 & \\ & \mathbf{\Sigma}_2\end{bmatrix} \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}^T,$$

the multiplication of a matrix by a scalar α becomes

$$\alpha\overline{\mathbf{X}} = \mathbf{U}(\alpha\mathbf{\Sigma})\mathbf{V}^T,$$

the Frobenius inner product takes the form

$$\left\langle \overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2 \right\rangle_F = \sum_{i=1}^{r_1}\sum_{j=1}^{r_2} \sigma_i^1 \sigma_j^2 \left\langle \mathbf{u}_i^1, \mathbf{u}_j^2 \right\rangle_2 \left\langle \mathbf{v}_i^1, \mathbf{v}_j^2 \right\rangle_2,$$

with $\langle \cdot, \cdot \rangle_2$ the standard Euclidean inner-product, and the left-multiplication by a matrix $\mathbf{P}$ becomes

$$\mathbf{P}\overline{\mathbf{X}} = (\mathbf{P}\mathbf{U})\mathbf{\Sigma}\mathbf{V}^T.$$

Notably, the computational complexity of the Frobenius norm scales as $\mathcal{O}(nmr^2)$. It is therefore crucial to ensure low-rank representations in order to reduce the computational complexity of algebraic operations. In particular, the optimal way for truncating a tensor is given by the Eckart-Young theorem [41]. Without loss of generality, we can assume that $\overline{\mathbf{X}}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ 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 tensor is then truncated by keeping only the first r_k columns of $\mathbf{U}_k$ and $\mathbf{V}_k$ according to the Eckart-Young theorem [41]. In practice, the rank $r_k \in \mathbb{N}$ is defined as the smallest integer such that the relative approximation error is less than a given tolerance $\eta \in [0, 1]$:

$$\left\|\mathbf{X}_k - \overline{\mathbf{X}}_k\right\|_F \leqslant \eta \left\|\mathbf{X}_k\right\|_F.$$

Let $\mathcal{T}(\mathbf{X}_k) = \overline{\mathbf{X}}_k$ be the resulting rounding operator, the tensor-extended version of Algorithm 1 is described in Algorithm 3. Notably, the rounding operator is used at lines 1, 5, 8 and 14 to balance the increase in rank occurring when tensors are added. In addition, the stopping criterion (4) is modified to take into account the approximation error:

$$\left\|\mathcal{A}(\overline{\mathbf{X}}_k) - \overline{\mathbf{B}}_k\right\|_F \leqslant \varepsilon\sqrt{1-\eta^2}\left\|\overline{\mathbf{B}}_k\right\|_F.$$

In the following, the rounding accuracy (which must be less than the GMRES target accuracy ε) is taken as $\eta = 10^{-8}$ to be small compared to ε, which will be set to $\varepsilon = 10^{-6}$ to provide sufficient accuracy in the numerical examples of Section 4.

Algorithm 2 Orthonormalization process

Require: $\mathbf{U}_1 \in \mathbb{R}^{n\times r}$, $\mathbf{\Sigma}_1 \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_1 \in \mathbb{R}^{m\times r}$, such that $\mathbf{X} = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T$
Ensure: $\mathbf{U}_2 \in \mathbb{R}^{n\times r}$, $\mathbf{\Sigma}_2 \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_2 \in \mathbb{R}^{m\times r}$, 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 $\mathbf{V}_2 = \mathbf{Q}_v\mathbf{V}_3$

Algorithm 3 Preconditioned TT-GMRES algorithm

Require: $\mathcal{A} \in \mathbb{R}^{n\times n\times m}$, $\overline{\mathbf{X}}_{k-1} \in \mathbb{R}^{n\times m}$, $\overline{\mathbf{B}}_k \in \mathbb{R}^{n\times m}$, $\mathbf{P} \in \mathbb{R}^{n\times n}$, $\varepsilon \in [0,1]$ and $\eta \in [0,1]$
Ensure: $\overline{\mathbf{X}}_k \in \mathbb{R}^{n\times m}$

```
R̄ = P^{-1} T(B̄_k − A(X̄_{k−1}))
Q̄_1 = R̄/‖R̄‖_F
β = [‖R̄‖_F, 0, ..., 0] ∈ R^q
for j = 1, ..., q do
    R̄ = P^{-1} T(A(Q̄_j))
    for i = 1, ..., j do
        H_{i,j} = ⟨Q̄_i, R̄⟩_F
        R̄ = T(R̄ − H_{i,j} Q̄_i)
    end for
    H_{j+1,j} = ‖R̄‖_F
    Q̄_{j+1} = R̄/H_{j+1,j}
end for
Find ω ∈ R^q minimizing ‖Hω − β‖_2
X̄_k = T(X̄_{k−1} + Σ_{j=1}^{q} ω_j Q̄_j)
```

3 Markov chain Monte Carlo method

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):

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

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.

3.1 Metropolis-Hastings algorithm

Given a temporal sequence of noisy partial observations $\mathbf{z}_{\text{obs}}(t)$ generated from the input model parameter $\boldsymbol{\theta}_{\text{obs}}$:

$$\mathbf{z}_{\text{obs}}(t) = \boldsymbol{f}\big(\mathbf{x}(t;\ \boldsymbol{\theta}_{\text{obs}}), t;\ \boldsymbol{\theta}_{\text{obs}}\big),$$

the Markov chain Monte Carlo (MCMC) method is employed to estimate the input parameter that best fits the data. In particular, the sampling $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$ of the posterior probability distribution is updated at each time-step t_k by the Metropolis-Hastings algorithm [42, 43]:

1. Uniformly sample the parameter domain $\mathcal{D}$ to initialize $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$.
2. For each time-step t_k:
   (a) Generate candidates
   $\widehat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i + \gamma \Delta\boldsymbol{\theta}_i$ where $\Delta\boldsymbol{\theta}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ for $i \in \{1, \ldots, m\}$.
   (b) Calculate the acceptance ratio
   $$\tau_i = \frac{\exp\left(-\sum_{l=1}^{k} \left\| \mathbf{z}_{\text{obs}}(t_l) - \boldsymbol{f}\big(\mathbf{x}(t_l;\ \widehat{\boldsymbol{\theta}}_i), t_l;\ \widehat{\boldsymbol{\theta}}_i\big) \right\|_2^2 \right)}{\exp\left(-\sum_{l=1}^{k} \left\| \mathbf{z}_{\text{obs}}(t_l) - \boldsymbol{f}\big(\mathbf{x}(t_l;\ \boldsymbol{\theta}_i), t_l;\ \boldsymbol{\theta}_i\big) \right\|_2^2 \right)}.$$
   (c) Accept candidates: if $\upsilon_i \leqslant \tau_i$ with $\upsilon_i \sim \mathcal{U}(0, 1)$, then set $\boldsymbol{\theta}_i = \widehat{\boldsymbol{\theta}}_i$.

