



MOX-Report No. 105/2023

**Uncertainty quantification for nonlinear solid mechanics using
reduced order models with Gaussian process regression**

Cicci, L.; Fresca, S.; Guo, M.; Manzoni, A.; Zunino, P.

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

mox-dmat@polimi.it

<https://mox.polimi.it>

Uncertainty quantification for nonlinear solid mechanics using reduced order models with Gaussian process regression

Ludovica Cicci^{1,2}, Stefania Fresca¹, Mengwu Guo³, Andrea Manzoni¹, and Paolo Zunino¹

¹MOX-Dipartimento di Matematica, Politecnico di Milano, Italy*

²School of Biomedical Engineering & Imaging Sciences, King's College London, UK

³Department of Applied Mathematics, University of Twente, the Netherlands†

Abstract

Uncertainty quantification (UQ) tasks, such as sensitivity analysis and parameter estimation, entail a huge computational complexity when dealing with input-output maps involving the solution of nonlinear differential problems, because of the need to query expensive numerical solvers repeatedly. Projection-based reduced order models (ROMs), such as the Galerkin-reduced basis (RB) method, have been extensively developed in the last decades to overcome the computational complexity of high fidelity full order models (FOMs), providing remarkable speedups when addressing UQ tasks related with parameterized differential problems. Nonetheless, constructing a projection-based ROM that can be efficiently queried usually requires extensive modifications to the original code, a task which is non-trivial for nonlinear problems, or even not possible at all when proprietary software is used. Non-intrusive ROMs – which rely on the FOM as a black box – have been recently developed to overcome this issue. In this work, we consider ROMs exploiting proper orthogonal decomposition to construct a reduced basis from a set of FOM snapshots, and Gaussian process regression (GPR) to approximate the RB projection coefficients. Two different approaches, namely a global GPR and a tensor-decomposition-based GPR, are explored on a set of 3D time-dependent solid mechanics examples. Finally, the non-intrusive ROM is exploited to perform global sensitivity analysis (relying on both screening and variance-based methods) and parameter estimation (through Markov chain Monte Carlo methods), showing remarkable computational speedups and very good accuracy compared to high-fidelity FOMs.

Keywords: uncertainty quantification; reduced order modeling; Gaussian process regression; nonlinear solid mechanics; sensitivity analysis; parameter estimation

1 Introduction

Applied sciences and engineering problems, such as those arising in structural mechanics, are often described in terms of (time-dependent, nonlinear) partial differential equations (PDEs) that may be parameterized, i.e., involving several parameters to account for, e.g., different material properties, source terms, data, or geometrical features. Discrete, high-fidelity approximations of the PDE solutions can be computed by means of full order models (FOMs), which are usually computationally demanding - both in terms of CPU time and memory requirements - as fine computational grids and small time steps are needed to ensure accuracy of the solution. Despite the computational power available nowadays, relying on FOMs remains prohibitive in multi-query contexts, such as in the case of uncertainty quantification (UQ) tasks.

When dealing with differential models hampered by uncertainty on the (possibly, many) input parameters, it may be useful for multiple purposes (i) to detect and rank those inputs which need to be measured in order to reduce the output variance, (ii) to detect the parameters that have a better chance of being estimated in a subsequent estimation process, and (iii) to identify non-influential inputs in order to fix them to nominal values within their range of variability. To perform these tasks, global sensitivity analysis (SA) methods, such as screening methods [Mor91] and variance-based methods [Sob90], are commonly used in a variety

*{ludovica.cicci,stefania.fresca,andrea1.manzoni,paolo.zunino}@polimi.it

†m.guo@utwente.nl

of applications. However, computing sensitivity indices in the case of high-dimensional parameter spaces requires a large number of input-output evaluations (especially when SA is performed through Monte Carlo integration), so that the overall computational cost can become prohibitively expensive when relying on high-fidelity FOMs. Efficient surrogate models may be built [LGMS17], e.g., to compute the sensitivity indices directly using analytic formulas based on suitable regression predictors [Sud08, CLLZ20], or to alleviate the cost of each forward simulation, eventually in a multi-fidelity framework [QPO⁺18].

Another important problem addressed in real-world applications is the necessity to recover information about input parameters from limited and noisy observations of quantities of interest (QoIs), that are outputs computed using the problem solution. When addressing parameter estimation in a Bayesian framework, the solution to this inverse UQ problem is provided in terms of the posterior distribution of the unknown inputs conditioned on the observations. However, this distribution is not known analytically, and one needs to rely on sampling techniques, such as Markov chain Monte Carlo (MCMC), to characterize it and derive suitable statistics. Nonetheless, these methods require repeated (often, millions of) evaluations of the input-output map, so that relying on expensive high-fidelity FOMs becomes unfeasible. Different approaches can be employed to reduce the cost of solving parameter estimation problems [FMWvBW10], such as lowering the number of forward simulations required, reducing the dimensionality of the input space, or relying on efficient surrogate models for the solution to the forward problem [LM14, LMQR14, CMW15, YMK21].

To tackle both SA and inverse UQ scenarios, we adopt a semi-intrusive approach by relying on a reduced order model (ROM) to compute efficient sensitivity estimates and posterior densities of the problem parameters, while keeping a good level of accuracy. ROMs featuring smaller dimension, i.e., a lower number of degrees of freedom (DOFs), have been extensively developed during the last decades to overcome the computational complexity of high-fidelity FOMs. Many of these reduction approaches rely on the assumption that the high-fidelity solution manifold, that is the set of all FOM solutions as the input parameters change, can be well approximated by a low-dimensional linear space, and employ the proper orthogonal decomposition (POD) to build such a reduced basis (RB) able to capture the most dominant features of the original system. The high-dimensional solution is thus reconstructed as a linear combination of these (hopefully) few modes with a small loss of accuracy. In particular, the evaluation of high-fidelity solutions at several points in the time and parameter domains, the construction of the POD basis and the assembling of the ROM arrays, can be performed only once, offline. Then, for unseen time-parameter instances, the RB coefficients can be rapidly computed online, in order to reconstruct the high-dimensional solution. The key of success of these ROMs lies in the full decoupling of the two stages, meaning that the online phase is independent of the high-fidelity dimension. ROMs can then be divided into intrusive and non-intrusive, according to the way the ROM approximation is generated and the reduced arrays are built.

Intrusive projection-based ROMs (see, e.g., [BGW15] and references therein), such as reduced basis methods [QMN16, HRS16], are constructed by projecting the high-fidelity model operators onto the linear subspace spanned by a set of problem-dependent basis functions. These methods have been successfully applied to a wide range of problems [DHO12, AZF12, RR16, GTS17, BMQ17], as they allow to retain the underlying structure of the high-fidelity FOM and to approximate the whole field variables that solve the FOM. Nonetheless, they require extensive code modifications, which might be non-trivial or even not possible when proprietary software is used. Moreover, in the case of nonlinear problems, or problems featuring a non-affine parametric dependence, assembling ROM operators requires a further level of hyper-reduction, relying on, e.g., the empirical interpolation method (EIM) [BMNP04] and its variants [CS10, TDR13, LMQR14, NMA15]. To overcome this difficulty, deep learning-based strategies for the approximations of nonlinear ROM operators have been recently developed in [CFM22], showing a remarkable speed-up of the online phase also for complex time-dependent nonlinear applications, as reported in [CFMQ22]. Nonetheless, at every time step one must still solve the reduced problem and compute the solution vector, thus hampering the efficiency of the ROM in those cases where only scalar or low-dimensional QoIs need to be evaluated, or only the solution at prescribed time instances is required.

To overcome these issues, alternative data-driven methods can be used for the approximation of the RB coefficients without resorting to (Galerkin) projection. In these cases, the FOM solution is projected onto the RB space and the combination coefficients are approximated by means of a surrogate model, exploiting, e.g., Gaussian process regression (GPR) [GH18, GH19, KGH20, ZGH19], radial basis function (RBF) approximation [ADVN13] or artificial neural networks (ANNs) [HU18, GLH⁺21, SDM21]. The high-fidelity solver is thus used only offline as a “black-box” to generate the necessary data for the RB construction and the training of the surrogate model. Non-intrusive POD-based approaches using RBFs to interpolate the ROM

coefficients have been proposed in [ADVN13, WHM13, XFPN17] and applied to nonlinear parameterized time-dependent PDEs. In [HU18] ANNs have been employed to build a regression model to compute the coefficients of a POD-based ROM, focusing on steady PDEs; a further extension to time-dependent nonlinear problems, i.e., unsteady flows, has been addressed in [WHR19]; see also, e.g., [SMPW19, YYG19] for the use of machine learning strategies to approximate the POD coefficient. Alternatively, POD-GPR reduced order models have been developed in [GH18] for steady nonlinear structural analysis and in [GH19] for time-dependent problems, and successfully used in engineering applications, such as electromagnetic scattering problems [ZLL21] and naval engineering problems [ODR20]. Non-intrusive ROMs using RBF, GPR, and ANN regressions are compared in [BHKE20] and applied to complex flow problems in a time-dependent, deforming computational domain. See instead [GMA⁺22, CGMH23] for an alternative use of ANNs to perform regression in the context of multi-fidelity methods relying on models of different fidelities. These latter may involve data-driven projection-based [PW16, GMW22] ROMs or recently developed deep learning-based ROMs [FDM21, FM22, FMZ23, BGB22].

In this work, we apply non-intrusive ROMs for the efficient solution to nonlinear time-dependent parameterized problems arising in solid mechanics, in order to accelerate SA and the solution to Bayesian inverse problems. Once POD has been used to construct a low-dimensional linear subspace – thus providing a way to embed physics into the ROM and to make it explainable – the RB expansion coefficients are approximated by means of GPR models, taking into account (and comparing) two different approaches [GH18] and [GH19], both ensuring non-intrusiveness and online efficiency. With respect to other data-driven approaches, such as DNNs, the advantages of POD-GPR ROMs are twofold: on one hand, they require a relatively small amount of data to be trained; on the other hand, they allow to reconstruct the whole solution field, so that any quantity of interest can be evaluated without the need to retrain the model, whereas surrogate models would directly approximate the input-output map. Moreover, relying on GPRs allows to take into account the emulator uncertainty, which represents a further source of uncertainty in our problem.

Compared to the existing literature – see, e.g., [GH18, GH19] – we address in this paper the approximation of time-dependent, nonlinear problems set over three-dimensional domains, also featuring a rather large number of input parameters. Moreover, a full UQ pipeline is shown, including both sensitivity analysis and parameter estimation, exploiting the non-intrusive ROM approximation enabled by the combined POD-GPR framework, and assessing its accuracy against the results obtained with a high-fidelity FOM based on the finite element method.

This work is structured as follows. In Section 2, we review some basic facts about the RB method for time-dependent parameterized problems, describing in Section 3 two different approaches based on GPRs for the efficient approximation of the RB coefficients. Numerical results related with two test cases in nonlinear solid mechanics are then presented in Section 4, where POD-GPR ROMs are compared in terms of efficiency and accuracy. In Section 5 the solution to SA and inverse UQ problems is carried out by means of the POD-GPR ROMs. Finally, some conclusions are drawn in Section 6.

2 RB methods for parametrized PDEs: basic facts and notation

Our goal is to efficiently solve nonlinear time-dependent PDE problems depending on a set of input parameters $\boldsymbol{\mu} \in \mathcal{P}$, where $\mathcal{P} \subset \mathbb{R}^p$ is a compact set representing the parameter space, and p is the number of inputs. In particular, we focus on problems governed by the elastodynamics equation

$$\rho \partial_t^2 \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) - \nabla_{\mathbf{X}} \cdot \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} \text{ in } \Omega_0 \times \mathcal{T}, \quad (1)$$

suitably complemented by initial and boundary conditions. We denote the unknown state solution, i.e., the displacement field, by $\mathbf{u} = \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu})$, depending on the spatial coordinate $\mathbf{X} \in \Omega_0 \subset \mathbb{R}^3$ in the reference configuration, the time variable $t \in \mathcal{T} \subset \mathbb{R}$ and the parameter vector $\boldsymbol{\mu} \in \mathcal{P}$, and by \mathbf{P} the first Piola-Kirchhoff stress tensor. Moreover, ρ refers to the material density, which is assumed to be constant throughout the deformation. Once this problem has been discretized in space relying, e.g., on the Galerkin finite element method, a dynamical system of dimension N_h is obtained under the form

$$\begin{cases} \rho \mathbf{M} \ddot{\mathbf{u}}_h(t; \boldsymbol{\mu}) + \mathbf{N}(\mathbf{u}_h(t; \boldsymbol{\mu}), t; \boldsymbol{\mu}) = \mathbf{F}^{ext}(t; \boldsymbol{\mu}) & \text{in } \mathcal{T}, \\ \mathbf{u}_h(0; \boldsymbol{\mu}) = \mathbf{u}_{h,0}(\boldsymbol{\mu}), \quad \dot{\mathbf{u}}_h(0; \boldsymbol{\mu}) = \dot{\mathbf{u}}_{h,0}(\boldsymbol{\mu}), \end{cases} \quad (2)$$

where $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ denotes the mass matrix, $\mathbf{N} \in \mathbb{R}^{N_h}$ is the (nonlinear) vector of internal forces, $\mathbf{F}^{ext} \in \mathbb{R}^{N_h}$ is the vector of external forces and $\mathbf{u}_h(t) \in \mathbb{R}^{N_h}$ collects the degrees of freedom (DOFs) that represent

the (semi-discrete) high fidelity solution at time t . Note that fully implicit schemes are employed for the sake of numerical stability, since they do not pose restrictions on the time step, otherwise required due to the highly nonlinear terms of the strain energy density function considered in this work. Here, N_h denotes the number of DOFs, which depends on the underlying computational mesh and the space discretization used. Relying on finite differences for time discretization, and therefore approximating

$$\dot{\mathbf{u}}_h(t^n, \boldsymbol{\mu}) \approx \frac{\mathbf{u}_h(t^n, \boldsymbol{\mu}) - \mathbf{u}_h(t^{n-1}, \boldsymbol{\mu})}{\Delta t}, \quad \ddot{\mathbf{u}}_h(t^n, \boldsymbol{\mu}) \approx \frac{\mathbf{u}_h(t^n, \boldsymbol{\mu}) - 2\mathbf{u}_h(t^{n-1}, \boldsymbol{\mu}) + \mathbf{u}_h(t^{n-2}, \boldsymbol{\mu})}{\Delta t^2},$$

where $\{t^0 < t^1 < \dots < t^{N_t}\}$ is a uniform partition of \mathcal{T} with time step Δt , we obtain a sequence of fully-discrete problems that in algebraic form read as follows: given $\boldsymbol{\mu} \in \mathcal{P}$, for t^n , $n = 1, \dots, N_t$, find the discrete high-fidelity displacement $\mathbf{u}_h(t^n; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\frac{\rho}{\Delta t^2} \mathbf{M} \mathbf{u}_h(t^n, \boldsymbol{\mu}) + \mathbf{N}(\mathbf{u}_h(t^n, \boldsymbol{\mu}), t^n; \boldsymbol{\mu}) - \frac{2\rho}{\Delta t^2} \mathbf{M} \mathbf{u}_h(t^{n-1}, \boldsymbol{\mu}) + \frac{\rho}{\Delta t^2} \mathbf{M} \mathbf{u}_h(t^{n-2}, \boldsymbol{\mu}) - \mathbf{F}^{ext}(t^n, \boldsymbol{\mu}) = \mathbf{0}, \quad (3)$$

where $\mathbf{u}_h(t^{-1}, \boldsymbol{\mu})$ and $\mathbf{u}_h(t^0, \boldsymbol{\mu})$ are given by the initial conditions. The solution to the nonlinear system (3) can be finally computed by means of iterative algorithms, such as the Newton-Raphson method. However, to achieve sufficient accuracy for real-world applications, either the FOM dimension N_h or the number of time steps N_t , as well as the total Newton iterations, can become extremely large, implying huge computational costs.

Therefore, we rely on the RB method [QMN16, HRS16, HPR22] to seek a low-dimensional approximation of $\mathbf{u}_h(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ as a linear combination of $N \ll N_h$ parameter-independent basis functions constructed by performing proper orthogonal decomposition (POD) on a set of sampled FOM solutions. By denoting as $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ the matrix collecting column-wise the DOFs of the RB functions, a reduced order solution $\mathbf{u}_h^{RB}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ can be computed by projecting the FOM solution onto $\text{Col}(\mathbf{V})$, i.e.,

$$\mathbf{u}_h^{RB}(t; \boldsymbol{\mu}) = \mathbf{V} \mathbf{V}^T \mathbf{u}_h(t; \boldsymbol{\mu}),$$

so that we only need to recover the underlying map between $(t, \boldsymbol{\mu})$ and the reduced coefficients, that is

$$\mathbf{q}: (t, \boldsymbol{\mu}) \mapsto \mathbf{V}^T \mathbf{u}_h(t; \boldsymbol{\mu}),$$

for $(t; \boldsymbol{\mu}) \in \mathcal{T} \times \mathcal{P}$. Note that, in this case, it holds $\mathbf{u}_h^{RB}(t; \boldsymbol{\mu}) = \arg \min_{\mathbf{x}_h^* \in \text{Col}(\mathbf{V})} \|\mathbf{u}_h(t; \boldsymbol{\mu}) - \mathbf{x}_h^*\|$. However, when employing a Galerkin projection to compute the RB coefficients, the assembly of the reduced system still depends on the high-fidelity dimension due to nonlinearity of the problem, so that a further level of hyper-reduction is required to ensure computational efficiency, see, e.g., [BMNP04, CS10, NMA15, TR13]. The RB method thus yields an intrusive strategy to generate the reduced-order solution, with hyper-reduction techniques possibly hampering the convergence properties of the nonlinear solvers at the ROM level. Based on [GH18, GH19], we resort to nonlinear regression to build efficient non-intrusive ROMs and obtain probabilistic distributions of the projection coefficients for each new value of the input vector, useful to perform UQ studies.