Here, the step size is taken as $\gamma = 10^{-2}$ and the correlation matrix $\mathbf{C} \in \mathbb{R}^{p \times p}$ 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 $\boldsymbol{\theta} \in \mathcal{D}$).

Notably, the solutions $\{\mathbf{x}(t_k;\ \boldsymbol{\theta}_1), \ldots, \mathbf{x}(t_k;\ \boldsymbol{\theta}_m)\}$ are evaluated using the low-rank tensor solver presented in the previous section, while the solutions $\{\mathbf{x}(t_l;\ \boldsymbol{\theta}_1), \ldots, \mathbf{x}(t_l;\ \boldsymbol{\theta}_m)\}_{l=1}^{k-1}$ and $\{\mathbf{x}(t_l;\ \widehat{\boldsymbol{\theta}}_1), \ldots, \mathbf{x}(t_l;\ \widehat{\boldsymbol{\theta}}_m)\}_{l=1}^{k}$ are provided by the interpolator described in the next section. In addition, the acceptance-rejection cycle (i.e. steps 2a-2b-2c) can be repeated several times at each time-step to ensure $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$ converges toward the posterior probability distribution. Once this one is known, one can estimate the mean solution

$$\boldsymbol{\mu}_e(t_k) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}(t_k;\ \boldsymbol{\theta}_i)$$

and the variance

$$\boldsymbol{\sigma}_e^2(t_k) = \frac{1}{m-1} \sum_{i=1}^{m} \big(\mathbf{x}(t_k;\ \boldsymbol{\theta}_i) - \boldsymbol{\mu}_e(t_k)\big)^2.$$

The mean solution and variance are obtained by means of a Monte Carlo estimate.

3.2 Interpolator

In order to evaluate $\mathbf{x}(t_{k+1};\ \boldsymbol{\theta})$ by the low-rank tensor solver, we first have to compute $\mathbf{x}(t_k;\ \boldsymbol{\theta})$, since $\mathbf{b}(t_{k+1};\ \boldsymbol{\theta})$ depends on $\mathbf{x}(t_k;\ \boldsymbol{\theta})$. To this end, the solution is approximated for $l \in \{1, \dots, k\}$ by

$$\widetilde{\mathbf{x}}(t_l;\ \boldsymbol{\theta}) = \sum_{i=1}^{s_k} \mathbf{w}_i^k y_i(t_l;\ \boldsymbol{\theta}) = \mathbf{W}_k \mathbf{y}(t_l;\ \boldsymbol{\theta}),$$

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

Proper orthogonal decomposition

At the beginning of step t_k in the Metropolis-Hasting algorithm, the snapshot database is defined as

$$\bigcup_{l=1}^{k} \left\{\mathbf{x}(t_l;\ \boldsymbol{\theta}_1), \dots, \mathbf{x}(t_l;\ \boldsymbol{\theta}_m)\right\},$$

where the solutions $\{\mathbf{x}(t_k;\ \ \boldsymbol{\theta}_1), \dots, \mathbf{x}(t_k;\ \ \boldsymbol{\theta}_m)\}$ are evaluated using the low-rank tensor solver and the solutions $\{\mathbf{x}(t_l;\ \boldsymbol{\theta}_1), \dots, \mathbf{x}(t_l;\ \boldsymbol{\theta}_m)\}_{l=1}^{k-1}$ are approximated by the interpolator. The corresponding snapshot matrix is given by

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

where $\mathbf{Y}_{k-1} = [\mathbf{y}(t_1;\ \boldsymbol{\theta}_1), \dots, \mathbf{y}(t_1;\ \boldsymbol{\theta}_m), \dots, \mathbf{y}(t_{k-1};\ \boldsymbol{\theta}_1), \dots, \mathbf{y}(t_{k-1};\ \boldsymbol{\theta}_m)]$. The reduced basis $\mathbf{W}_k$ is then build by Proper Orthogonal Decomposition (POD) [44, 45] 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 $\delta \in [0, 1]$:

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

In the following, the projection accuracy is chosen as $\delta = \varepsilon = 10^{-6}$ (which corresponds to 99.9999% of the singular value energy) to be of the same order of magnitude as the GMRES 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 4.2, 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

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

Algorithm 4 Brand's algorithm [46]

Require: $\mathbf{W}_{k-1} \in \mathbb{R}^{n \times s_{k-1}}$, $\mathbf{Y}_{k-1} \in \mathbb{R}^{s_{k-1} \times m(k-1)}$, $\mathbf{U}_k \in \mathbb{R}^{n \times r_k}$, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{r_k \times r_k}$ and $\delta \in [0, 1]$

Ensure: $\mathbf{W}_k \in \mathbb{R}^{n \times s_k}$

1: Define $\mathbf{M} = \mathbf{W}_{k-1} - \mathbf{U}_k \mathbf{U}_k^T \mathbf{W}_{k-1} \in \mathbb{R}^{n \times s_{k-1}}$

2: Compute the QR decomposition of $\mathbf{M} = \mathbf{QR}$

3: Assemble $\mathbf{K} = \begin{bmatrix} \boldsymbol{\Sigma}_k & \mathbf{U}_k^T \mathbf{W}_{k-1} \mathbf{Y}_{k-1} \\ \mathbf{0} & \mathbf{R} \mathbf{Y}_{k-1} \end{bmatrix} \in \mathbb{R}^{(r_k + s_{k-1}) \times m(k-1)}$

4: Compute the SVD (or eigendecomposition) of $\mathbf{K}\mathbf{K}^T = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$

5: Set $\mathbf{W}_k = [\mathbf{W}_{k-1} \ \mathbf{Q}][\mathbf{v}_1, \ldots, \mathbf{v}_{s_k}]$