3 Reduced order surrogate modeling with GP Regression

To obtain the projection coefficients $\mathbf{q}(t; \boldsymbol{\mu})$, for $(t, \boldsymbol{\mu}) \in \mathcal{T} \times \mathcal{P}$, and thus recover the RB solution $\mathbf{u}_h^{RB}(t; \boldsymbol{\mu}) \approx \mathbf{u}_h(t; \boldsymbol{\mu})$, we rely on two different regression-based approaches, to which we refer to as *global*, outlined in Section 3.2, and *tensor-decomposition-based*, proposed in [GH19] and described in Section 3.3. We point out that, since the RB coefficients are uncorrelated, we rely on independent single-output GPs. This strategy has the great advantage that the output to each GP is a scalar, thus requiring the inversion of a matrix of dimension N_{data} , rather than $N \cdot N_{data}$ as formulated in vector-valued GPRs. Since the time required to invert a matrix scales almost cubically, a lower overall training cost is expected for single-output GPs with respect to multi-output ones. Moreover, multiple GPs can be trained simultaneously on different processors.

3.1 Gaussian Process (GP) Regression

Given a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R} \mid i = 1, \dots, N_{data}\}$, where

$$\mathbf{X} = [\mathbf{x}^{(1)} \mid \dots \mid \mathbf{x}^{(N_{data})}] \quad \text{and} \quad \mathbf{y} = [y^{(1)}, \dots, y^{(N_{data})}]^T$$

are the observed input matrix and the associated output vector, respectively, one wants to build a model $f: \mathcal{X} \rightarrow \mathcal{Y}$ which is able to infer the relationship between the d -dimensional independent variable \mathbf{x} and the scalar dependent variable y , hence to determine the conditional distribution of the target output given the input. Since there may be more than a regression function f that fit the data equally well, a Gaussian process (GP) [WR06] is used to assign a probability to each of these functions. Hence, given a mean $m(\mathbf{x})$ and a covariance function (or kernel) $\kappa(\mathbf{x}, \mathbf{x}')$, one assumes that the prior of f is a GP and write

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')).$$

Remark 1. Typically, the mean is chosen to be constant, e.g., equal to the mean of the training set (or to zero), whereas many different choices are available for the covariance $\kappa: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, such as the radial basis function (RBF) kernel

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\gamma^2}\right) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{j=1}^d \frac{(x_j - x'_j)^2}{\gamma^2}\right), \quad (4)$$

whose hyperparameters are the signal variance σ_f^2 and the length scale γ . Alternatively, the automatic relevance determination (ARD) RBF kernel assumes each dimension as independent from the others, thus allowing for different length scales γ_j , for $j = 1, \dots, d$. A non-stationary covariance function may also be defined via a deep neural network architecture [LBN⁺17, Guo21].

The goal of GP regression is to incorporate the knowledge provided by the training data \mathcal{D} into the prior to obtain the posterior (predictive) distribution. Even though in this work we rely on synthetic data, GPs can be trained on experimental outputs, so that measurement errors should be taken into account. Hence, in order to be as extensive as possible, we assume that the training output labels are corrupted by an additive independent identically distributed Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$, i.e., $y^{(i)} = f(\mathbf{x}^{(i)}) + \epsilon$, for $i = 1, \dots, N_{data}$. Given a new set of testing point $\mathbf{x}^* \notin \mathcal{D}$, according to the theorem of conditional Gaussian (see, e.g., [Mur12]), the conditional distribution of the predicted GP realization is given by

$$f(\mathbf{x}^*) \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^* \sim \text{GP}(\widehat{m}(\mathbf{x}^*), \widehat{\kappa}(\mathbf{x}^*, \mathbf{x}^*)),$$

where

$$\begin{aligned} \widehat{m}(\mathbf{x}^*) &= m(\mathbf{x}^*) + \kappa(\mathbf{x}^*, \mathbf{X})(\kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I})^{-1}(\mathbf{y} - m(\mathbf{X})), \\ \widehat{\kappa}(\mathbf{x}^*, \mathbf{x}^{*'}) &= \kappa(\mathbf{x}^*, \mathbf{x}^{*'}) - \kappa(\mathbf{x}^*, \mathbf{X})(\kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I})^{-1} \kappa(\mathbf{X}, \mathbf{x}^{*'}). \end{aligned}$$

The hyperparameters of the GPR model, such as $\boldsymbol{\theta} = (\sigma_y, \sigma_f, \gamma)$ for the RBF kernel, highly influence the predictive performances and should be optimized, e.g., by maximizing the log-likelihood

$$\ln(\pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})) = -\frac{1}{2} (\mathbf{y}^T (\kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y} + \ln(\det(\kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I})) + N_{data} \ln(2\pi)). \quad (5)$$

Once the hyperparameters have been optimized on the training set \mathcal{D} , the GPR model can be used to perform inference on an unseen test set, using for the regression the mean function calculated by the posterior, denoted from now on – with a slight abuse of notation – as \widehat{f} .

Remark 2. A common practice to handle inputs of varying magnitude is to perform feature scaling on the training and test datasets. Widely used transformations of the data $\mathbf{v} \in \mathbb{R}^m$ and $\mathbf{M} \in \mathbb{R}^{m \times n}$ are:

1. the min-max normalization $\boldsymbol{\xi}_{min-max}^{\mathbf{v}}$ and $\boldsymbol{\xi}_{min-max}^{\mathbf{M}}$, such that for $i = 1, \dots, m$,

$$(\boldsymbol{\xi}_{min-max}^{\mathbf{v}})_i: \mathbf{v}_i \mapsto \frac{\mathbf{v}_i - \min \mathbf{v}}{\max \mathbf{v} - \min \mathbf{v}}, \quad (\boldsymbol{\xi}_{min-max}^{\mathbf{M}})_i: \mathbf{M}_i \mapsto \frac{\mathbf{M}_i - \min_k \mathbf{M}_{ik}}{\max_k \mathbf{M}_{ik} - \min_k \mathbf{M}_{ik}}; \quad (6)$$

2. the standardization $\boldsymbol{\xi}_{stand}^{\mathbf{M}}$ or $\boldsymbol{\xi}_{stand}^{\mathbf{v}}$, such that for $i = 1, \dots, m$,

$$(\boldsymbol{\xi}_{stand}^{\mathbf{v}})_i: \mathbf{v}_i \mapsto \frac{\mathbf{v}_i - \text{mean } \mathbf{v}}{\text{std } \mathbf{v}}, \quad (\boldsymbol{\xi}_{stand}^{\mathbf{M}})_i: \mathbf{M}_i \mapsto \frac{\mathbf{M}_i - \text{mean}_k \mathbf{M}_{ik}}{\text{std}_k \mathbf{M}_{ik}}. \quad (7)$$

3.2 Global approach

In a first attempt, our goal is to directly learn the map between the time/parameter values and the projection coefficients onto the RB space, that is

$$(t, \boldsymbol{\mu}) \mapsto \mathbf{q}(t; \boldsymbol{\mu}) = \mathbf{V}^T \mathbf{u}_h(t; \boldsymbol{\mu}),$$

by means of a regression model. In particular, we want to infer each RB coefficient $q_\ell(t; \boldsymbol{\mu})$, for $\ell = 1, \dots, N$, by means of a GP trained on a $N_t N_s$ -dimensional dataset of input-output pairs $(\mathbf{X}, \mathbf{y}_\ell)$, where

$$\mathbf{X} = \left[\begin{array}{ccc|ccc} t^1 & \dots & t^{N_t} & \dots & t^1 & \dots & t^{N_t} \\ \boldsymbol{\mu}_1 & \dots & \boldsymbol{\mu}_1 & \dots & \boldsymbol{\mu}_{N_s} & \dots & \boldsymbol{\mu}_{N_s} \end{array} \right] \in \mathbb{R}^{(p+1) \times N_t N_s},$$

$$\mathbf{y}_\ell = [q_\ell(t^1; \boldsymbol{\mu}_1), \dots, q_\ell(t^{N_t}; \boldsymbol{\mu}_1) \mid \dots \mid q_\ell(t^1; \boldsymbol{\mu}_{N_s}), \dots, q_\ell(t^{N_t}; \boldsymbol{\mu}_{N_s})]^T + \boldsymbol{\varepsilon} \in \mathbb{R}^{N_t N_s},$$

such that the mean of the posterior distribution, for $\ell = 1, \dots, N$, can be used to make predictions for every new input $\mathbf{x}^* = (t^*, \boldsymbol{\mu}^*) \in \mathcal{T} \times \mathcal{P}$. This approach has been proposed in [GH18] for nonlinear structural analysis, however restricted to quasi-static problems, and we refer to it as *global*, since time and parameter are treated as a unique input variable. According to the notation of Section 3.1, we have $N_{data} = N_t N_s$, $\mathbf{x}^{(i)} = (t^{n_i}, \boldsymbol{\mu}_{m_i})$ and $y^{(i)} = q_\ell(t^{n_i}, \boldsymbol{\mu}_{m_i}) + \varepsilon$, for $\ell = 1, \dots, N$, where $n_i \in \{1, \dots, N_t\}$ and $m_i \in \{1, \dots, N_s\}$.

To handle values of different magnitudes, we apply the transformation (6) (or (7)) reported in Remark 2 to $(\mathbf{X}, \mathbf{X}^*)$ and $(\mathbf{y}_\ell, \hat{f}(\mathbf{X}^*))$, for $\ell = 1, \dots, N$, with $\mathbf{M} = \mathbf{X}$ and $\mathbf{v} = \mathbf{y}_\ell$, respectively. Here, $\mathbf{X}^* \in \mathbb{R}^{(p+1) \times N_t N_s^{test}}$ denotes the matrix whose columns are inputs \mathbf{x}^* unseen during training. Note that the minimum and the maximum (or the mean and the standard deviation) are computed over the training arrays.

3.3 Tensor-decomposition-based regression approach

A different approach is the so-called tensor-decomposition-based regression developed in [GH19], here briefly described. Let $\mathbb{P} \in \mathbb{R}^{N_t \times N_s \times (p+1)}$ be such that

$$\mathbb{P}_{n,m,:} = \begin{bmatrix} t^n \\ \boldsymbol{\mu}_m \end{bmatrix}, \text{ for } n = 1, \dots, N_t, \quad m = 1, \dots, N_s,$$

and let $\mathbb{Q}_\ell \in \mathbb{R}^{N_t \times N_s}$ be the corresponding output matrix for the ℓ -th RB coefficient, that is

$$\mathbb{Q}_\ell = \begin{bmatrix} q_\ell(t^1; \boldsymbol{\mu}_1) & \dots & q_\ell(t^1; \boldsymbol{\mu}_{N_s}) \\ \vdots & & \vdots \\ q_\ell(t^{N_t}; \boldsymbol{\mu}_1) & \dots & q_\ell(t^{N_t}; \boldsymbol{\mu}_{N_s}) \end{bmatrix},$$

for $\ell = 1, \dots, N$. The (truncated) singular value decomposition (SVD) of \mathbb{Q}_ℓ can be written as

$$\mathbb{Q}_\ell \approx \boldsymbol{\Psi}^\ell \boldsymbol{\Lambda}^\ell (\boldsymbol{\Phi}^\ell)^T = \sum_{k=1}^{N_{q_\ell}} \lambda_k^\ell \boldsymbol{\Psi}_k^\ell (\boldsymbol{\Phi}_k^\ell)^T,$$

where $\boldsymbol{\Psi}_k^\ell$ and $\boldsymbol{\Phi}_k^\ell$ are the k -th discrete time- and parameter-modes, respectively, and the truncation rank N_{q_ℓ} is obtained for a chosen value of the SVD tolerance $\varepsilon_{SVD} > 0$, according to the percentage of variance one wants to account for. By splitting \mathbb{P} into two input matrices, we define the following arrays

$$\mathbf{X}_t = \begin{bmatrix} t^1 \\ \vdots \\ t^{N_t} \end{bmatrix} \in \mathbb{R}^{N_t \times 1}, \quad \mathbf{X}_\mu = \begin{bmatrix} (\boldsymbol{\mu}_1)^T \\ \vdots \\ (\boldsymbol{\mu}_{N_s})^T \end{bmatrix} \in \mathbb{R}^{N_s \times p}$$

related to the time instances and the model parameters, respectively, and the corresponding target outputs $\mathbf{y}_{\ell,t} := \boldsymbol{\Psi}^\ell \in \mathbb{R}^{N_t \times N_{q_\ell}}$ and $\mathbf{y}_{\ell,\mu} := \boldsymbol{\Phi}^\ell \in \mathbb{R}^{N_s \times N_{q_\ell}}$, possibly corrupted by noise.

For $k = 1, \dots, N_{q_\ell}$, our aim is to learn the k -th continuous time-mode $\widehat{\psi}_k^\ell(t)$ given $\{\mathbf{X}_t, (\mathbf{y}_{\ell,t})_{1:N_t,k}\}$, and the k -th continuous parameter-mode $\widehat{\phi}_k^\ell(\boldsymbol{\mu})$ given $\{\mathbf{X}_\mu, (\mathbf{y}_{\ell,\mu})_{1:N_s,k}\}$, so that, for a new value $\mathbf{x}^* = (t^*; \boldsymbol{\mu}^*) \in \mathcal{T} \times \mathcal{P}$, we can compute

$$q_\ell(\mathbf{x}^*) \approx \widehat{q}_\ell(\mathbf{x}^*) = \sum_{k=1}^{N_{q_\ell}} \lambda_k^\ell \widehat{\psi}_k^\ell(t^*) \widehat{\phi}_k^\ell(\boldsymbol{\mu}^*).$$

Remark 3. For each component of the truncated SVD, i.e., for $k = 1, \dots, N_{q_\ell}$, two Gaussian processes – one for the time variable and the other for the parameter vector variable – are trained, so that $2 \sum_{\ell=1, \dots, N} N_{q_\ell}$ GPs are considered. Nonetheless, the dimensions of the training datasets are now much smaller with respect to the previous global approach (N_t or N_s instead of $N_t N_s$), thus reducing the overall computational time required for computing the optimal hyper-parameters.

As before, we employ two scaling strategies, namely the min-max normalization and the standardization. In this case, the transformations in (6) or in (7) are applied to the time- and parameter-datasets separately. To be more specific, the input datasets $(\mathbf{X}_t, \mathbf{X}_t^*)$ and $(\mathbf{X}_\mu, \mathbf{X}_\mu^*)$ are scaled according to maps $\boldsymbol{\xi}_\star^{\mathbf{X}_t}$ and $\boldsymbol{\xi}_\star^{\mathbf{X}_\mu}$, respectively, where $\star \in \{\text{min-max, stand}\}$. Meanwhile, for $\ell = 1, \dots, N$ and $k = 1, \dots, N_{q_\ell}$, the output datasets are scaled by applying $\boldsymbol{\xi}_\star^{(\mathbf{y}_{\ell,t})_{1:N_t,k}}$ to $((\mathbf{y}_{\ell,t})_{1:N_t,k}, \widehat{\psi}_k^\ell(\mathbf{X}_t^*))$ and $\boldsymbol{\xi}_\star^{(\mathbf{y}_{\ell,\mu})_{1:N_s,k}}$ to $((\mathbf{y}_{\ell,\mu})_{1:N_s,k}, \widehat{\phi}_k^\ell(\mathbf{X}_\mu^*))$. We recall that $\mathbf{X}_t^* \in \mathbb{R}^{(p+1) \times N_t}$ and $\mathbf{X}_\mu^* \in \mathbb{R}^{(p+1) \times N_s^{test}}$ denote the matrices whose columns are inputs t^* and $\boldsymbol{\mu}^*$ unseen during training, respectively.

4 Numerical results

In this section, we present the results obtained on two benchmark problems inspired by those used for verification of cardiac mechanics software [LGA⁺15], namely the deformation of a clamped rectangular beam and the active contraction of a truncated ellipsoid, governed by the elastodynamics equation (1). In particular, we focus on soft tissue mechanics [HO03], a vivid area of research in computational mechanics characterized by many difficulties due to the presence of strong material anisotropy, large deformations, highly nonlinear stress-strain behaviors. Although specific for this field, the proposed test cases can also be of interest for other fields of application sharing similar features. For both test cases, we evaluate the performances of the global and the tensor-decomposition-based approaches, either in terms of training time, computational speed-up with respect to the high fidelity model and accuracy. Specifically, to test the fidelity of the GPR predictions on the RB coefficients q_ℓ , for $\ell = 1, \dots, N$, over the test set, we use the mean square error (MSE)

$$\text{MSE}(q_\ell) = \frac{1}{N_s^{test} N_t^{test}} \sum_{m=1}^{N_s^{test}} \sum_{n=1}^{N_t^{test}} (q_\ell(t^n; \boldsymbol{\mu}_m) - \widehat{q}_\ell(t^n; \boldsymbol{\mu}_m))^2$$

and the relative squared error (RSE)

$$\text{RSE}(q_\ell) = \frac{\sum_{m=1}^{N_s^{test}} \sum_{n=1}^{N_t^{test}} (q_\ell(t^n; \boldsymbol{\mu}_m) - \widehat{q}_\ell(t^n; \boldsymbol{\mu}_m))^2}{\sum_{m=1}^{N_s^{test}} \sum_{n=1}^{N_t^{test}} (q_\ell(t^n; \boldsymbol{\mu}_m) - \bar{q}_\ell)^2}, \quad \text{with } \bar{q}_\ell = \frac{1}{N_s^{test} N_t^{test}} \sum_{m=1}^{N_s^{test}} \sum_{n=1}^{N_t^{test}} q_\ell(t^n; \boldsymbol{\mu}_m).$$

The overall accuracy of the ROMs with respect to the FOM is evaluated using the time-averaged absolute and relative L^2 -errors defined as

$$\begin{aligned} \text{tAE}(\boldsymbol{\mu}) &= \frac{1}{N_t^{test}} \sum_{n=1}^{N_t^{test}} \|\mathbf{u}_h(t^n; \boldsymbol{\mu}) - \mathbf{V} \widehat{\mathbf{q}}(t^n; \boldsymbol{\mu})\|_{L^2}, \\ \text{tRE}(\boldsymbol{\mu}) &= \frac{1}{N_t^{test}} \sum_{n=1}^{N_t^{test}} \frac{\|\mathbf{u}_h(t^n; \boldsymbol{\mu}) - \mathbf{V} \widehat{\mathbf{q}}(t^n; \boldsymbol{\mu})\|_{L^2}}{\|\mathbf{u}_h(t^n; \boldsymbol{\mu})\|_{L^2}}, \end{aligned}$$

respectively, and their mean over the test set, i.e., $\text{mean}_{m=1, \dots, N_s^{test}} \text{tAE}(\boldsymbol{\mu}_m)$ and $\text{mean}_{m=1, \dots, N_s^{test}} \text{tRE}(\boldsymbol{\mu}_m)$.