Galerkin method

The reduced coordinates $\mathbf{y}(t_l;\ \boldsymbol{\theta})$ are finally determined at a low cost for $l \in \{1, \ldots, k\}$ by the Galerkin method:

$$\widetilde{\mathbf{A}}(\boldsymbol{\theta})\mathbf{y}(t_l;\ \boldsymbol{\theta}) = \widetilde{\mathbf{b}}(t_l;\ \boldsymbol{\theta}), \tag{5}$$

where $\widetilde{\mathbf{A}}(\boldsymbol{\theta}) = \mathbf{W}_k^T \mathbf{A}(\boldsymbol{\theta}) \mathbf{W}_k$ and $\widetilde{\mathbf{b}}(t_l;\ \boldsymbol{\theta}) = \mathbf{W}_k^T \mathbf{b}(t_l;\ \boldsymbol{\theta})$. As this system (5) scales with the dimension of the reduced basis $s_k \ll n$, the interpolator typically enables a speedup factor of several orders of magnitude compared to the resolution of system (1). It is also important to notice that, according to (3), the matrix $\widetilde{\mathbf{A}}(\boldsymbol{\theta})$ can be efficiently computed since

$$\widetilde{\mathbf{A}}(\boldsymbol{\theta}) = \sum_{i=1}^{n_a} a_i(\boldsymbol{\theta}) \widetilde{\mathbf{A}}_i,$$

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

Notably, this interpolator can be seen as a reduced order model (ROM) in which the reduced basis is updated on the fly during the online prediction stage. There is therefore no offline training stage. In particular, this enables the ROM to assimilate the new solution snapshots provided by the low-rank tensor solver in order to adapt at each step t_k to the new sampling $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$ of

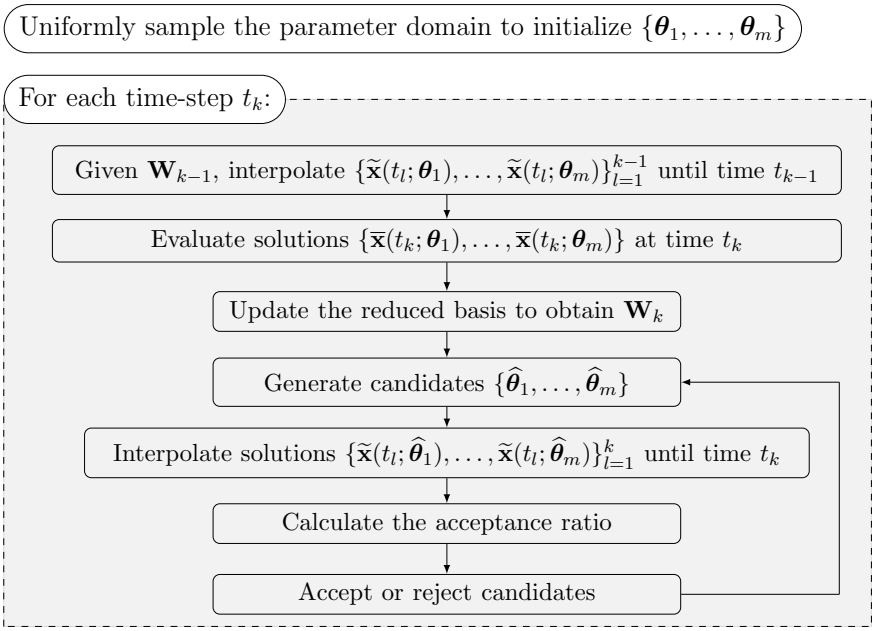


Figure 1: Workflow of the proposed method

the parameter domain. Additionally, this ROM inherits the stability property of the discretization methods used in the following to obtain the linear system (1). The workflow of the resulting method is presented in Figure 1.

4 Numerical experiments.

In this section, the performance of the proposed method is assessed for three applications. In each case, we will perform parameter estimation and uncertainty quantification from partial noisy observations. The accuracy of the estimate $\boldsymbol{\mu}_{\text{qoi}}$ with respect to a selection of ground truth QoIs $\mathbf{z}_{\text{qoi}}$ is evaluated using the relative space-time error:

$$\text{Error} = \sqrt{\frac{\displaystyle\int_0^{t_{max}} \|\mathbf{z}_{\text{qoi}} - \boldsymbol{\mu}_{\text{qoi}}\|_2^2 \, \mathrm{d}t}{\displaystyle\int_0^{t_{max}} \|\mathbf{z}_{\text{qoi}}\|_2^2 \, \mathrm{d}t}}.$$

Furthermore, the computational speedup of the ROM with respect to a standard finite element solver is evaluated in each case in order to quantify the reduction in computational cost provided by this low-rank framework.

4.1 Heat conduction in 3D

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 is governed by the heat equation:

$$\begin{cases} \partial_t T - \kappa \Delta T = 0 & \text{in} \quad \Omega \times \mathbb{R}_+, \\ T = 0 & \text{in} \quad \Omega \times \{0\}, \\ \kappa \dfrac{\partial T}{\partial n} + \nu (T - g) = 0 & \text{on} \quad \Gamma_{\text{int}} \times \mathbb{R}_+, \\ \dfrac{\partial T}{\partial n} = 0 & \text{on} \quad \Gamma_{\text{ext}} \times \mathbb{R}_+, \end{cases} \tag{6}$$

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

$$g(\boldsymbol{x};\, \boldsymbol{\theta}) = c_1 + c_2 \cos(\varphi) + c_3 \sin(\varphi) + c_4 \cos(2\varphi) + c_5 \sin(2\varphi)$$

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 $\Delta t = 10^{-1}$. The heat equation (6) 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 (1) having $n = 35,328$ degrees of freedom (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.

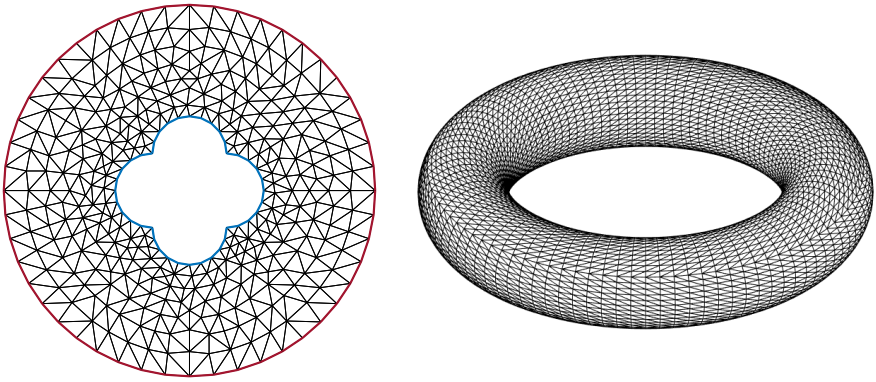


Figure 2: Left: cross section of the domain Ω with boundaries Γ_{int} and Γ_{ext} plotted in blue and red, respectively. Right: computational mesh for the heat conduction problem.

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 for the heat conduction problem.

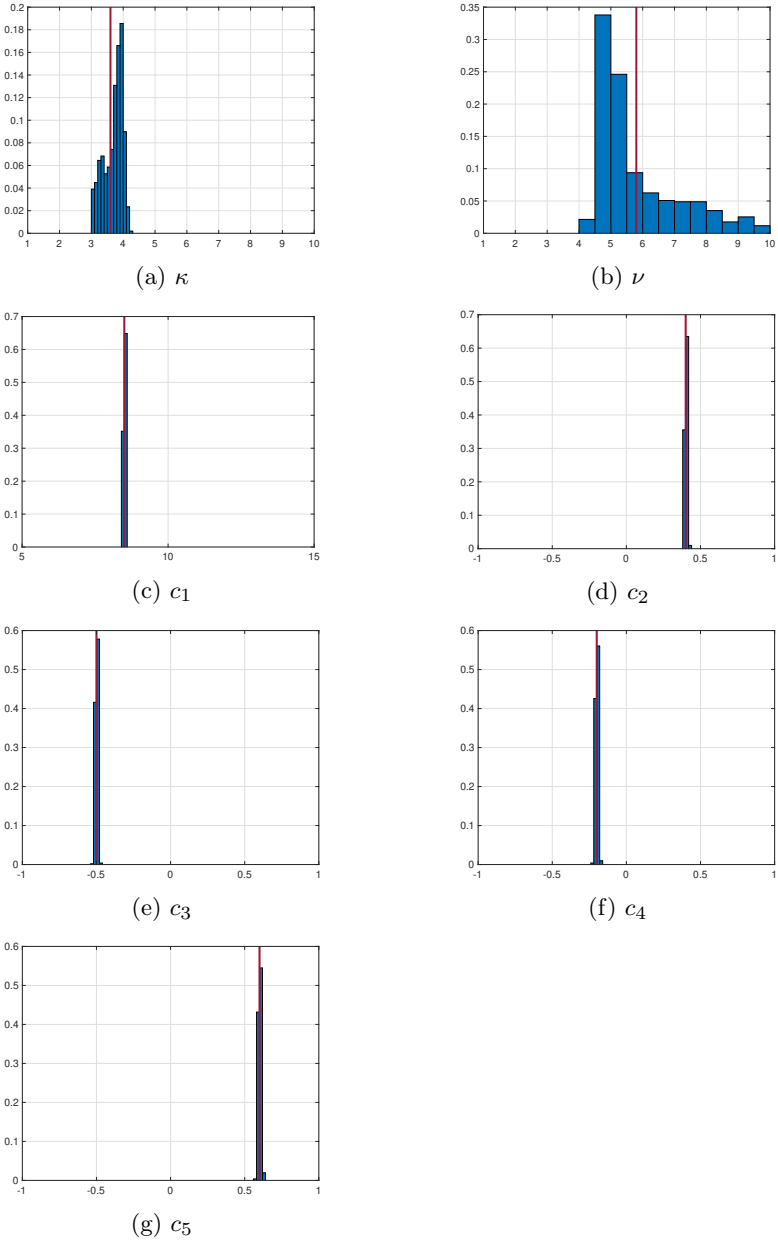
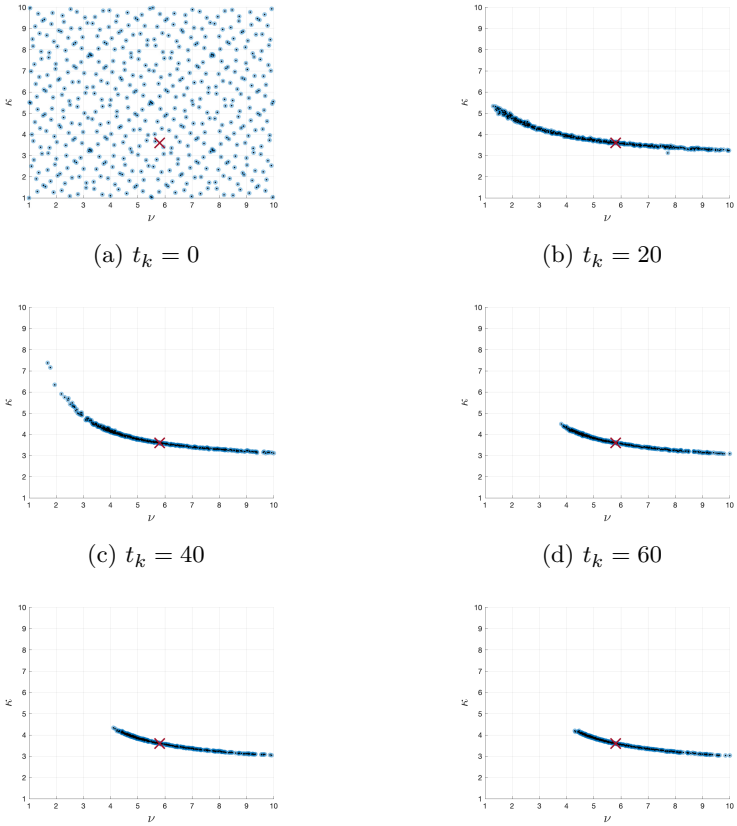


Figure 3: Parameter estimation resulting from the sampling of the posterior probability distribution for the heat conduction problem. The input parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red

This first 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}. These observations contain 5% of noise and were generated from the input 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 $m = 512$ 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 $\boldsymbol{\sigma}_{\text{qoi}}$ is two orders of magnitude smaller than the mean $\boldsymbol{\mu}_{\text{qoi}}$, leading to robust confidence intervals. In particular, the relative space-time error between the ground truth temperature and the estimate $\boldsymbol{\mu}_{\text{qoi}}$ delivered by the proposed method is 0.533%. Figure 6 reports the time histories 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 tensor 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 ROM 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, with a maximum reduced basis dimension of $s_k = 69$, the computational speedup factor delivered by the ROM for solving system (5) compared to a standard finite element solver for the resolution of system (1) is 1,249. Of the time required for sampling the posterior probability distribution, 82.01% is spent in the low-rank tensor solver, 0.05% is spent in updating the reduced basis, 16.75% is spent in the solution interpolation and 1.19% is spent in the other parts of the method.