The GPR models are implemented in Python using the library GPy [GP12], a Gaussian processes framework. All the computations have been performed on a Notebook with 2.60GHz Intel Core i7-9750H CPU and 16GB RAM.

Remark 4. *Numerical simulations have been run in serial and GPs have been trained and tested sequentially, i.e., on a outer loop from $\ell = 1$ to $\ell = N$. However, better performances with respect to CPU time can be achieved by exploiting the independence of the GPR models associated to different RB coefficients, thus performing training and testing in parallel.*

4.1 Benchmark tests for nonlinear solid mechanics

For the setup of the benchmark test cases, we partition the boundary in three regions, corresponding to displacement, pressure and stress-free conditions respectively. More precisely, we set Dirichlet boundary conditions on a specific portion of the boundary $\partial\Omega_0^D$ and the pressure load

$$\mathbf{p}(t; \boldsymbol{\mu}) = \tilde{p} \frac{t}{T}$$

is applied on $\partial\Omega_0^P$. Moreover, Neumann conditions are set on all other surfaces, i.e., $\partial\Omega_0^N = \partial\Omega_0 \setminus (\partial\Omega_0^D \cup \partial\Omega_0^P)$, and homogeneous initial conditions are prescribed at time $t = 0$ s.

Equation (1) must be further complemented with a suitable constitutive relation for the material which can be expressed, e.g., as a nonlinear operator mapping the Green-Lagrange strain tensor

$$\mathbf{E}(\mathbf{u}) = \frac{1}{2}(\mathbf{F}(\mathbf{u})^T \mathbf{F}(\mathbf{u}) - \mathbf{I})$$

to the first Piola-Kirchhoff stress tensor $\mathbf{P}(\mathbf{u})$, where $\mathbf{F}(\mathbf{u}) = \mathbf{I} + \nabla_{\mathbf{X}} \mathbf{u}$ is the deformation gradient. In this work, we consider a nearly-incompressible hyperelastic material with the following strain-energy function

$$\mathcal{W}(\mathbf{F}) = \frac{C}{2}(e^{Q(\mathbf{E})}-1) + \frac{K}{2}(J-1)\ln(J), \quad (8)$$

such that $\mathbf{P}(\mathbf{u}) = \partial\mathcal{W}(\mathbf{F})/\partial\mathbf{F}$. The first term is given by the Guccione's constitutive law proposed in [GCM95], and the second one is used to enforce incompressibility. Here, $C > 0$ is a scaling constant, $J = \det(\mathbf{F}(\mathbf{u}))$ measures the change in volume during motion, and $K > 0$ is the bulk modulus penalizing large volume variations. In particular, we consider the following form for Q ,

$$Q = b_x E_{xx}^2 + b_y E_{yy}^2 + b_z E_{zz}^2 + b_{xy}(E_{xy}^2 + E_{yx}^2) + b_{xz}(E_{xz}^2 + E_{zx}^2) + b_{yz}(E_{yz}^2 + E_{zy}^2),$$

where b_i , for $i \in \{x, y, z\}$, are related to the material stiffness in the different directions, and E_{ij} , for $i, j \in \{x, y, z\}$, are the components of the strain tensor $\mathbf{E}(\mathbf{u})$. Therefore, we end up with the following initial-boundary value problem: given $\boldsymbol{\mu} \in \mathcal{P}$, find the parameterized displacement field $\mathbf{u}(\boldsymbol{\mu}): \Omega_0 \times (0, T) \rightarrow \mathbb{R}^3$ such that

$$\begin{cases} \rho \partial_t^2 \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) - \nabla_{\mathbf{X}} \cdot \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} & \text{in } \Omega_0 \times (0, T) \\ \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) = \mathbf{0} & \text{on } \partial\Omega_0^D \times (0, T) \\ \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}); \boldsymbol{\mu}) \boldsymbol{\nu} = -\mathbf{p}(t; \boldsymbol{\mu}) J \mathbf{F}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}); \boldsymbol{\mu})^{-T} \boldsymbol{\nu} & \text{on } \partial\Omega_0^P \times (0, T) \\ \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}); \boldsymbol{\mu}) \boldsymbol{\nu} = \mathbf{0} & \text{on } \partial\Omega_0^N \times (0, T) \\ \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) = \mathbf{0}, \partial_t \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) = \mathbf{0} & \text{in } \Omega_0 \times \{0\}, \end{cases} \quad (9)$$

where $\boldsymbol{\nu}$ denotes the outer normal unit vector.

4.1.1 Test 1: deformation of a beam

The first benchmark test deals with the deformation of a clamped rectangular beam due to a pressure load acting on the bottom face of the body. The problem is defined on the reference domain $\Omega_0 = [0, 10^{-2}] \text{m} \times [0, 10^{-3}] \text{m} \times [0, 10^{-3}] \text{m}$, reported in Figure 1 together with the computational hexahedral mesh. With respect to system (9), we define $\partial\Omega_0^D = \Omega_0 \cap \{x = 0\}$ and $\partial\Omega_0^P = \Omega_0 \cap \{z = 0\}$.

As input parameters, we consider those of the Guccione's law, i.e., $b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K$ and G , and the slope \tilde{p} of the pressure load, such that $\boldsymbol{\mu} = [b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K, G, \tilde{p}] \in \mathcal{P} = \bigcup_{i=1}^9 [l_i, r_i] \subset \mathbb{R}^9$, see Table 1.

The FOM is built on an hexahedral mesh with 1025 vertices using \mathbb{Q}_1 -FE in space and a BDF1 scheme in time with time step $\Delta t = 0.005$ s and final time $T = 0.25$ s, so that $N_h = 3075$ and $N_t = 50$. In this setting, the average CPU time required for computing the FOM solution for a given parameter vector is 48 s.

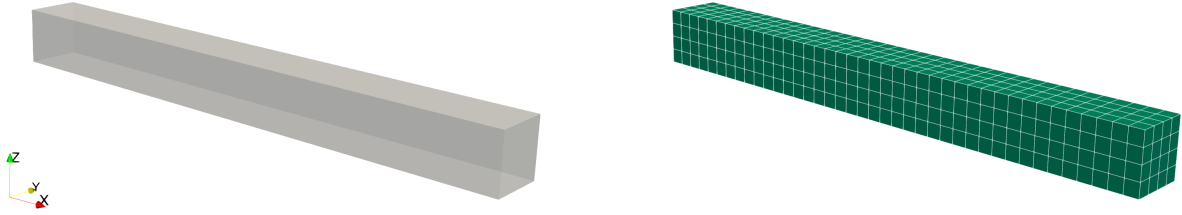


Figure 1: Deformation of a beam. Reference geometry (left) and computational mesh (right).

	b_f	b_s	b_n	b_{fs}	b_{fn}	b_{sn}	K [kPa]	C [kPa]	\tilde{p} [kPa]
l_i	4	1	1	2	2	1	25	1	0.002
r_i	12	3	3	6	6	3	75	3	0.006

Table 1: Deformation of a beam. Lower and upper bounds for the components of the parameter vector.

With the aim of analyzing and comparing the two POD-GPR approaches, namely the global and tensor-decomposition-based, we collect full-order snapshots at $N_s = 100$ parameter locations sampled in \mathcal{P} by latin hypercube sampling (LHS) for the construction of the RB functions. The same set solution vectors is then split (8:2) into training and testing datasets, and used for the construction of the GPR models. We point out that, unless otherwise specified, we use the same time points for training and testing, so that $N_t^{test} = N_t$.

Remark 5. For the POD-based ROMs to accurately approximate the FOM displacement field for any input parameter, one needs to guarantee that the snapshots used for the construction of the RB basis sufficiently capture the solution dynamics, so that the choice of the sampled parameter locations highly influences the accuracy of the ROM. Relying on random strategies may require a large number of high-fidelity snapshots, leading to extremely expensive offline computational times. To overcome this issue, [GH18, APLK19, GLH⁺21, CHJ⁺18] have adopted suitable adaptive sampling strategies. However, since the overall computational cost of the test cases considered in this work is not cumbersome, these options have not been taken into account. For instance, in [GH18] an active data selection strategy is adopted, based on standard-deviation-based error indicator; GPR performances are then evaluated on a training set of increasing size, up to the fulfillment of a suitable quality criterion. A locally adaptive sparse grid is employed in [APLK19], whereas an adaptive greedy approach is considered in [CHJ⁺18], based on the idea of selecting as samples those candidate points with highest ROM errors, according to a prescribed criterion.

4.1.2 Test 2: active contraction of a truncated ellipsoid

For the second problem, we assume as reference geometry Ω_0 the truncated ellipsoid described in [LGA⁺15] and reported in Figure 2, and consider a nearly-incompressible material governed by the *Guccione et al.* strain-energy function (8). Moreover, we integrate active tension into the passive stress tensor by adding a parameterized time-dependent term which is assumed to act only in the fiber direction, that is

$$\mathbf{P}(\mathbf{u}) = \frac{\partial \mathcal{W}(\mathbf{F})}{\partial \mathbf{F}} + T_a(t; \boldsymbol{\mu})(\mathbf{F}\mathbf{f} \otimes \mathbf{f}),$$

where $\mathbf{f} \in \mathbb{R}^3$ denotes the reference unit vector in the fiber direction and we assume

$$T_a(t; \boldsymbol{\mu}) = \tilde{T}_a \frac{t}{T},$$

for $\tilde{T}_a > 0$. The base plane $\partial\Omega_0^D = \Omega_0 \cap \{z = \bar{z}\}$ is fixed in all directions, and an external pressure is applied at the inner boundary $\partial\Omega_0^P$. Neumann conditions are finally applied at the outer surface $\partial\Omega_0^N$.

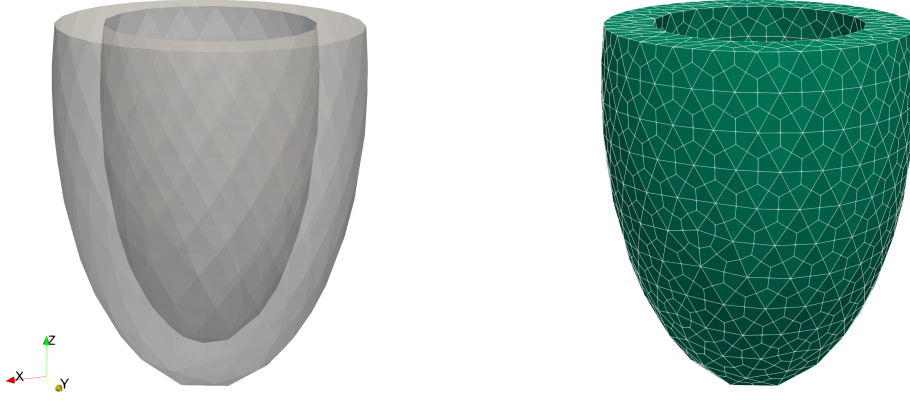


Figure 2: Active contraction of a truncated ellipsoid. Reference geometry (left) and computational mesh (right) of a truncated ellipsoid.

In addition to the material parameters and the slope \tilde{p} of the pressure load, we consider the maximum value \tilde{T}_a of the active tension and the fiber orientation angles α_{epi} and α_{endo} at the outer and the inner surfaces, respectively. Hence, $\boldsymbol{\mu} = [b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K, C, \tilde{p}, \tilde{T}_a, \alpha_{epi}, \alpha_{endo}] \in \mathcal{P} \subset \mathbb{R}^{12}$, where the parameter space \mathcal{P} is reported in Table 2.

	b_f	b_s	b_n	b_{fs}	b_{fn}	b_{sn}	K	C	\tilde{p}	\tilde{T}_a	α_{epi}	α_{endo}
							[kPa]	[kPa]	[kPa]	[kPa]	[°]	[°]
l_i	6.6	1.65	1.65	3.3	3.3	1.65	40	1.5	14	49.5	-105.5	74.5
r_i	9.4	2.35	2.35	4.7	4.7	2.35	60	2.5	16	70.5	-74.5	105.5

Table 2: Active contraction of a truncated ellipsoid. Parameters' range.

The FOM is built on an hexahedral mesh with 6455 vertices using \mathbb{Q}_1 -FE, thus obtaining $N_h = 19365$, and a BDF1 scheme in time with $\Delta t = 0.005$ s and $T = 0.25$ s, such that $N_t = 50$. For the construction and evaluation of the reduced model, we sample 100 points in \mathcal{P} using LHS and compute the corresponding high-fidelity displacements, which are then split (8:2) into training and test sets. In this case, the CPU time required to compute a single FOM solution is in average 6 min 11 s.

4.2 Comparison of global and tensor-decomposition POD-GPR approaches

4.2.1 Test 1: deformation of a beam

Reduced basis of different dimensions N can be constructed by performing POD on the FOM snapshots matrix \mathbf{S}_d using different truncation tolerances ε_{POD} , as reported in Table 3. For the case at hand, $N = 5$ RB functions (corresponding to a POD tolerance of $5 \cdot 10^{-4}$) are sufficient to obtain a good approximation of the high-fidelity FOM solutions, given the fast decay of the singular values.

ε_{POD}	10^{-3}	$5 \cdot 10^{-4}$	10^{-4}	$5 \cdot 10^{-5}$	10^{-5}
N	4	5	8	12	22

Table 3: Deformation of a beam. Dimension of RB subspaces for different POD tolerances.

Since the GPR models highly depend on the choice of the covariance functions, we analyze their predictive performances using different kernels, namely polynomial, RBF and ARD-RBF, and different scaling techniques, i.e., standardization and min-max normalization. For this comparison, we consider only the first $N_s = 50$ training samples collected offline for the construction of the RB matrix \mathbf{V} . We point out that, for the tensor-decomposition-based approach, time- and parameter-modes are trained as GPs after the truncated

SVD with tolerance 10^{-2} is computed, obtaining $3 \leq N_\ell \leq 6$. We present here the results subdivided in two parts: (i) first we address the GPRs predictions on the RB coefficients; (ii) then we assess the overall accuracy of the ROM with respect to the FOM.

Predictions of GPRs on RB coefficients. Figure 3 reports the MSEs computed using different covariance functions and scaling techniques, for both global and tensor-decomposition-based GPR models. In this case, the ARD-RBF kernel is more accurate than the other covariance functions, whereas the standardization and min-max normalization are almost comparable, with slightly better accuracy obtained using the former option. As a consequence, from now on we will focus only on ARD-RBF kernels and scale the dataset according to (7).

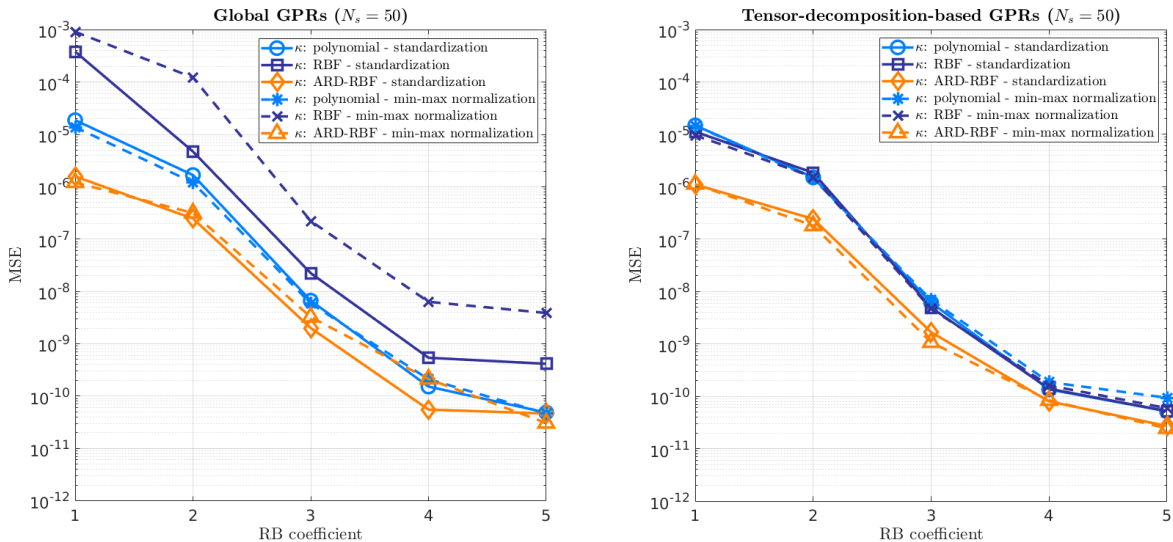


Figure 3: Deformation of a beam. Mean square error of global (left) and tensor-decomposition-based (right) POD-GPR ROMs computed using different kernel types and scaling techniques, for $N = 5$.

We analyze the performances of the global and the tensor-decomposition-based POD-GPR ROMs in terms of accuracy and efficiency by increasing the size of the training set from $N_s = 10$ to $N_s = 80$. The RSE computed over the testing set is reported in Figure 4, showing a good improvement of the approximation by both approaches as we increase the size of the training set from $N_s = 10$ to $N_s = 40$. As a larger number of data has to be processed for increasing values of N_s , the offline computational cost also becomes larger. However, we point out that the tensor-decomposition-based approach beats the global one in terms of computational efficiency during training, as this is performed for each GP on a much smaller dataset (see Remark 3).

In Table 4, we report the training time and the RSEs computed when $N_s = 50$. The true projection coefficients, the predictive means and their 95% confidence level computed for two different input parameters are shown in Figure 5 and 6 for the global and the tensor-decomposition-based GPR models, respectively. We observe that, in both cases, the errors between the POD-GPR ROMs approximation and the true solution are essentially bounded by the 95% confidence levels.