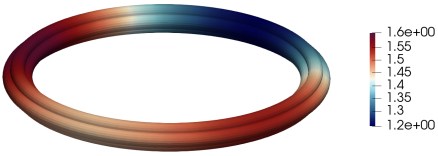


(a) $t_k = 0$ (b) $t_k = 20$

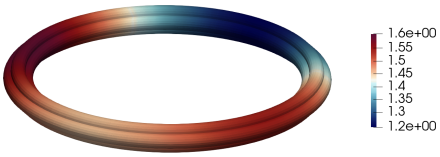
(c) $t_k = 40$ (d) $t_k = 60$

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

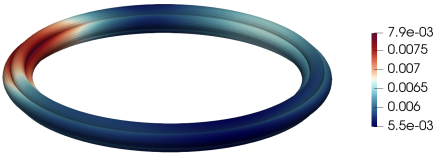
Figure 4: Sampling of the posterior probability distribution at different steps t_k for the heat conduction problem. The input parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red



(a) Ground truth temperature T on Γ_{int} at $t = 10$.



(b) $\boldsymbol{\mu}_{\text{qoi}}$ at $t = 10$.



(c) $\boldsymbol{\sigma}_{\text{qoi}}$ at $t = 10$.

Figure 5: Temperature estimate of the interior surface for the heat conduction problem.

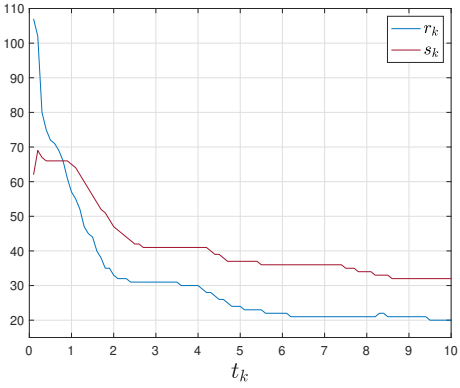


Figure 6: Solution rank and reduced basis dimension over time for the heat conduction problem.

4.2 Pressure-wave propagation in 2D

We consider the well-known fluid-structure interaction benchmark of the propagation of a two-dimensional pressure-wave (see, e.g., [47, 48]). 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).

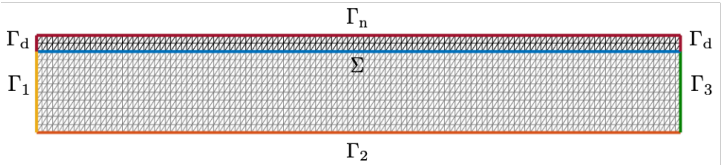


Figure 7: Computational mesh for the 2D pressure-wave propagation problem: the fluid domain Ω_f is plotted in gray, the solid domain Ω_s in black and the interface Σ in blue.

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

$$\begin{cases} \rho_\mathrm{f}\partial_t \boldsymbol{u} - \operatorname{div} \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p) = 0 & \text{in} \quad \Omega_\mathrm{f} \times \mathbb{R}_+, \\ \operatorname{div} \boldsymbol{u} = 0 & \text{in} \quad \Omega_\mathrm{f} \times \mathbb{R}_+, \end{cases} \tag{7a}$$

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

$$\begin{cases} \rho_\mathrm{s}\partial_t \dot{\boldsymbol{d}} - \operatorname{div} \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d}) + \beta \boldsymbol{d} = 0 & \text{in} \quad \Omega_s \times \mathbb{R}_+, \\ \dot{\boldsymbol{d}} = \partial_t \boldsymbol{d} & \text{in} \quad \Omega_s \times \mathbb{R}_+. \end{cases} \tag{7b}$$

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

$$\boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p) := 2\mu_\mathrm{f}\boldsymbol{\epsilon}(\boldsymbol{u}) - p\mathbf{I}, \quad \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d}) := 2\lambda_1\boldsymbol{\epsilon}(\boldsymbol{d}) + \lambda_2(\operatorname{div} \boldsymbol{d})\mathbf{I}.$$

Here, μ_f stands for the fluid dynamics viscosity, λ_1 and λ_2 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 (7a) and (7b) are given by

$$\begin{cases} \boldsymbol{u} = \dot{\boldsymbol{d}} & \text{on} \quad \Sigma \times \mathbb{R}_+, \\ \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p)\boldsymbol{n} = \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d})\boldsymbol{n} & \text{on} \quad \Sigma \times \mathbb{R}_+, \end{cases} \tag{7c}$$

ρ_{f}	μ_{f}	ρ_{s}	λ_1	λ_2
$[0.9, 1.1]$	$[0.03, 0.04]$	$[1, 1.2]$	$[10^6, 10^7]$	$[10^6, 10^7]$

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

where $\boldsymbol{n}$ denotes the outward unit normal to $\partial\Omega_{\mathrm{f}}$. The coupled problem (7) is completed with zero initial conditions on $\boldsymbol{u}$, $\boldsymbol{d}$ and $\dot{\boldsymbol{d}}$. A sinusoidal pressure profile $p_{\mathrm{in}}(t) = p_{\mathrm{max}}(1 - \cos(2t\pi/T^*))/2$, with maximum pressure $p_{\mathrm{max}} = 2 \times 10^4$, 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}.

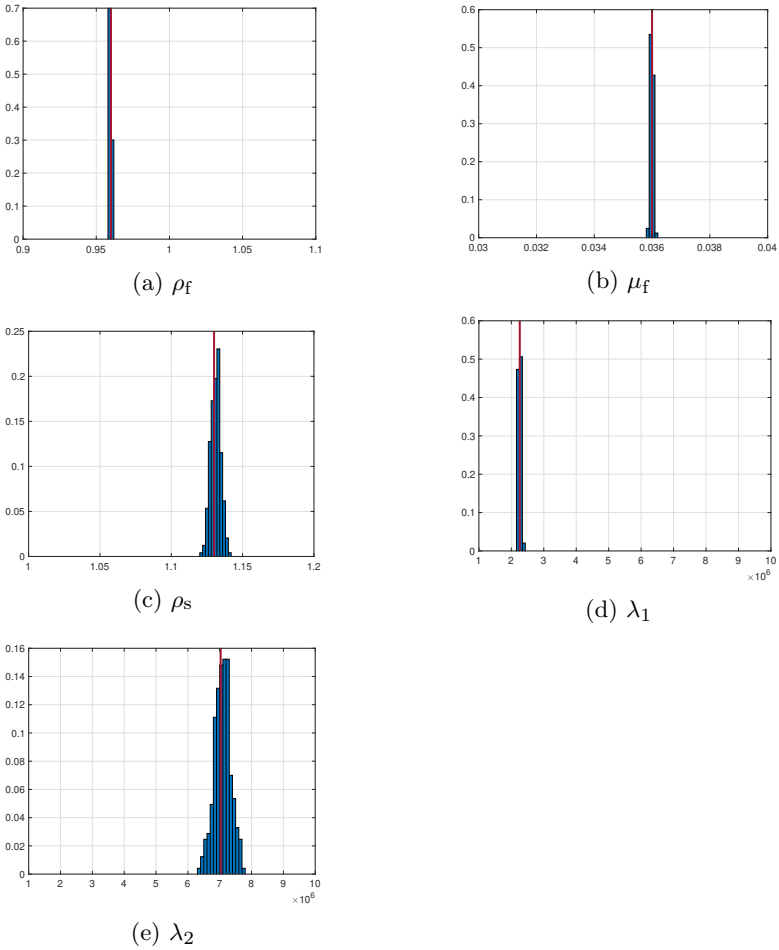


Figure 8: Parameter estimation resulting from the sampling of the posterior probability distribution for the 2D pressure-wave propagation problem. The input parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.

For the time discretization of (7) we consider a fully implicit method (strongly coupled scheme), using a backward Euler scheme in the fluid (7a) and a neutrally stable mid-point scheme for the solid (7b). For the discretization in space we consider continuous $\mathbb{P}_1$ finite elements, with a Nitsche based treatment of the interface conditions (see [49, 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 [50]), 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 length is $\Delta t = 10^{-4}$. The Nitsche penalty parameter is set to $\gamma = 100$. The resulting linear discrete system (1) has $n = 5,445$ DOFs (that is, $2 \cdot 1,331$ for $\boldsymbol{u}$ + $1,331$ for p + $2 \cdot 363$ for $\boldsymbol{d}$ + $2 \cdot 363$ for $\dot{\boldsymbol{d}}$). The input parameters are $\boldsymbol{\theta} = [\rho_{\mathrm{f}}, \mu_{\mathrm{f}}, \rho_s, \lambda_1, \lambda_2]$, and we consider the parameter domain given in Table 2. The solid parameter β is set to $4 \cdot 10^6$.

This second application focuses on the estimation of the fluid pressure p from measurements of the fluid velocity $\boldsymbol{u}$ in the whole computation domain Ω_{f}. This is type of data typically available in 4D-flow MRI measurements. These observations were generated from the input 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 $m = 243$ queried input parameters. The parameters $\{\rho_{\mathrm{f}}, \mu_{\mathrm{f}}, \lambda_1\}$ are accurately estimated, while the values of $\{\rho_s, \lambda_2\}$ are more uncertain since the observations come only from the fluid and, as a consequence, do not provide any direct information about the solid. In particular, the values of λ_2 looks correlated with λ_1 through the affine-like relations that can be identified from the sampling of the posterior probability distribution displayed in Figure 9. Figure 10 presents the pressure estimate at final time resulting from this sampling. The standard deviation $\boldsymbol{\sigma}_{\mathrm{qoi}}$ is three orders of magnitude smaller than the mean $\boldsymbol{\mu}_{\mathrm{qoi}}$, leading to robust confidence intervals. Notably, the relative space-time error between the ground truth pressure and the estimate $\boldsymbol{\mu}_{\mathrm{qoi}}$ delivered by the proposed method is 0.008%. Figure 11 reports the time histories of the solution rank and reduced basis dimension. The former tends to increase for all variables, which is symptomatic of convection-dominated flow problems, while the latter increases at the beginning and then decreases as the sampling of the posterior probability distribution starts to converge. In particular, with a maximum reduced basis dimension of $s_k = 191$, the computational speedup factor delivered by the ROM for solving system (5) compared to a standard finite element solver for the resolution of system (1) is 106.891. Of the time required for sampling the posterior probability distribution, 93.80% is spent in the low-rank tensor solver, 0.15% is spent in updating the reduced basis, 5.73% is spent in the solution interpolation and 0.32% is spent in the other parts of the method.

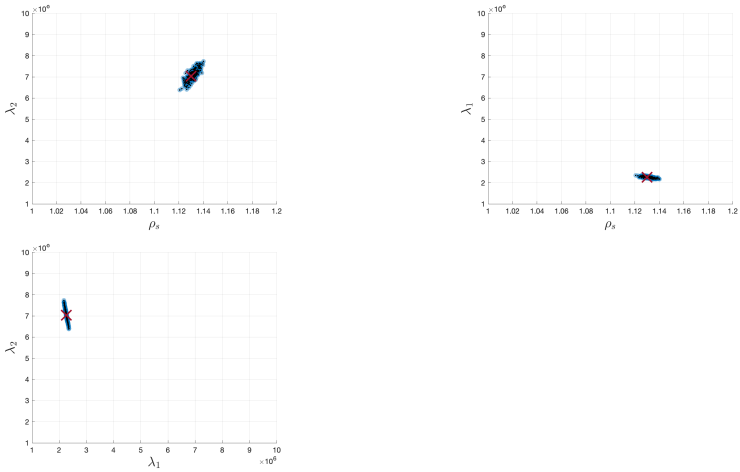
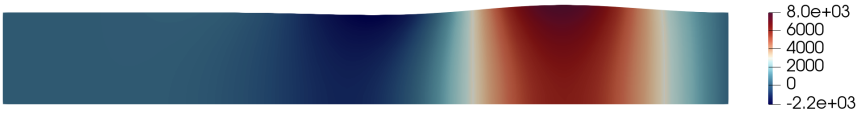
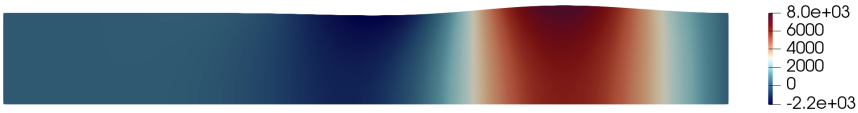