Accuracy of ROM w.r.t FOM. The reduced-order solutions $\mathbf{V}\hat{\mathbf{q}}(t^n; \cdot)$, for $n = 1, \dots, N_t$, are reconstructed online to validate the overall accuracy of the regression models with respect to the FOM. First of all, we compute the errors between the high-fidelity solution \mathbf{u}_h and its projection onto the RB space, i.e., $\mathbf{V}\mathbf{V}^T \mathbf{u}_h$, averaged over the whole test set, obtaining $\text{tAE} \approx 1.27 \cdot 10^{-5}$ and $\text{tRE} \approx 2.01 \cdot 10^{-3}$. These represent the best (linear) approximation errors in $\text{Col}(\mathbf{V})$, and thus the comparative values for our ROMs. From the results reported in Table 5, we observe good accuracy of the GPR models with respect to either the high-fidelity solutions as well as to the RB projection approximations. We recall that the tensor-decomposition-based approach beats the global one in terms of CPU time required during the offline training. For this test case the FOM requires in average 48 s to compute the solution dynamics given the parameter vector,

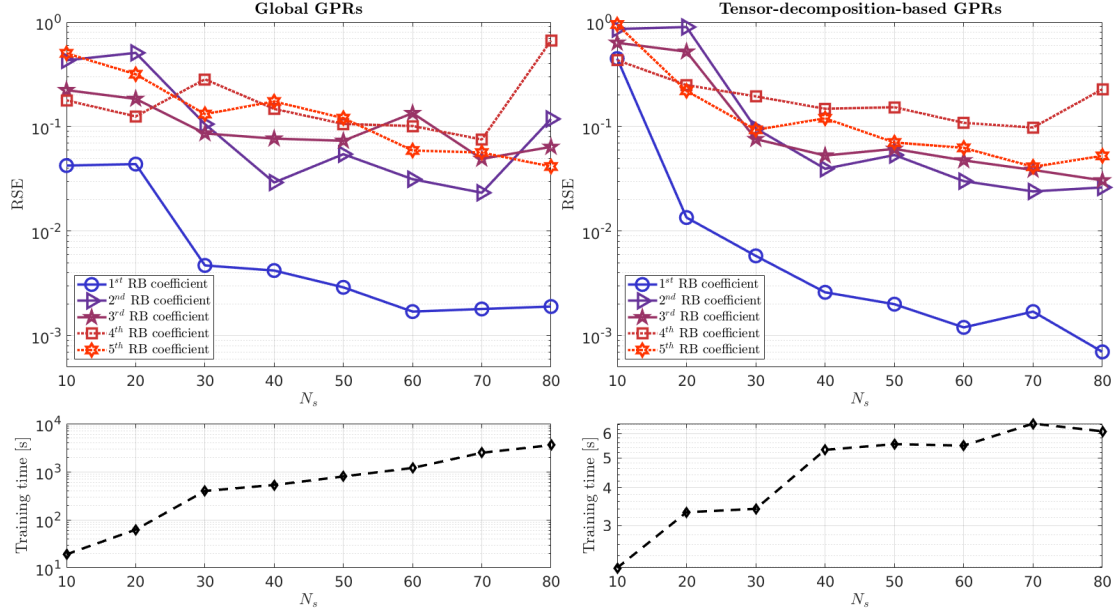


Figure 4: Deformation of a beam. Relative squared error of the global (top left) and the tensor-decomposition-based (top right) POD-GPR ROMs over the size of the training set. Corresponding training times (bottom) are also reported. These results have been obtained taking into account a testing set of $N_s^{test} = 20$ parameters, unseen during training.

	Global GPR	TD-based GPR
Training CPU time	881 s	6 s
RSE(q_1)	0.0029	0.0020
RSE(q_2)	0.0545	0.0535
RSE(q_3)	0.0732	0.0612
RSE(q_4)	0.1061	0.1532
RSE(q_5)	0.1206	0.0705

Table 4: Deformation of a beam. Computational data related to the global and the tensor-decomposition-based regression approaches with ARD-RBF kernels, for $N = 5$, $N_s = 50$ and $N_t = 50$.

whereas the global and the tensor-decomposition-based POD-GPR ROMs are able to reduce the online CPU time up to 364 and 270 times, respectively. Furthermore, we compare the accuracy and efficiency of standard POD-Galerkin ROMs equipped with the discrete empirical interpolation method (DEIM) [CS10] hyper-reduction technique. In particular, we take into account $|\mathcal{I}| = 83$ and $|\mathcal{I}| = 119$ DEIM interpolation points, corresponding to a POD tolerance equal to 10^{-4} and 10^{-5} on residual snapshots, respectively. The results are reported in Table 5.

Figure 7 shows the evolution in time of the mean (over the test set) absolute error for a different number of training samples N_s , whilst the corresponding computational speed-ups are reported in Figure 8. We observe that both approaches perform well and with comparable accuracy when the training dataset is sufficiently large. In particular, the tensor-decomposition-based approach achieves very high accuracy for $t < 0.05$ s. We believe that this is due to the fact that this method requires the solution to 1-dimensional regression tasks, which is less involved than the 2-dimensional regression task solved by the global approach. This is especially true for low time instances, as no data are available for $t < 0$ s. On the other hand, the global GPRs are able to achieve higher computational speed-ups than tensor-decomposition-based ones, although the difference between the two POD-GPR ROMs reduces as N_s increases.

The FOM displacement and the approximation error of the POD-GPR ROMs at different time instances

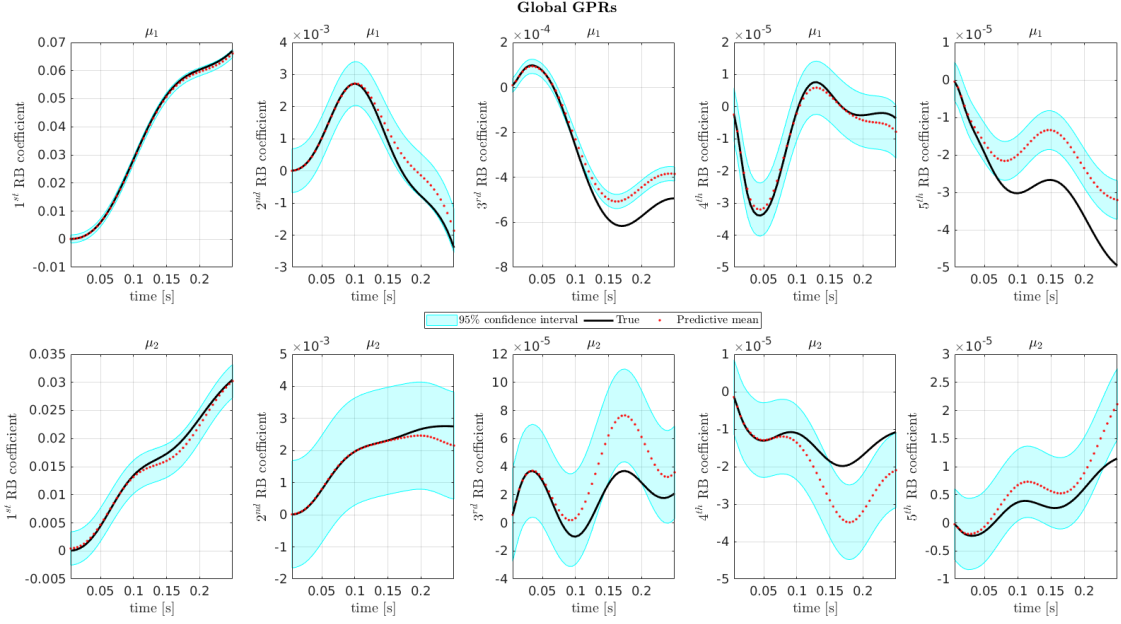


Figure 5: Deformation of a beam. Evolution over time of the exact RB coefficients (black) – from left to right – and the corresponding global POD-GPR ROM means (dotted red) for different testing parameters, from top to bottom. Moreover, we report the 95% confidence levels.

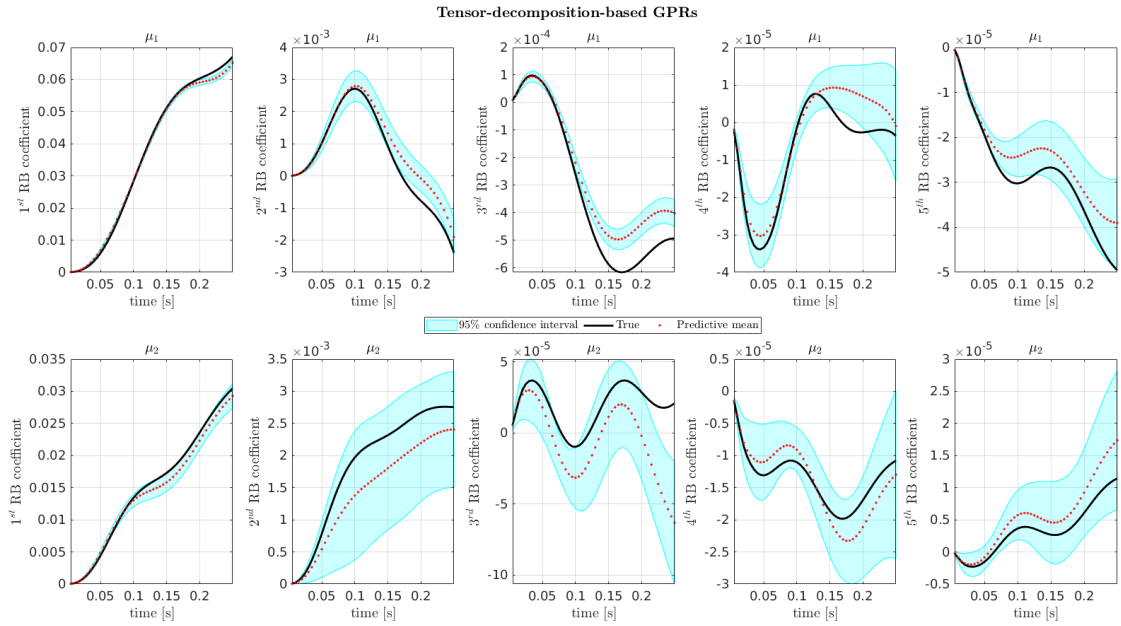


Figure 6: Deformation of a beam. Evolution over time of the exact RB coefficients (black) – from left to right – and the corresponding tensor-decomposition-based POD-GPR ROM means (dotted red) for different testing parameters, from top to bottom. Moreover, we report the 95% confidence levels.

are reported in Figure 9, for a given values of the input vector.

So far, we have always considered during the online stage the same time instances used for training the GPRs, since our main focus is on the influence of the parameters on the deformation of the body. However, GPRs can generalize to unseen values also for the time component, as shown by the computational

	POD-Galerkin-DEIM		Global GPR	TD-based GPR
	$ \mathcal{I} = 83$	$ \mathcal{I} = 119$		
Online CPU time	11.4 s	13.4 s	0.13 s	0.18 s
Speed-up	4.2	3.6	364	270
$\text{mean}_{\mu} \text{tAE}(\mu)$	$2.3 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$4.4 \cdot 10^{-4}$	$4.1 \cdot 10^{-4}$
$\text{mean}_{\mu} \text{tRE}(\mu)$	$1.6 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$2.1 \cdot 10^{-2}$

Table 5: Deformation of a beam. Efficiency and accuracy of POD-Galerkin-DEIM ROM, global POD-GPR ROM and tensor-decomposition-based POD-GPR ROM.

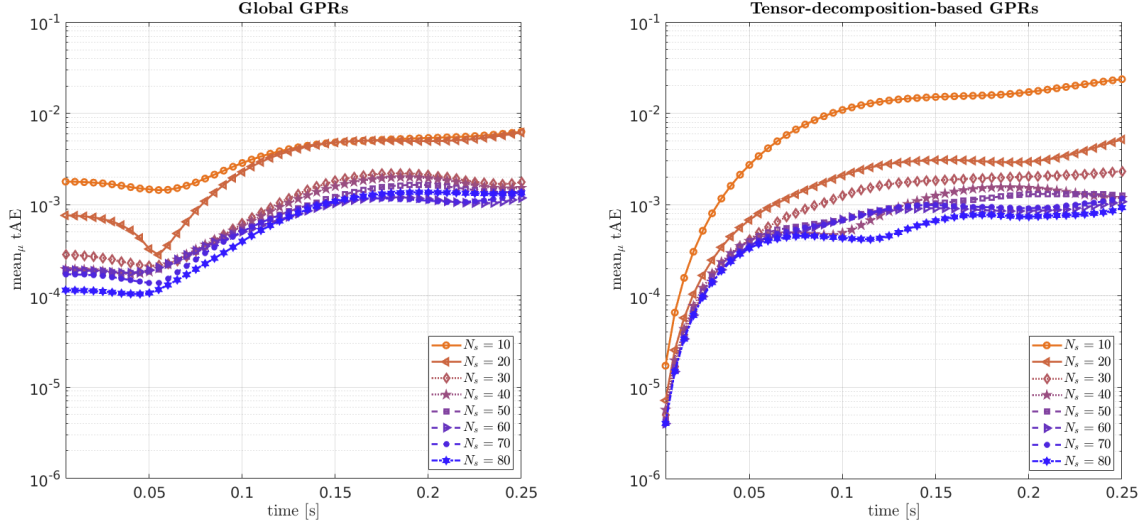


Figure 7: Deformation of a beam. Average absolute error of the global (left) and tensor-decomposition-based (right) POD-GPR ROMs approximations over the size of the training set.

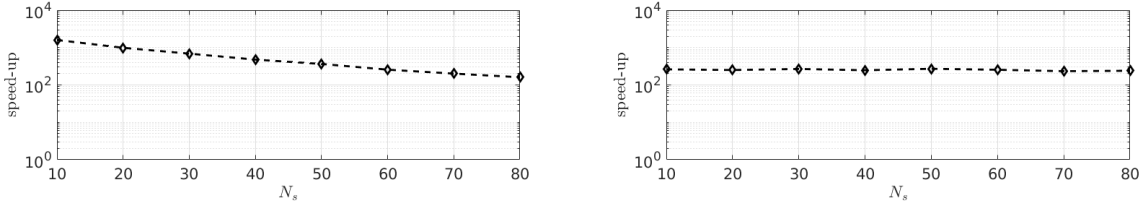


Figure 8: Deformation of a beam. Average computational speed-up of the global (left) and tensor-decomposition-based (right) POD-GPR ROMs approximations over time, varying the size of the training set, with respect to the full-order solution.

results reported in Table 6 when a time step size $\Delta t = 3.5 \cdot 10^{-3}$ s is used online. In this case, the total number of time steps is $N_t^{test} = 71$, and the FOM requires 75 s in average for a given parameter vector. The corresponding time-average L^2 -absolute error is reported in Figure 10 for different values of the input parameter. Within each plot we also report, in light gray color, the error computed using the same time step used during training, showing good prediction properties of the POD-GPR ROMs with respect both time and parameters.

4.2.2 Test 2: active contraction of a truncated ellipsoid

In this second test case, a larger number of RB functions is required to achieve the same POD accuracy with respect to the previous problem (see Table 7), given the slower decay of the singular values, thus increasing the complexity of the surrogate model due to the higher number of GPs that must be trained.

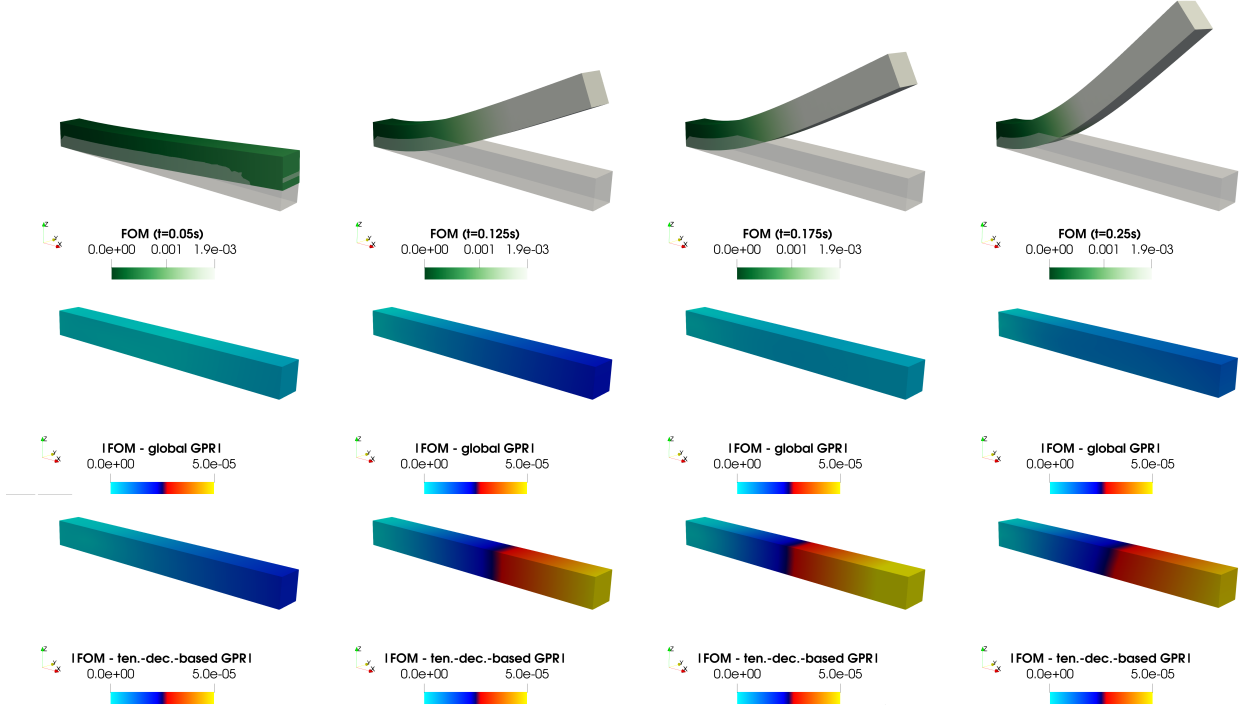


Figure 9: Deformation of a beam. FOM displacement (top), and approximation error of the global (middle) and tensor-decomposition-based (bottom) POD-GPR ROM, at different time instances, for $\mu = [4.20, 2.63, 2.51, 4.54, 4.46, 2.81, 27.25 \text{ kPa}, 2.51 \text{ kPa}, 0.0052 \text{ kPa}]$.

	Global GPR	TD-based GPR
Online CPU time	0.18 s	0.20 s
Speed-up	385	366
mean tAE μ	$1.1 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
mean tRE μ	$3.9 \cdot 10^{-1}$	$7 \cdot 10^{-2}$

Table 6: Deformation of a beam. Efficiency and accuracy of global and tensor-decomposition-based POD-GPR ROMs, computed over 20 testing parameters, using $\Delta t = 3.5 \cdot 10^{-3} \text{ s}$.

ϵ_{POD}	10^{-3}	$5 \cdot 10^{-4}$	10^{-4}	$5 \cdot 10^{-5}$	10^{-5}
N	20	30	71	100	202

Table 7: Active contraction of a truncated ellipsoid. Dimension of RB subspaces for different POD tolerances.

Predictions of GPRs on RB coefficients. Once fixed the RB dimension equal to $N = 30$, we analyze the fidelity of approximations computed using the global and the tensor-decomposition-based POD-GPR ROMs built using different covariance functions, trained on $N_s = 50$ parameter samples. Regarding the tensor-decomposition-based approach, time- and parameter-modes for each RB coefficient are obtained by computing the truncated SVD, obtaining $3 \leq N_\ell \leq 9$. Based on the observation of Section 4.2.1, we only rely on the standardization scaling of the training and test sets.

From the MSE reported in Figure 11, we observe that the choice of the kernel function has a smaller impact on the accuracy of the GPR predictions with respect to the previous test. However, since the ARD-RBF kernel shows better agreement to the true values of the first RB coefficient than the other kernels, we restrict ourselves to this choice for the GPs covariance function, unless otherwise specified.

Hence, we analyze the accuracy and the efficiency of the POD-GPR ROMs by increasing the size of the

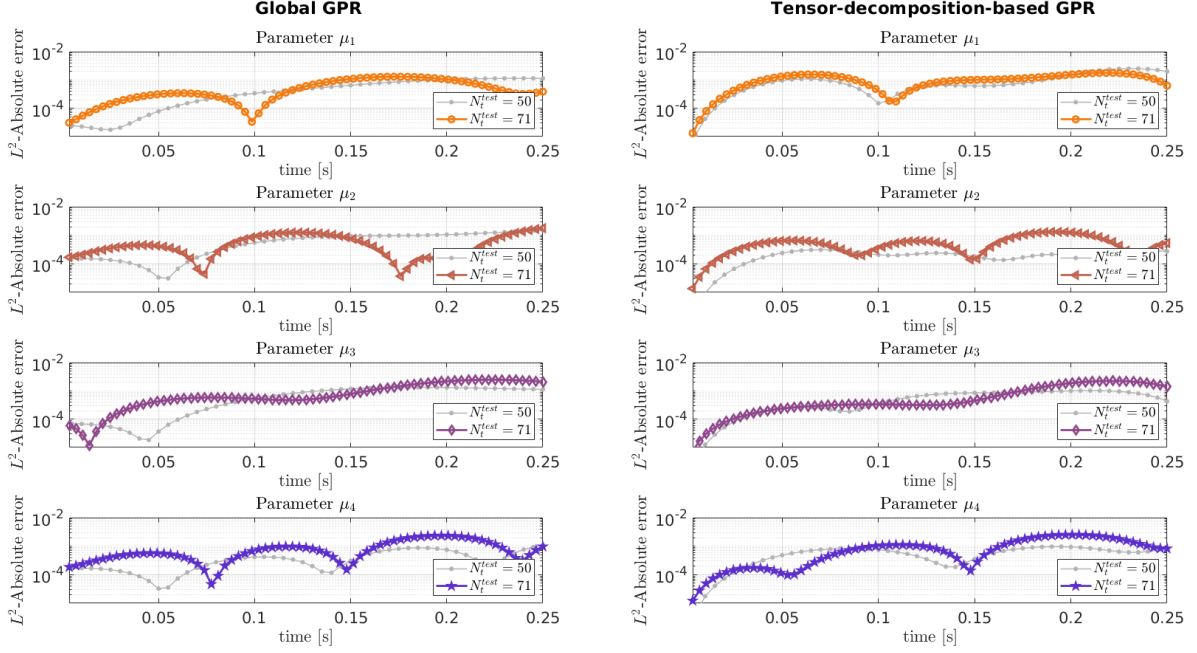


Figure 10: Deformation of a beam. Evolution of the time-average L^2 -absolute error of the global (left) and tensor-decomposition-based (right) POD-GPR ROMs, using $\Delta t = 3.5 \cdot 10^{-3}$ s.

training set from $N_s = 10$ up to 80. We point out that the global approach requires approximately 4 h to train the GPs on $N_s = 50$ sample parameters, probably due to the high FOM dimension N_h , in addition to the large number of RB coefficients. To reduce this burden without restoring to parallel computing, we restrict from now on the size of the training set by keeping only the data associated with time instant t^{5i} , for $i = 1, \dots, \lfloor N_t/5 \rfloor$. From Figure 12, we observe that a dataset composed of $N_s = 40$ or $N_s = 50$ sampled parameters represents a good trade-off between accuracy and CPU training times.

For the RB coefficients q_1, \dots, q_5 , we report in Table 8 the computational data obtained with $N_s = 50$, whilst the corresponding true projection coefficients, predictive means and 95% confidence level are depicted in Figures 13 and 14. The two approaches are almost comparable in terms of accuracy, whilst the tensor-decomposition-based approach shows faster training times.

	Global GPR	TD-based GPR
Training CPU time	501 s	28 s
RSE(q_1)	0.1487	0.0753
RSE(q_2)	0.2517	0.1384
RSE(q_3)	0.0092	0.0103
RSE(q_4)	0.2331	0.1369
RSE(q_5)	0.2501	0.5711

Table 8: Active contraction of a truncated ellipsoid. Computational data related to the global and the tensor-decomposition-based regression approaches with ARD-RBF kernels, for $N = 30$, $N_s = 50$ and $N_t = 50$.

Accuracy of ROM w.r.t. FOM. Finally, we compare the accuracy of the surrogate GPR models on the approximation of the whole high-fidelity displacement field, that is how $\mathbf{V}\hat{\mathbf{q}}(\cdot; \boldsymbol{\mu})$ differs from $\mathbf{u}_h(\cdot; \boldsymbol{\mu})$, as well as its computational efficiency. We recall that the FOM requires 6 min 11 s in average to compute the solution dynamics for a given parameter. The results obtained are reported in Figure 15 and 16, and summarized in Table 9. As already observed, the speed-up achieved online by the global POD-GPR ROMs

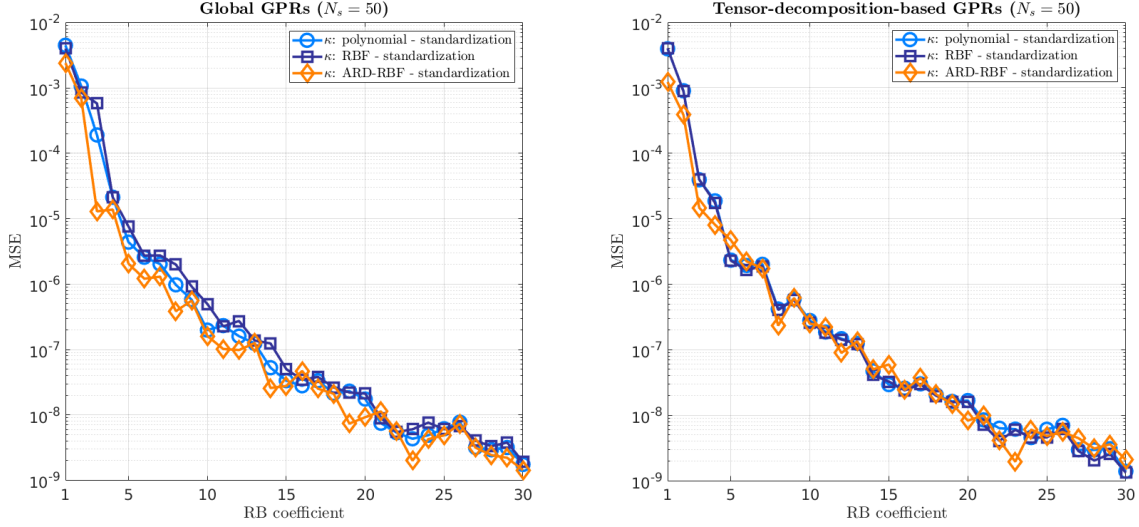


Figure 11: Active contraction of a truncated ellipsoid. Mean square error of global (left) and tensor-decomposition-based (right) POD-GPR ROMs computed using different kernel types, for $N = 30$.

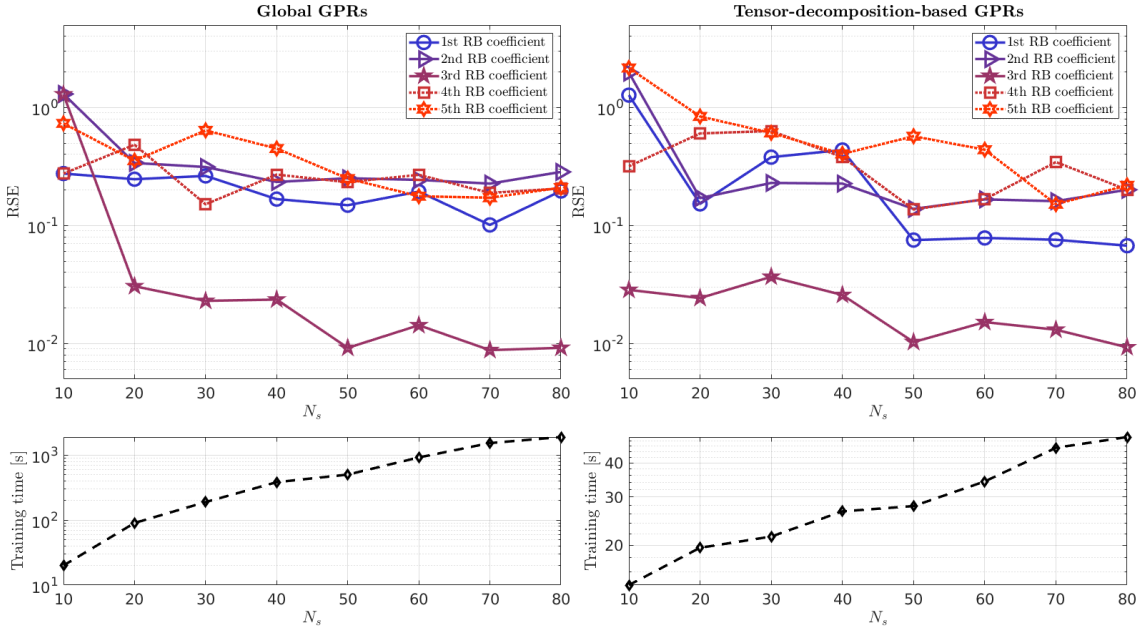


Figure 12: Active contraction of a truncated ellipsoid. Relative squared error – associated with the first 5 RB coefficients – of the global (top left) and the tensor-decomposition-based (top right) POD-GPR ROMs over the size of the training set. Corresponding training times (bottom) are also reported. These results have been obtained taking into account a testing set of $N_s^{test} = 20$ parameters, unseen during training.

is the highest between the two approaches. Figure 17 shows the FOM displacement and the approximation error of the POD-GPR ROMs at different time instances, for a given values of the input vector.

5 Multi-query problems in UQ using POD-GPR ROMs

Mathematical models describing physical phenomena depend on a (large) set of parameters that characterize different physical responses, and whose knowledge is severely limited due difficulties in performing measure-

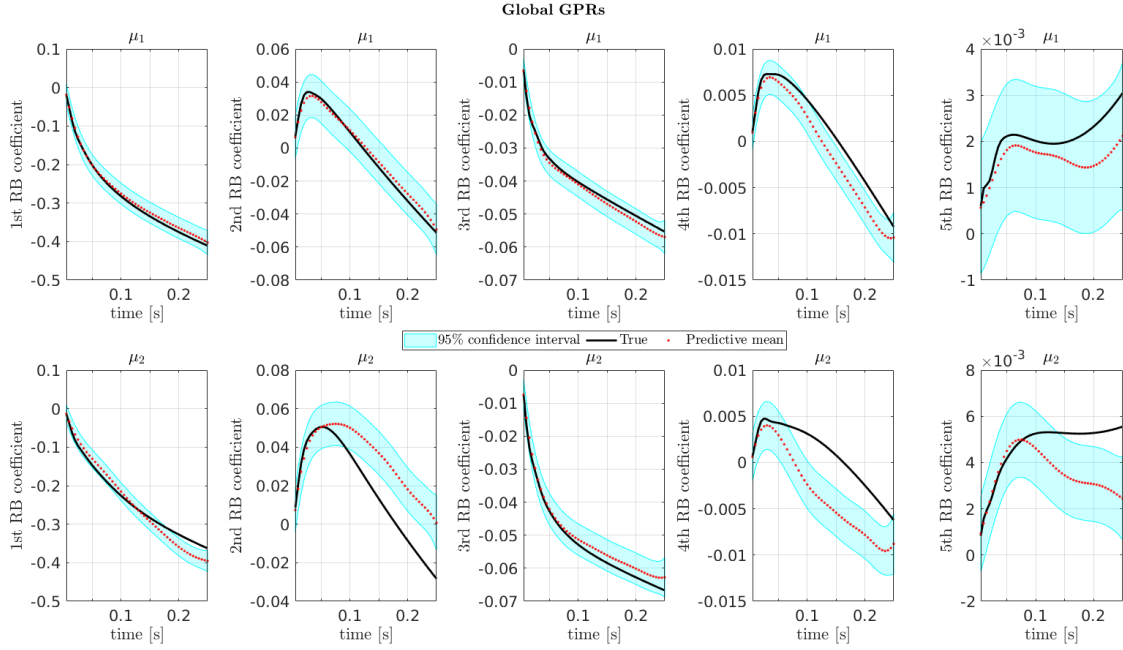


Figure 13: Active contraction of a truncated ellipsoid. Evolution over time of the exact RB coefficients (black) – from left to right – and the corresponding global POD-GPR ROM means (dotted red) for different testing parameters, from top to bottom. Moreover, we report the 95% confidence levels.

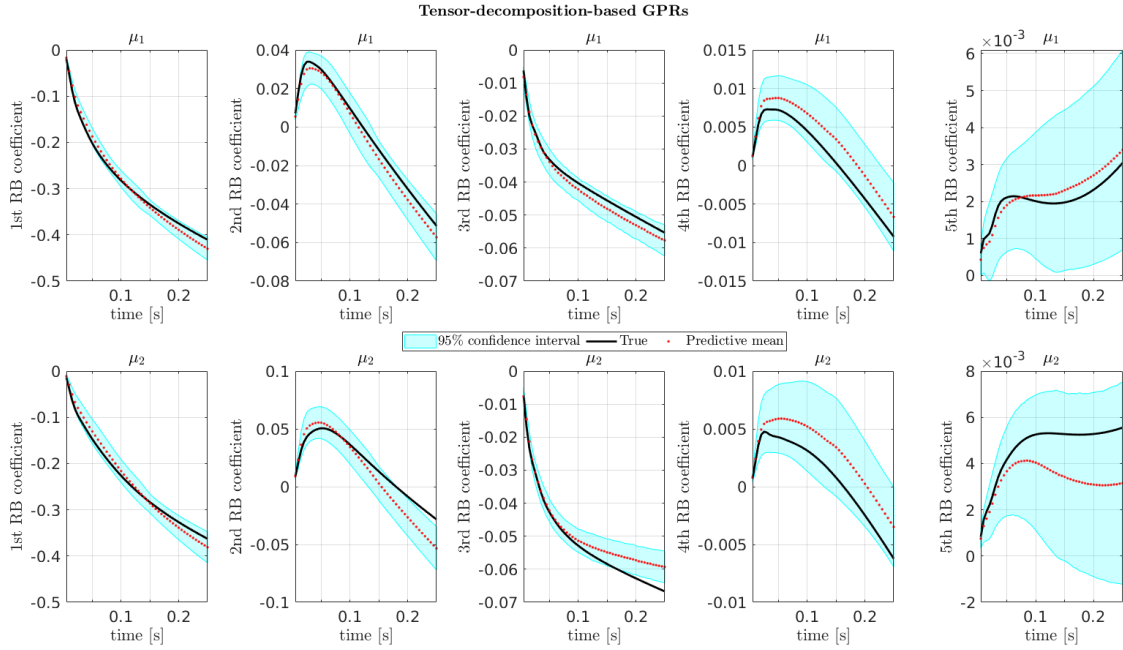


Figure 14: Active contraction of a truncated ellipsoid. Evolution over time of the exact RB coefficients (black) – from left to right – and the corresponding tensor-decomposition-based POD-GPR ROM means (dotted red) for different testing parameters, from top to bottom. Moreover, we report the 95% confidence levels.

ments. However, only a smaller subset of inputs have a great impact on specific outputs quantities, that are scalar-valued functions of the displacement field, for the application at hand. Hence, it is important to identify which factors should be properly measured and which others can be arbitrarily fixed without

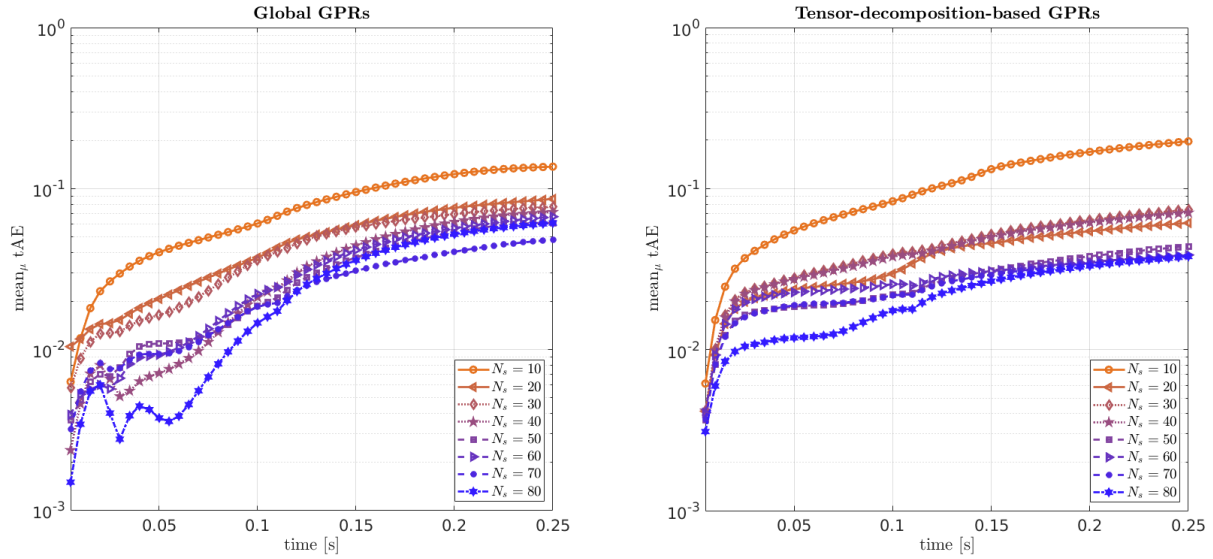


Figure 15: Active contraction of a truncated ellipsoid. Average absolute error of the global (left) and tensor-decomposition-based (right) POD-GPR ROMs approximations over time, varying the size of the training set.