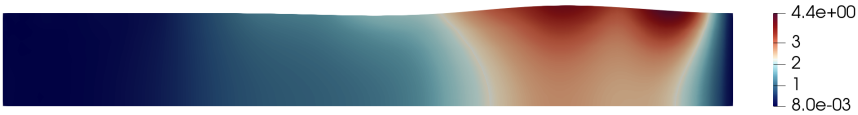
Figure 9: Sampling of the posterior probability distribution for the 2D pressure-wave propagation problem. The input parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red.



(a) Ground truth pressure p at $t = 0.015$.



(b) $\boldsymbol{\mu}_{\text{qoi}}(0.015)$.



(c) $\boldsymbol{\sigma}_{\text{qoi}}(0.015)$.

Figure 10: Estimation of the pressure for the 2D pressure-wave propagation problem.

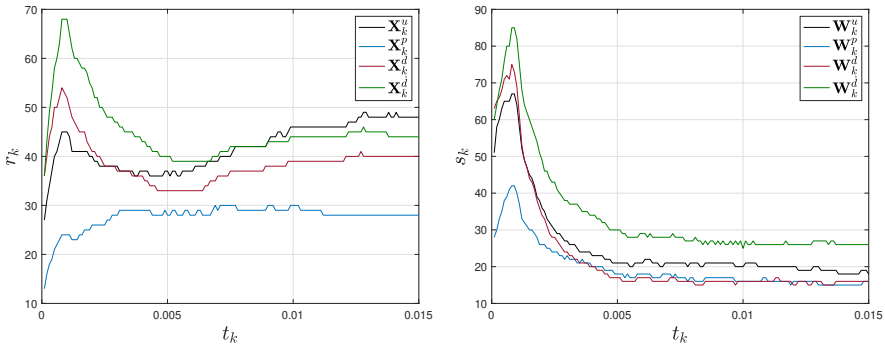


Figure 11: Solution rank and reduced basis dimension over time for the 2D pressure-wave propagation problem.

4.3 Flow over an elastic wall in 3D

We consider the three-dimensional version of the FSI problem (7), 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 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 (7) is discretized using the same numerical method that in Section 4.2. 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 $[0, 0.1]$, and the time-step size is $\Delta t = 10^{-3}$. The resulting linear discrete system (1) having $n = 22,264$ DOFs. The input parameters are $\boldsymbol{\theta} = [\rho_{\mathrm{f}}, \mu_{\mathrm{f}}, \rho_{\mathrm{s}}, \lambda_1, \lambda_2]$, and we consider the parameter domain given in Table 3.

ρ_f	μ_f	ρ_s	λ_1	λ_2
$[0.9, 1.1]$	$[0.03, 0.04]$	$[1, 1.2]$	$[10^5, 10^6]$	$[10^5, 10^6]$

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

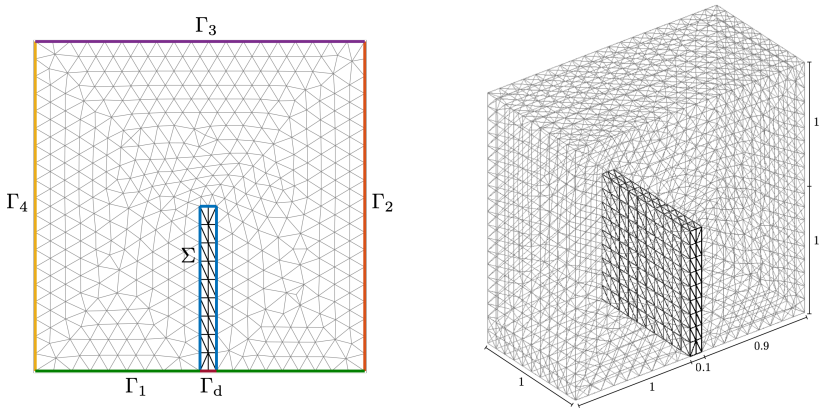
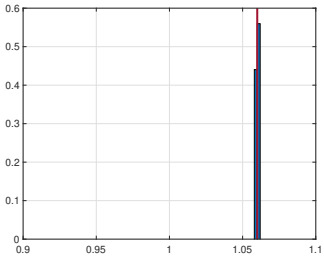
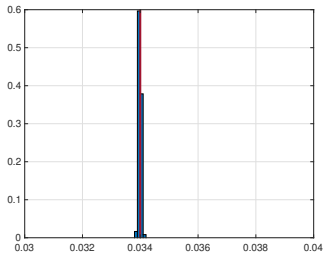


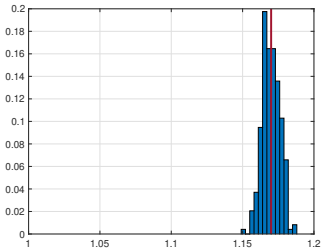
Figure 12: Left: cross section of the computational mesh with the fluid domain Ω_f plotted in gray, the solid domain Ω_s in black and the interface Σ in blue. Right: computational mesh for the 3D FSI example.



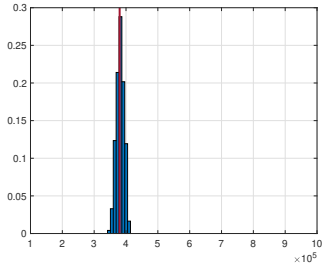
(a) ρ_{f}



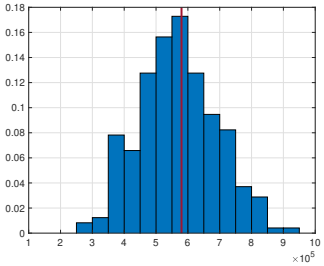
(b) μ_{f}



(c) ρ_{s}



(d) λ_1



(e) λ_2

Figure 13: Parameter estimation resulting from the sampling of the posterior probability distribution for the 3D FSI example. The input parameter $\boldsymbol{\theta}_{\mathrm{obs}}$ is plotted in red.