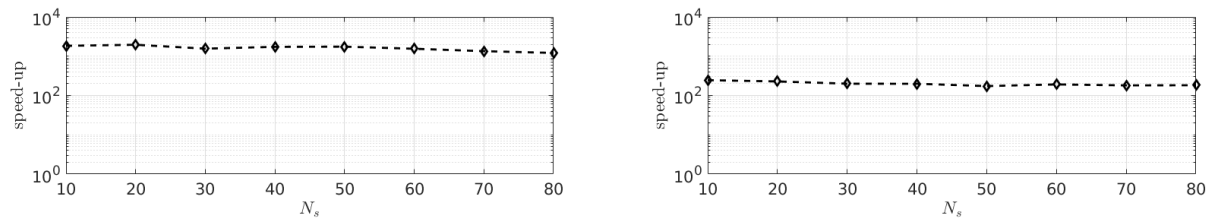


Figure 16: Active contraction of a truncated ellipsoid. Average computational speed-up of the global (left) and tensor-decomposition-based (right) POD-GPR ROMs approximations over the size of the training set, with respect to the full-order solution.

	Global GPR	TD-based GPR
Online CPU time	0.25 s	2.1 s
Speed-up	1487	174
$\text{mean}_{\mu} \text{ tAE}(\mu)$	$3.1 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$
$\text{mean}_{\mu} \text{ tRE}(\mu)$	$1.6 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$

Table 9: Active contraction of a truncated ellipsoid. Efficiency and accuracy of global POD-GPR ROM and tensor-decomposition-based POD-GPR ROM.

affecting the selected QoIs. Another key task when dealing with parameterized models is to infer about the unknown parameters based on observed data. We show how, using the mean of the posterior distribution of the GPRs to make predictions, we are able to efficiently address the solution to both types of problems, i.e., sensitivity analysis and parameter estimation, which require the repeated evaluation of the input-output map.

In the following, we address the solution to these multi-query problems on the numerical test cases reported in Sections 4.1.1 and 4.1.2. For the deforming beam, we consider as k -th QoI the displacement along the z -axis of a prescribed point with coordinates $\bar{\mathbf{X}} = (10^{-2}\text{m}, 5 \cdot 10^{-2}\text{m}, 5 \cdot 10^{-2}\text{m})$ in the reference

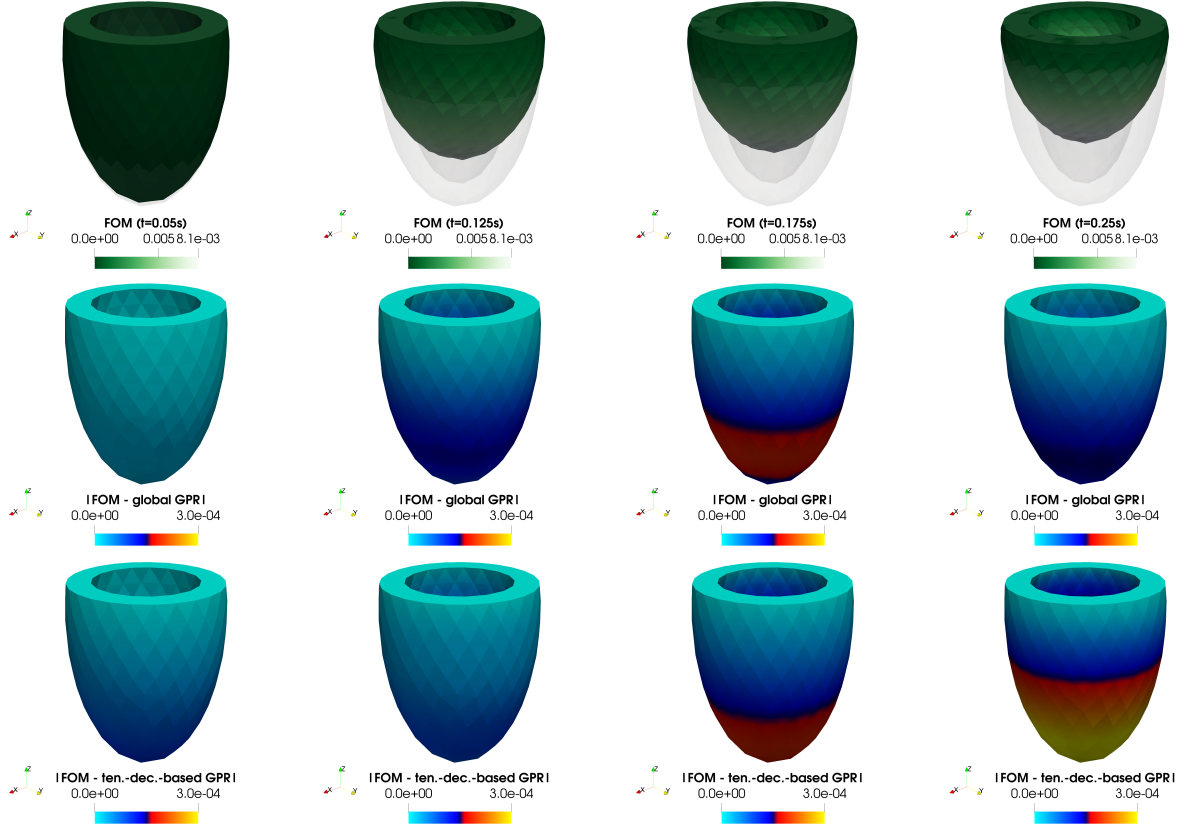


Figure 17: Active contraction of a truncated ellipsoid. FOM displacement (top), and approximation error of the global (middle) and tensor-decomposition-based (bottom) POD-GPR ROM, at different time instances, for $\boldsymbol{\mu} = [7.85, 1.81, 2.21, 4.11, 4.29, 1.76, 51.30 \text{ kPa}, 1.92 \text{ kPa}, 64.94 \text{ kPa}, 14.73 \text{ kPa}, -98.53^\circ, 104.72^\circ]$.

domain Ω_0 , measured at time $t = j_k \Delta t$, that is

$$y_k(\boldsymbol{\mu}) = \mathbf{u}(\bar{\mathbf{X}}, j_k \Delta t; \boldsymbol{\mu}) \cdot \mathbf{e}_z,$$

for $j_k \in \{10, 30, 50\}$. In the case of the active contraction of a truncated ellipsoid, we take into account the horizontal and vertical displacements of different points located on the outer and inner surfaces and reposted in Figure 18, acquired at different time instances, such that the outputs are given by

$$y_k(\boldsymbol{\mu}) = \mathbf{u}(\bar{\mathbf{X}}_{i_k}, j_k \Delta t; \boldsymbol{\mu}) \cdot \mathbf{e}_{\eta_k},$$

for $i_k \in \{1, \dots, 8\}$, $j_k \in \{10, 30, 50\}$, and $\eta_k \in \{x, y, z\}$. The index k spans through all possible combinations of the triplet (i, j, η) . With an abuse of notation, we denote as $\boldsymbol{\mu}$ (or y) the parameter vector both as random variable (or model output random variable) and as its outcome. To perform the UQ studies, we employ the Dakota toolkit [ABD⁺20], exploiting its Python direct interface.

5.1 Sensitivity Analysis (SA)

First of all, we perform global SA, which allows us to rank the parameters according to their influence on the QoIs and thus gain useful information. To begin, we employ the screening method of Morris' elementary effects [Mor91], which provides a qualitative analysis relying only on a small number of numerical simulations. In this case, we compare the results obtained using the GPR models (that, we recall, are trained on a dataset built from $N_s = 50$ sampled parameters, and suitable scaled using the standardization technique) with the high-fidelity ones. Hence, we perform a variance-based global SA by computing Sobol' first order and total effects indices [Sob90], providing a more quantitative ranking criterion, although a much larger number

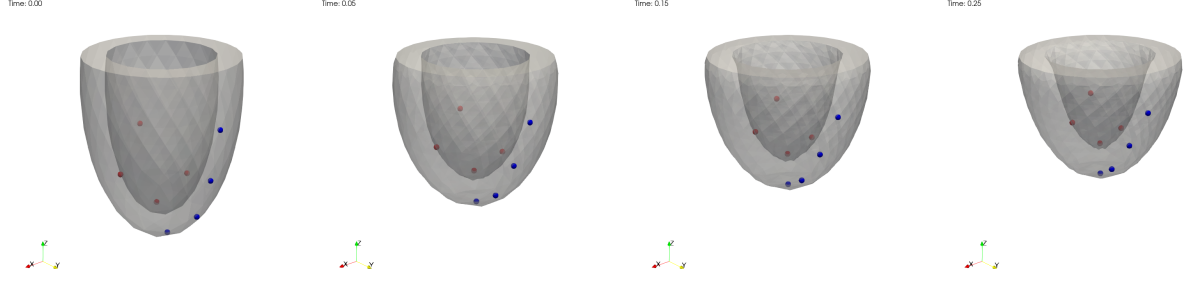


Figure 18: Active contraction of a truncated ellipsoid. Tissue displacement is measured at the outer (light red) and at the inner (dark red) surfaces, at three different time instances.

of input-output evaluations is required. Since multiple outputs are analyzed by conducting independent sensitivity studies, we restrict the formal introduction to a univariate model output.

5.1.1 Elementary Effects

The elementary effect test is a screening method able to overcome the main limitation of one-at-a-time designs and derivative-based methods, by considering wide ranges of variations for the inputs and averaging over a number of local derivative approximations, which allows to identify those input factors that are (i) negligible, (ii) linear and additive, or (iii) nonlinear or involved in interactions with other factors, with a relatively small number of model evaluations. Without loss of generality, we assume that the parameter domain is the p -dimensional unit hypercube. Hence, given a partition Ω_ι of $[0, 1]^p$ into $\iota \in \mathbb{N}$ levels, the elementary effect associated with the i -th input factor, for $i = 1, \dots, p$, is defined as

$$EE_i(\boldsymbol{\mu}) = \frac{y(\boldsymbol{\mu} + \mathbf{e}_i \Delta^\mu) - y(\boldsymbol{\mu})}{\Delta^\mu},$$

where $\boldsymbol{\mu}, \boldsymbol{\mu} + \mathbf{e}_i \Delta^\mu \in \Omega_\iota$, being \mathbf{e}_i the unit vector of all zeros but the i -th component, and Δ the step size with value in $\{1/(\iota - 1), \dots, (\iota - 2)/(\iota - 1)\}$. Since each elementary effect quantifies a local behavior, statistics of their distribution are estimated to obtain a global sensitivity measure. The mean and standard deviation of the distribution of EE_i is constructed by randomly sampling different $\boldsymbol{\mu}$ from Ω_ι . Moreover, to avoid cancellation effects, the mean of the distribution $|EE_i|$ of the absolute values is also computed. Thus, for $r \in \mathbb{N}$ trajectories, that is, sets of points $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_r$ sampled in Ω_ι , we obtain the following sensitivity measures for the i -th input

$$m_i = \frac{1}{r} \sum_{j=1}^r EE_i(\boldsymbol{\mu}_j), \quad m_i^* = \frac{1}{r} \sum_{j=1}^r |EE_i(\boldsymbol{\mu}_j)|, \quad sd_i = \frac{1}{r-1} \sum_{j=1}^r (EE_i(\boldsymbol{\mu}_j) - m_i)^2, \quad i = 1, \dots, p.$$

In our setting, we fix the number of levels $\iota = 6$ and choose $\Delta^\mu = \frac{\iota}{2(\iota-1)}$ to guarantee an equal probability sampling. Moreover, we consider $r = 20$ trajectories, corresponding to $r(p + 1)$ input-output evaluations, yielding a good balance between reliability of the results and computational resources required for the analysis.

Both m_i and m_i^* can be used to quantify the individual influence of the i -th input factor on the output y , whereas sd_i estimates the ensemble of its effects due to nonlinearities and interactions with the other factors. Low values for sd_i can be interpreted as the fact that the effect of the i -th parameter is independent of the point in which EE_i is evaluated, and thus of the values taken by the other inputs.

Test 1: deformation of a beam. For the first problem, we report in Figure 19 (top panel) the estimated mean $\mathbf{m}^* = [m_1^*, \dots, m_p^*]^T$ and standard deviation $\mathbf{sd} = [sd_1, \dots, sd_p]^T$ of the elementary effects of each QoI computed using the FOM, whereas on the middle and bottom panels we show the same statistical quantities computed using the POD-GPR ROMs outputs. Further details are reported in Table 10.

The elementary effects computed by means of the global and the tensor-decomposition-based approaches are overall in agreement with the high-fidelity ones. In particular, the most influential input parameters are always correctly identified. Relying on surrogates models allows to obtain accurate results of the SA in

seconds, rather than hours, and thus reduce the size of the parameter space by identifying the less influential inputs.

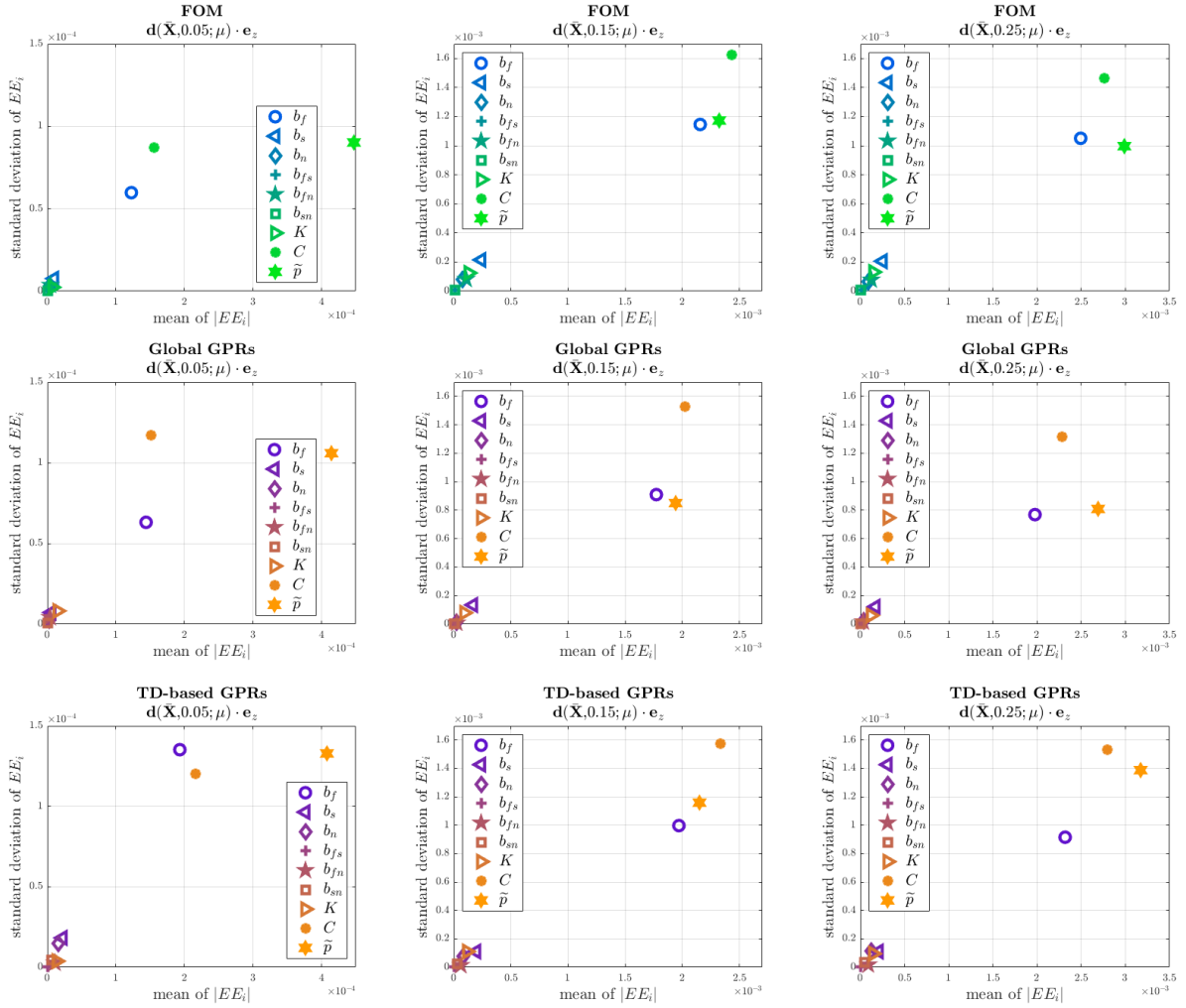


Figure 19: Deformation of a beam. Morris metrics computed using the outputs of the FOM (top), the global POD-GPR ROM (middle), and the tensor-decomposition-based POD-GPR ROM (bottom). The QoIs are given by the displacement along the z -axis of a point P , at three different time instants (from left to right, $t = 0.05, 0.15, 0.25$).

	FOM	Global GPR	TD-based GPR
Online CPU time	3 h 11 min	18 s	25 s
Speed-up	—	645	450
mean of $ EE_1 $ (b_f) at $t = 0.25$ s	$2.5 \cdot 10^3$	$2.0 \cdot 10^3$	$2.3 \cdot 10^3$
mean of $ EE_8 $ (C) at $t = 0.25$ s	$2.8 \cdot 10^3$	$2.3 \cdot 10^3$	$2.8 \cdot 10^3$
mean of $ EE_9 $ (\tilde{p}) at $t = 0.25$ s	$3.0 \cdot 10^3$	$2.7 \cdot 10^3$	$3.2 \cdot 10^3$

Table 10: Deformation of a beam. Computational performance of the POD-GPR-ROM methods for the computation of the elementary effects.

Test 2: active contraction of a truncated ellipsoid. Due to the higher complexity of this benchmark, we compute the elementary effects relying only on the POD-GPR ROMs. In particular, the global approach requires 18 s to run the analysis, whereas the tensor-decomposition-based approach takes around 6 min, almost 100% of the time required to compute $r(p + 1) = 260$ input-output evaluations.

In Figures 21 and 22, we report the scatter plots of Morris indices computed using the POD-GPR ROMs for different QoIs associated with two points located on the outer surface and highlighted in Figure 20. In particular, the outputs reported in Figures 21 show the vertical displacement at different time instances of a point P_1 located near the apex, whereas 22 focuses on the displacements along the x -axis of a point P_2 located between the apex and the base. The choice of the outputs of interest, differing in time, location or direction of the displacement, may have a great impact on the ranking of the input parameters, so that they have to be carefully chosen. However, by relying on POD-GPR approaches, one does not need to define the outputs in advance, since the training of the GPR is independent of the QoIs. Despite few discrepancies between the global and the tensor-decomposition-based approaches, we observe that the slope \tilde{T}_a of the active tension has large values for both the statistics, especially when taking into account the displacement along the z -axis. On the other hand, the fiber angle α_{epi} shows a stronger influence on QoIs associated with the horizontal displacement. In any case, the less influential parameters are correctly located by both POD-GPR ROMs, thus allowing to identify those inputs that can be arbitrarily fixed without affecting the variance of the output, which is the main purpose of screening methods.

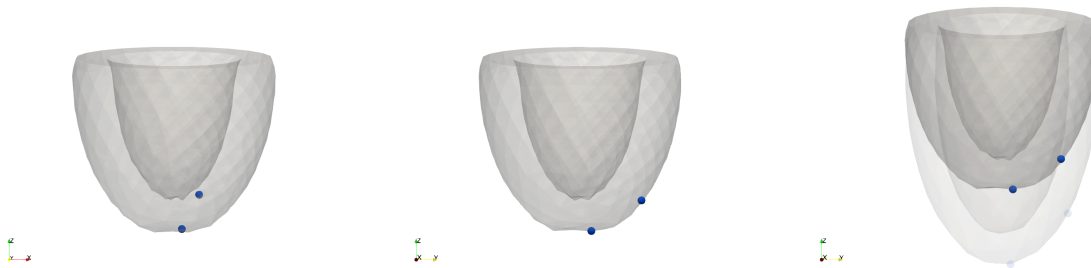


Figure 20: Active contraction of a truncated ellipsoid. Points P_1 and P_2 at time 0.25s from three different points of view.

5.1.2 Sobol' indices

Variance-based methods are even more powerful tool to quantify the relative importance of individual factors, and are based on a decomposition of the variance of the model output y into terms related to each input and to the interactions between them, that is

$$\text{Var}(y) = \sum_i V_i + \sum_{i < j} V_{ij} + \dots + V_{1\dots p}, \quad (10)$$

where $V_i = \text{Var}(\mathbb{E}[y|\boldsymbol{\mu}^{(i)}])$, $V_{ij} = \text{Var}(\mathbb{E}[y|\boldsymbol{\mu}^{(i)}, \boldsymbol{\mu}^{(j)}]) - V_i - V_j$, and so on, being $\boldsymbol{\mu}^{(i)}$ the i -th component of the parameter vector, for $i = 1, \dots, p$. A primary measure of sensitivity is given by the *first Sobol' index*

$$S_i = \frac{V_i}{\text{Var}(y)} = \frac{\text{Var}(\mathbb{E}[y|\boldsymbol{\mu}^{(i)}])}{\text{Var}(y)}, \quad i = 1, \dots, p,$$

which measures the effect of varying the i -th input alone averaged over variations in all input parameters: the higher is S_i , the greater is the influence of $\boldsymbol{\mu}^{(i)}$ on the output y . Note that $\sum_i S_i \leq 1$. However, it is often impossible to separate the effects of the inputs on the output and one should look for higher order interactions. Rather than performing a full sensitivity analysis, which requires the evaluation of $2^p - 1$ indices, we rely on the *total effect indices*

$$S_{T_i} = \frac{\mathbb{E}[\text{Var}(y|\boldsymbol{\mu}^{(\sim i)})]}{\text{Var}(y)} = \frac{\text{Var}(y) - \text{Var}(\mathbb{E}[y|\boldsymbol{\mu}^{(\sim i)}])}{\text{Var}(y)} = 1 - \frac{\text{Var}(\mathbb{E}[y|\boldsymbol{\mu}^{(\sim i)}])}{\text{Var}(y)},$$

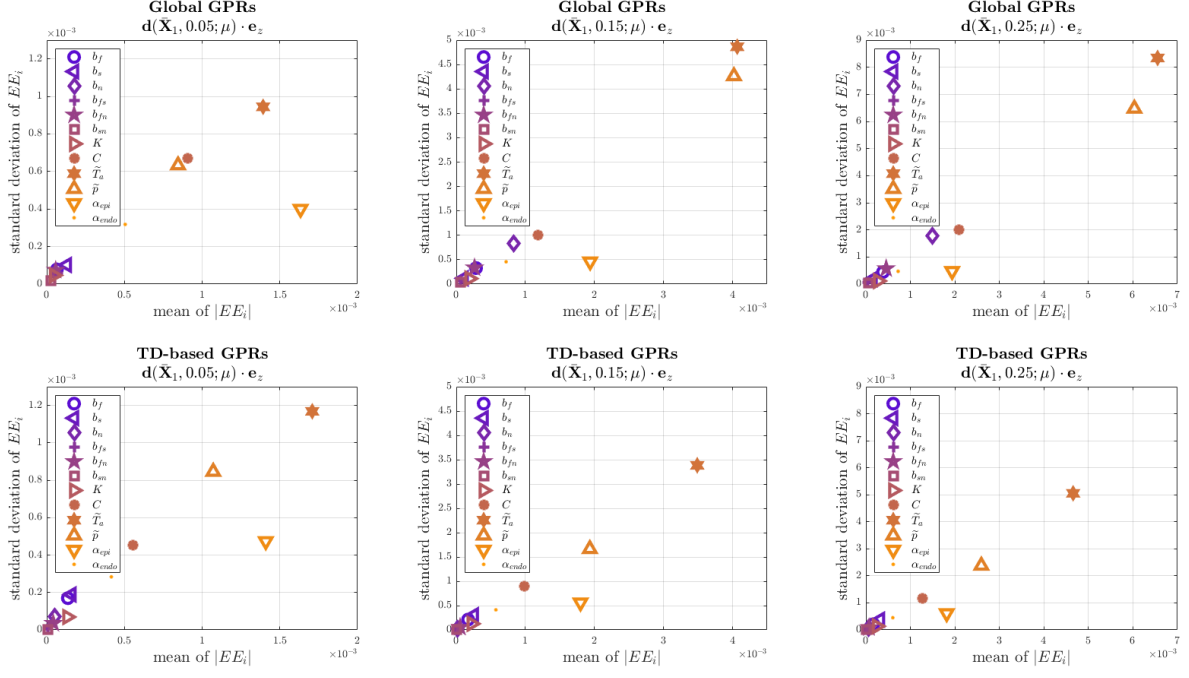


Figure 21: Active contraction of a truncated ellipsoid. Scatter plots of Morris metrics computed using the global POD-GPR ROM (top) and the tensor-decomposition-based POD-GPR ROM (bottom). The QoIs are given by the displacement along the z -axis of a point P_1 at the outer surface located near the apex, at three different time instants (from left to right, $t = 0.05, 0.15, 0.25$).

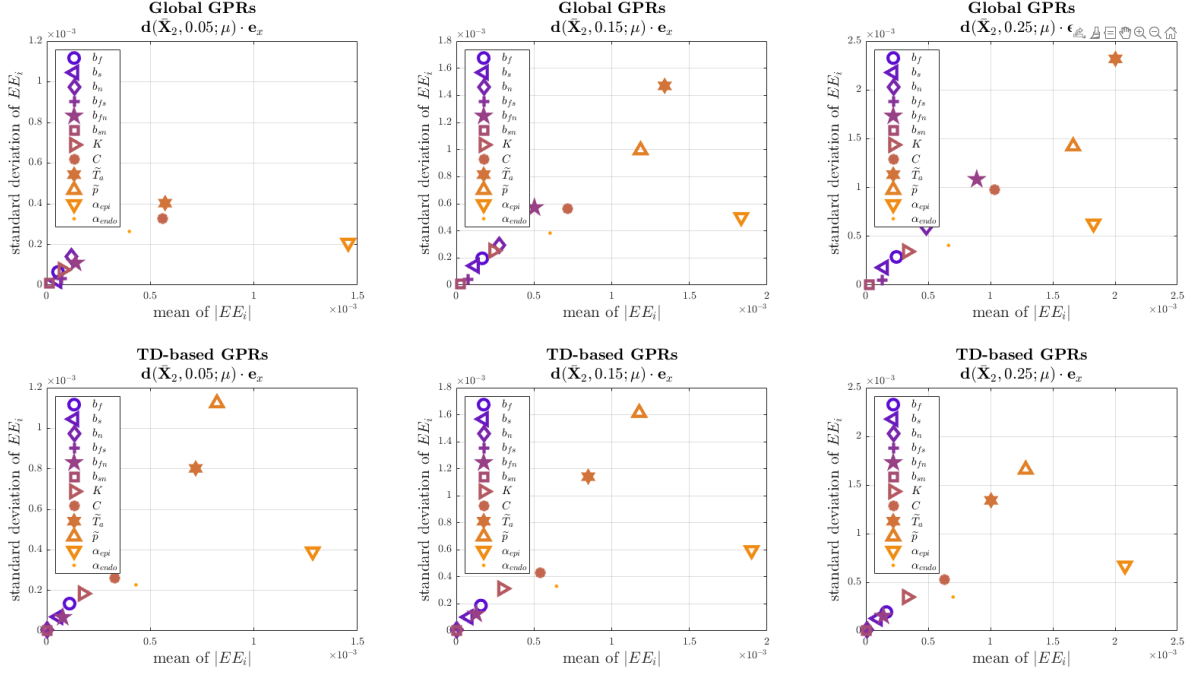


Figure 22: Active contraction of a truncated ellipsoid. Scatter plots of Morris metrics computed using the global POD-GPR ROM (top) and the tensor-decomposition-based POD-GPR ROM (bottom). The QoIs are given by the displacement along the x -axis of a point P_2 at the outer surface located between the apex and the base, at three different time instants (from left to right, $t = 0.05, 0.15, 0.25$).

for $i = 1, \dots, p$, where $\boldsymbol{\mu}^{(\sim i)} = (\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(i-1)}, \boldsymbol{\mu}^{(i+1)}, \dots, \boldsymbol{\mu}^{(k)})$ denotes the random vector of all input factors but $\boldsymbol{\mu}^{(i)}$. In this case, the smaller is S_{T_i} , the less influential is $\boldsymbol{\mu}^{(i)}$, which can thus be arbitrarily fixed within its range of uncertainty without appreciably affecting the output of interest. Unlike S_i , it holds $\sum_i S_{T_i} \geq 1$, since $S_{T_{i_1}}$ and $S_{T_{i_2}}$ ($i_1 < i_2$) take both into account the interactions between $\boldsymbol{\mu}^{(i_1)}$ and $\boldsymbol{\mu}^{(i_2)}$.

In this work, we employ the *Saltelli method* [Sal02], which is one of the most used and efficient procedure to evaluate Sobol' indices relying on $N_{samples}$ evaluations of the output quantity of interest, yielding a total number of $N_{samples}(p+2)$ output evaluations to compute both first Sobol' and total effect indices – therefore entailing a huge gain compared to a crude Monte Carlo estimate of the variance-based sensitivity indices.

Test 1: deformation of a beam We consider $N_{samples} = 200$ output evaluations, thus corresponding to $N_{samples}(p+2) = 2200$ input-output evaluations, and rely on the global and the tensor-decomposition-based GPR models. Figure 23 confirms the results of the SA previously conducted using Morris elementary effects, that is the scaling factor C , the slope \tilde{p} of the pressure load and the material stiffness in the fiber direction b_f are the most influential parameters for the test at hand, whereas little to no influence on the selected output is given by the other parameters, which can thus be fixed in their domain. Further details are reported in Table 11.

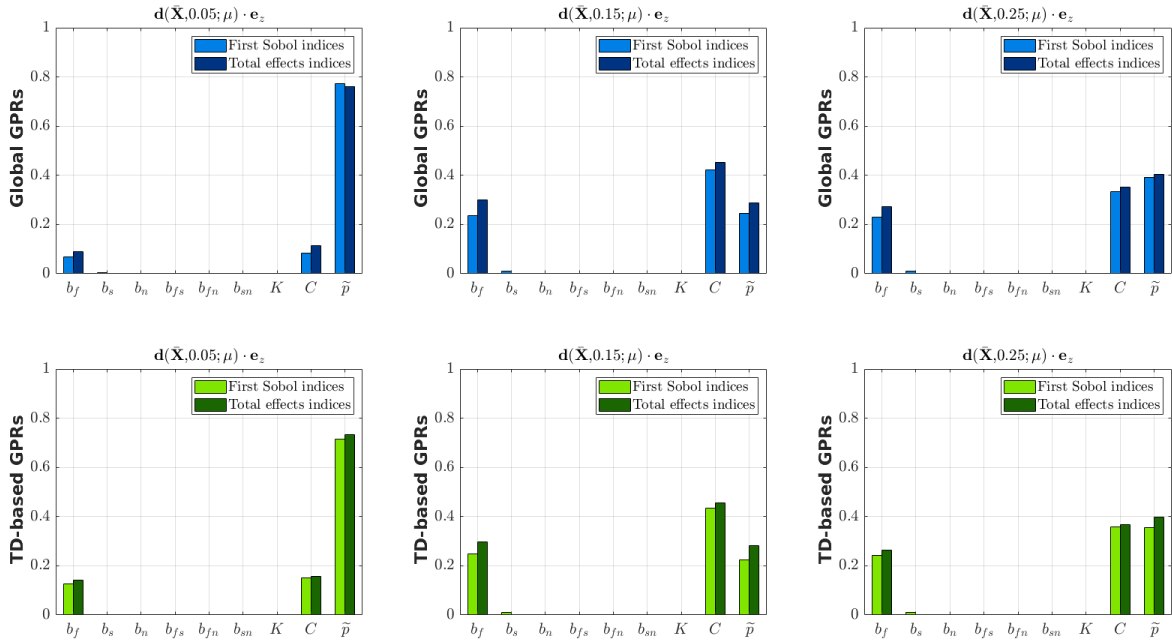


Figure 23: Deformation of a beam. Bar plot of Sobol' indices computed using the global (top) and the tensor-decomposition-based (bottom) POD-GPR ROMs. The QoIs are given by the displacement along the z -axis of a point P , at three different time instants.

Furthermore, we compute Sobol' indices for all $n = 1, \dots, N_t$. Figure 24 shows how the following generalization of Sobol' indices

$$\frac{\int_0^t \text{Var}(\mathbb{E}[y(\tau)|\boldsymbol{\mu}^{(i)}])d\tau}{\int_0^t \text{Var}(y(\tau))d\tau}, \quad \frac{\int_0^t \mathbb{E}[\text{Var}(y(\tau)|\boldsymbol{\mu}^{(\sim i)})]d\tau}{\int_0^t \text{Var}(y(\tau))d\tau},$$

evolve over time for the most effective inputs, i.e., b_f , C and \tilde{p} . We observe that the indices associated with the material inputs increase as time grows, whereas the pressure parameter reduces its influence on the output as the maximum slope is reached. However, when interested in, e.g., time-dependent model outputs, performing independent SAs on each prediction associated with a given time step t^n , with $n = 1, \dots, N_t$, may lead to unsatisfactory results. In fact, the N_t variables are correlated, so that important information cannot be extracted from separated analysis. In this scenario, one needs to rely on methods specific for SA on functional or multivariate outputs, such as [CMW06], where the output is expanded in a proper basis and

	FOM	Global GPR	TD-based GPR
Online CPU time	1 d 8 h 45 min	147 s	290 s
Speed-up	–	804	406
$S_1 (b_f)$ at $t = 0.25$ s	0.25	0.23	0.24
$S_8 (C)$ at $t = 0.25$ s	0.37	0.33	0.36
$S_9 (\tilde{p})$ at $t = 0.25$ s	0.35	0.39	0.36
$S_{T_1} (b_f)$ at $t = 0.25$ s	0.28	0.27	0.26
$S_{T_8} (C)$ at $t = 0.25$ s	0.38	0.35	0.37
$S_{T_9} (\tilde{p})$ at $t = 0.25$ s	0.38	0.41	0.40

Table 11: Deformation of a beam. Computational performance of the POD-GPR-ROM methods for the computation of Sobol’ indices.

standard SA methods are then applied to the coefficients of this expansion, and [GJKL14, LMM11, AGS20], where generalized Sobol’ sensitivity indices are proposed.