As previously, we focus on the estimation of the fluid pressure p from measurements of the fluid velocity $\boldsymbol{u}$. These observations were generated from the input 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 $m = 243$ queried input parameters. The parameters $\{\rho_{\mathrm{f}}, \mu_{\mathrm{f}}\}$ are accurately estimated, while the values of $\{\rho_{\mathrm{s}}, \lambda_1, \lambda_2\}$ are more uncertain. In particular, λ_1 and λ_2 seems to be correlated through the affine-like relations that can be identified from the sampling of the posterior probability distribution displayed in Figure 14. Figure 15 presents the pressure estimate at final time resulting from this sampling. The standard deviation $\boldsymbol{\sigma}_{\mathrm{qoi}}$ is three orders

of magnitude smaller than the mean $\boldsymbol{\mu}_{\text{qoi}}$, leading to robust confidence intervals. Notably, the relative space-time error between the ground truth pressure and the estimate $\boldsymbol{\mu}_{\text{qoi}}$ delivered by the proposed method is 0.003%. Figure 16 reports the time histories of the rank dimension and reduced basis dimension. The latter tends to increase for all variables, which is symptomatic of convection-dominated flow problems. In particular, with a maximum reduced basis dimension of $s_k = 144$, the computational speedup factor delivered by the ROM for solving system (5) compared to a standard finite element solver for the resolution of system (1) is 2,968. Of the time required for sampling the posterior probability distribution, 99.41% is spent in the low-rank tensor solver, 0.25% is spent in updating the reduced basis 0.32% is spent in the solution interpolation and 0.02% is spent in the other parts of the method.

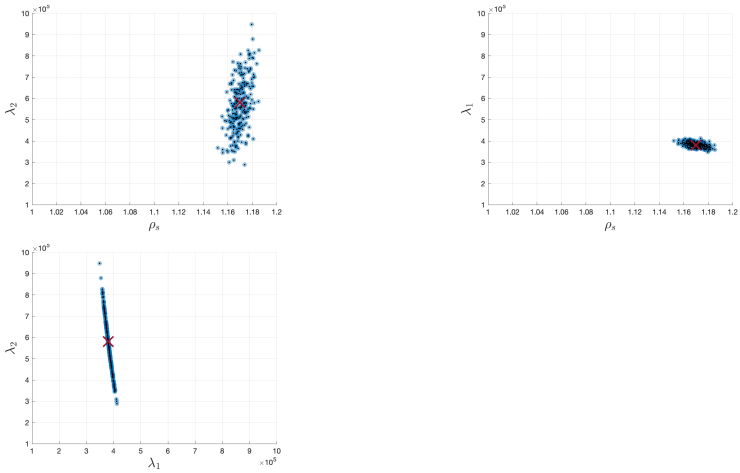
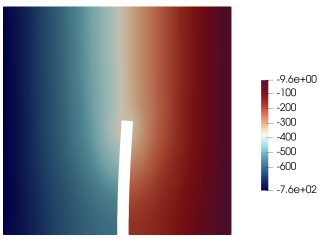
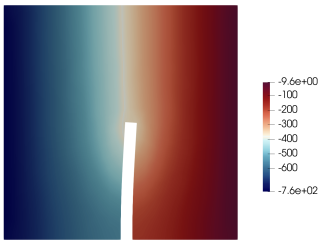


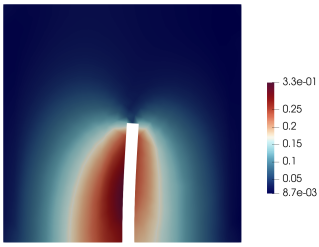
Figure 14: Sampling of the posterior probability distribution for the 3D pressure-wave propagation problem. The input parameter $\boldsymbol{\theta}_{\text{obs}}$ is plotted in red



(a) Ground truth presure p at $t = 0.1$.



(b) $\boldsymbol{\mu}_{\text{qoi}}$ at $t = 0.1$.



(c) $\boldsymbol{\sigma}_{\text{qoi}}$ at $t = 0.1$.

Figure 15: Cross section of the pressure estimate for the 3D FSI example.

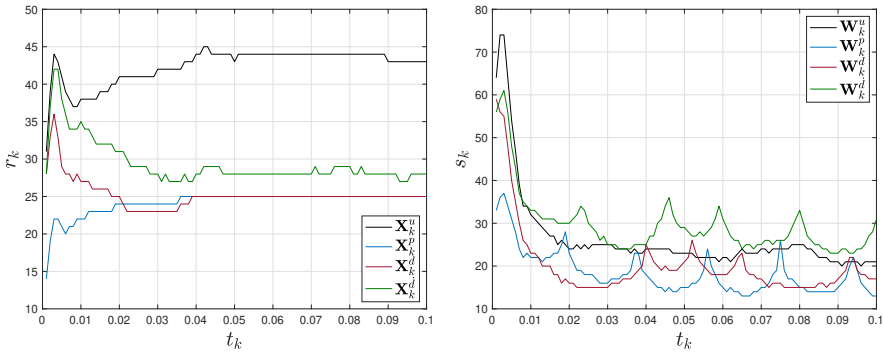


Figure 16: Solution rank and reduced basis dimension over time for the 3D FSI problem.

5 Conclusion

In this work, we have presented a low-rank tensor framework for performing parameter estimation and uncertainty quantification. In this approach, a low-rank solver based on the GMRES algorithm is developed to significantly reduce the storage requirement associated with numerical solutions of parametrized

time-dependent linear 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 three 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.

In future perspective, this low-rank tensor framework will be extended to parametrized time-dependent quadratic problems, which typically arise when discretizing the incompressible Navier-Stokes equations.

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

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

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

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

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

[24] 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)

[25] 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)

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

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

[28] 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)

[29] 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)

[30] 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)

[31] 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)

[32] 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)

[33] 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)

[34] 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)

[35] 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)

[36] 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)

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

[38] 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)

[39] 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)

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

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

[42] 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)

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

[44] Sirovich, L.: Turbulence and the dynamics of coherent structures. i. coherent structures. Quarterly of applied mathematics **45**(3), 561–571 (1987)

[45] 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)

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

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

[48] 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)

[49] 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)

[50] 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)