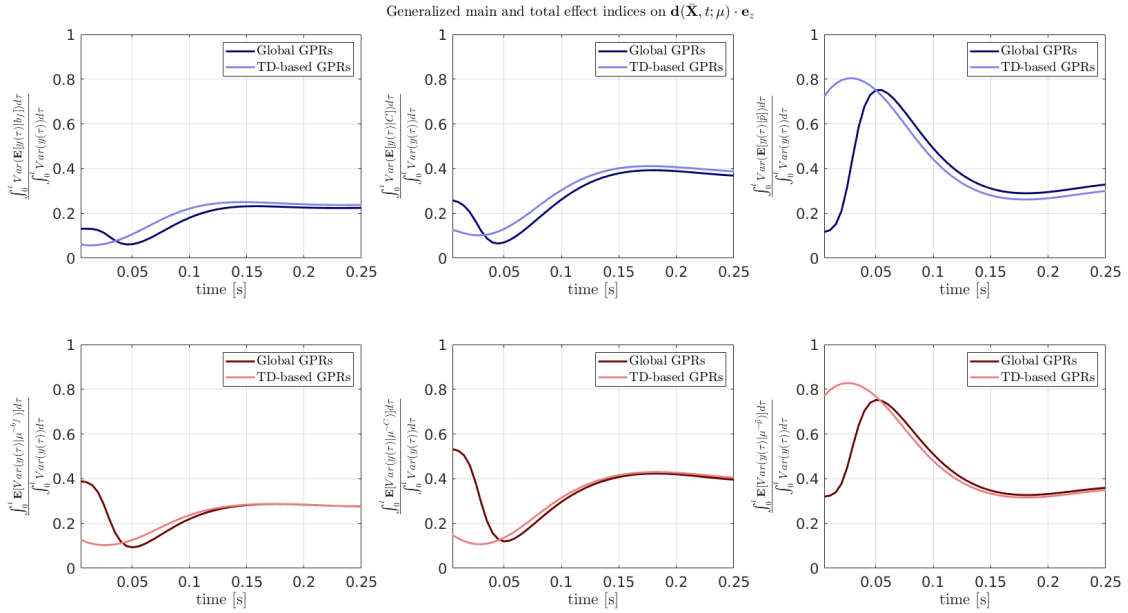


Figure 24: Deformation of a beam. Evolution in time of Sobol’ main (top) and total (bottom) effect indices for b_f , C and \tilde{p} computed using the global and the tensor-decomposition-based POD-GPR ROMs.

Test 2: active contraction of a truncated ellipsoid For the second benchmark, we consider again $N_{samples} = 200$, corresponding in this case to $N_{samples}(p + 2) = 2800$ runs of the POD-GPR ROMs. The results obtained with Sobol SA regarding the ranking of the input parameters, especially the detection of the less influential ones, is in agreement with Morris screening. Figures 25 and 26 show the sensitivity indices obtained for the QoIs related to the same points reported in Section 5.1.1, showing that \tilde{T}_a has a great influence, in particular on the displacement towards the base; moreover, it exhibits strong nonlinear interactions, as pointed out by the large value of the total effects at different times. Regarding the fiber angle on the outer surface α_{epi} , its impact is clearly visible on the horizontal displacement of the second point, which is located between the apex and the base. Almost all the parameters related to the material law have very small indices, meaning that they can be arbitrarily fixed within their range of variation without significantly affecting the computation of the outputs of interest.

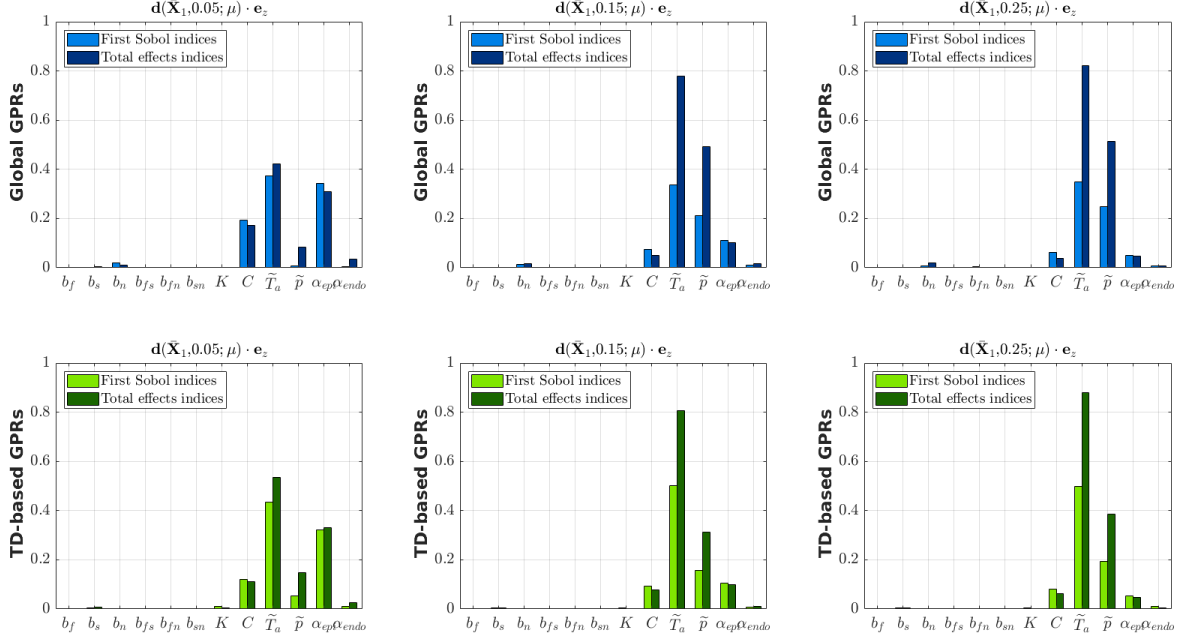


Figure 25: Active contraction of a truncated ellipsoid. Bar plot of Sobol' indices computed using the global (top) and the tensor-decomposition-based (bottom) POD-GPR ROMs. The QoIs are given by the displacement along the z -axis of a point P_1 at the outer surface located near the apex, at three different time instants.

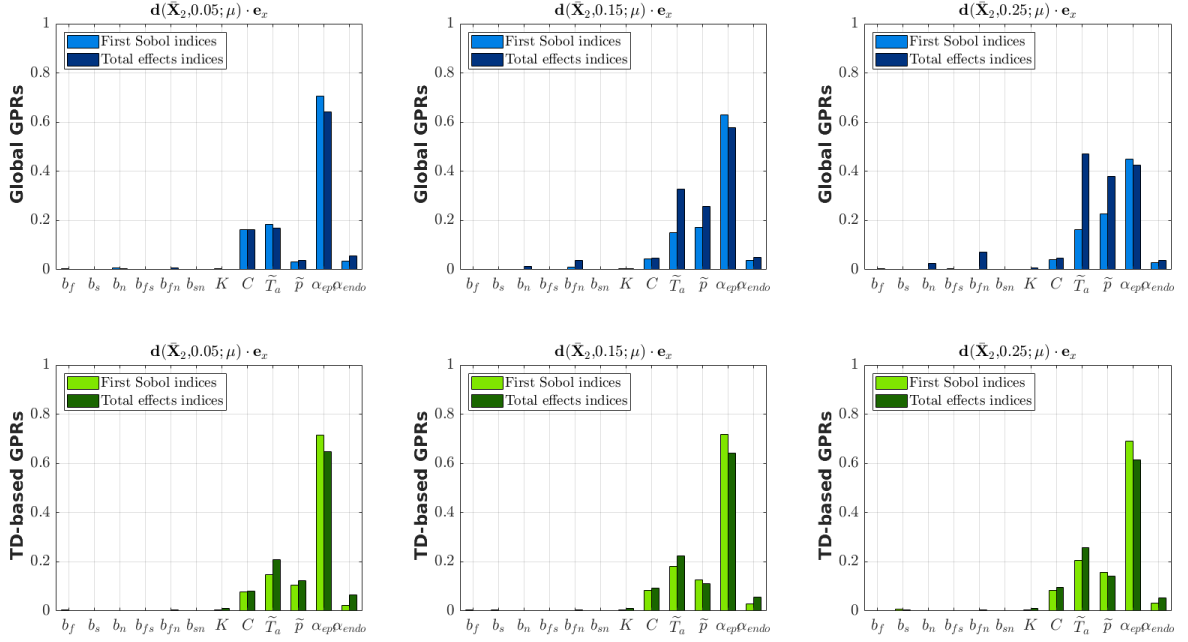


Figure 26: Active contraction of a truncated ellipsoid. Bar plot of Sobol' indices computed using the global (top) and the tensor-decomposition-based (bottom) POD-GPR ROMs. The QoIs are given by the displacement along the x -axis of a point P_2 at the outer surface located between the apex and the base, at three different time instants.

5.2 Parameter estimation

Finally, we address the solution of parameter estimation problems in a Bayesian framework, thus describing all inputs as random variables and seeking their probability density functions (PDFs), which, starting from a prior intuition, are updated as new data becomes available.

Given an observation \mathbf{y}_{obs} , corresponding to the realization of the output random variable \mathbf{y} , we want to infer the underlying probability distribution that produced the data. Note that, with respect to the previous section, we do not assume \mathbf{y} to be a univariate model output, hence the change in notation. In particular, we consider \mathbf{y}_{obs} to be the vector of QoIs evaluated on the FOM solution $\mathbf{u}_h(\boldsymbol{\mu}^*)$, computed for a specified parameter vector $\boldsymbol{\mu}^* \in \mathcal{P}$. To mimic experimental error, we consider an additive independent identically distributed Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$, so that $\mathbf{y}_{obs} = \mathbf{y}(\boldsymbol{\mu}^*) + \boldsymbol{\varepsilon}$. The *posterior* PDF of the random vector $\boldsymbol{\mu}$ can be obtained using Bayes' formula as

$$\pi(\boldsymbol{\mu}|\mathbf{y}_{obs}) \propto \pi(\mathbf{y}_{obs}|\boldsymbol{\mu})\pi_0(\boldsymbol{\mu}),$$

where

- $\pi(\mathbf{y}_{obs}|\boldsymbol{\mu})$ is the *likelihood*;
- $\pi_0(\boldsymbol{\mu})$ is the *prior* PDF of the inputs, which reflects any previous knowledge one might have about the parameters. If no previous knowledge is available, non-informative priors can be used.

Thanks to the assumption on the measurement error, the likelihood is given by $\pi(\mathbf{y}|\boldsymbol{\mu}) = \pi_{\boldsymbol{\varepsilon}}(\mathbf{y} - \mathbf{y}(\boldsymbol{\mu}))$, where $\pi_{\boldsymbol{\varepsilon}}$ is the PDF of the noise $\boldsymbol{\varepsilon}$. Finally, one can compute useful statistical indicators, such as the posterior mean and covariance, by repeatedly sampling from the posterior distribution.

Markov Chains Monte Carlo (MCMC) methods represent a well-known class of techniques used for sampling from a probability distribution. In this work, we rely on the Metropolis-Hastings algorithm, whose key idea is to generate a sequence of N_{MC} samples (with N_{MC} fixed, but sufficiently large) such that, at each iteration, the new candidate value is chosen according to a specified proposal distribution π_{prop} , and it is either accepted or rejected with some probability determined by the properties of the likelihood and the prior. To reduce the bias introduced by the choice of the initial sample, we perform *burn-in*, i.e., we discard $N_{burn-in}$ iterations at the beginning of an MCMC run; moreover, we keep only one sample every N_{thin} , so that the total number of chain iterations is given by $\lfloor (N_{MC} - N_{burn-in})/N_{thin} \rfloor$.

Test 1: deformation of a beam. Going back to the problem of the deformation of a beam discussed in Section 4.1.1, we employ the global and the tensor-decomposition-based POD-GPR ROMs built for $N_s = 50$, which are characterized by a time-average $L^2(\Omega_0)$ -absolute error of $4 \cdot 10^{-4}$ with respect to the FOM for the solution to the forward problem. The values of the target parameter $\boldsymbol{\mu}^*$ used to generate the observation \mathbf{y}_{obs} and those of the chain starting point $\boldsymbol{\mu}^0$ are reported in Table 12. The inputs to the Metropolis-Hastings algorithm are instead listed in Table 13.

	b_f	b_s	b_n	b_{fs}	b_{fn}	b_{sn}	K	C	\tilde{p}
							[kPa]	[kPa]	[kPa]
$(\cdot)^*$	6.2	1.2	2.8	5.8	5.8	2.8	27	1.2	0.0058
$(\cdot)^0$	8	2	2	4	4	2	50	2	0.004

Table 12: Deformation of a beam. Target and starting values of the Markov chain for the input parameters.

N_{MC}	10000	π_0	$\mathcal{U}[\mathcal{P}]$
$N_{burn-in}$	500	$\pi_{\boldsymbol{\varepsilon}}$	$\mathcal{N}(0, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I})$, with $\sigma_{\boldsymbol{\varepsilon}}^2 = 10^{-5}$ or 10^{-6}
N_{thin}	4	π_{prop}	$\mathcal{U}[\mathcal{P}]$

Table 13: Deformation of a beam. Inputs to the Metropolis-Hastings algorithm.

The samples of the MCMC chain obtained after the burn-in and the thinning operations are used to compute the posterior density functions reported in Figure 27 for $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} = 10^{-5}\mathbb{I}$ and $10^{-6}\mathbb{I}$, where $\mathbb{I} \in \mathbb{R}^{N_{QoI} \times N_{QoI}}$

denotes the identity matrix and $N_{QoI} = 3$. Moreover, the chain starting point $\boldsymbol{\mu}^0$ and the target parameter $\boldsymbol{\mu}^*$ are plotted as a black dashed line and a black solid line, respectively, for each parameter. We observe that, using the POD-GPR ROMs, we are able to correctly identify the parameter related to the material stiffness in the fiber direction b_f , the multiplicative constant C and the slope of the pressure \tilde{p} , whereas the other parameters are not captured. These results confirm those of the SA reported, e.g., in Figure 23, which identify b_f , C and \tilde{p} as the most influential inputs, while the others as non-influential. For what concerns the CPU time required for the generation of the MCMC chain, the global approach requires less than 12 min of computation, while the tensor-decomposition-based approach has a total run time around 23 min. Relying on the FOM would have required 8 d 17 h, showing how crucial is the need to rely on efficient reduced models in multi-query scenarios.

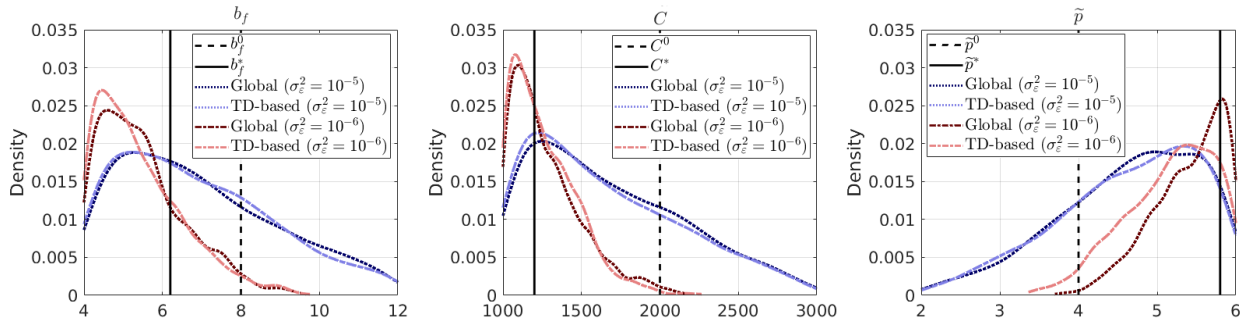


Figure 27: Deformation of a beam. Fitted posterior density function obtained from the MCMC samples of the most influential input parameters computed using the global and tensor-decomposition-based POD-GPR ROMs, for two different values of σ_ε^2 . The vertical lines show the chain starting point (dashed) and the target values (solid).

Remark 6. *Since the likelihood is defined by the input-output map and the level of noise, the choices of both the ROM and σ_ε^2 have an important influence on the posterior estimate. Nonetheless, the level of noise – chosen by taking into account the computed values for the QoIs – is predominant over the predictive errors between the two ROMs, and thus the posterior estimations perform similarly.*

Test 2: active contraction of a truncated ellipsoid. To conclude, we perform parameter estimation in the case of a contracting truncated ellipsoid. However, due to the high computational complexity of this benchmark test and the large number of numerical simulations required, we rely only on the global POD-GPR based model, which is characterized by fast CPU times during the online stage. We recall that, in this case, the time-average $L^2(\Omega_0)$ -absolute error is around $3 \cdot 10^{-2}$ with respect to the FOM.

The chain starting point and the target are reported in Table 14. Moreover, assuming some previous knowledge about the input values that have generated the observation, we reduce the support of the uniform distribution with respect to the whole parameter domain. The numerical setting used for the MCMC simulation is the same as in the previous test case reported in Table 13, apart from the number of chain samples, which is increased to $N_{MC} = 25000$, and the standard deviation of the Gaussian noise for which we now define with $\sigma_\varepsilon^2 = 10^{-7}$.

	b_f	b_s	b_n	b_{fs}	b_{fn}	b_{sn}	K	C	\tilde{p}	\tilde{T}_a	α_{epi}	α_{endo}
							[kPa]	[kPa]	[kPa]	[kPa]	[°]	[°]
$(\cdot)^*$	7.16	2.21	1.79	4.42	3.58	2.21	56	1.7	66.3	14.4	-99.3	80.7
$(\cdot)^0$	8	2	2	4	4	2	50	2	60	15	-90	90

Table 14: Active contraction of a truncated ellipsoid. Target and starting values of the Markov chain for the input parameters.

The PDF computed using the global GPR model, the chain starting point $\boldsymbol{\mu}^0$ and the target parameter $\boldsymbol{\mu}^*$ are reported in Figure 28 using a dark blue dotted line, a black dashed line and a black solid line,

respectively, for four different parameters. We observe that both the active tension \tilde{T}_a and the fiber angle at the outer surface α_{epi} are correctly inferred, as we might expect from the SA results, which identified these inputs to be overall the most influential ones. On the other hand, the posterior distribution of both b_f and b_s does not give any insight on which could be the most plausible observed value. Coherently, the identification of the remaining parameters is in general not good. For what concerns the CPU time required for the generation of the MCMC chain, the global approach requires 25 min of computation, while relying on the FOM would have required more than 100 days, becoming computationally unaffordable if the dimension N_h of the underlying problem is further increased.

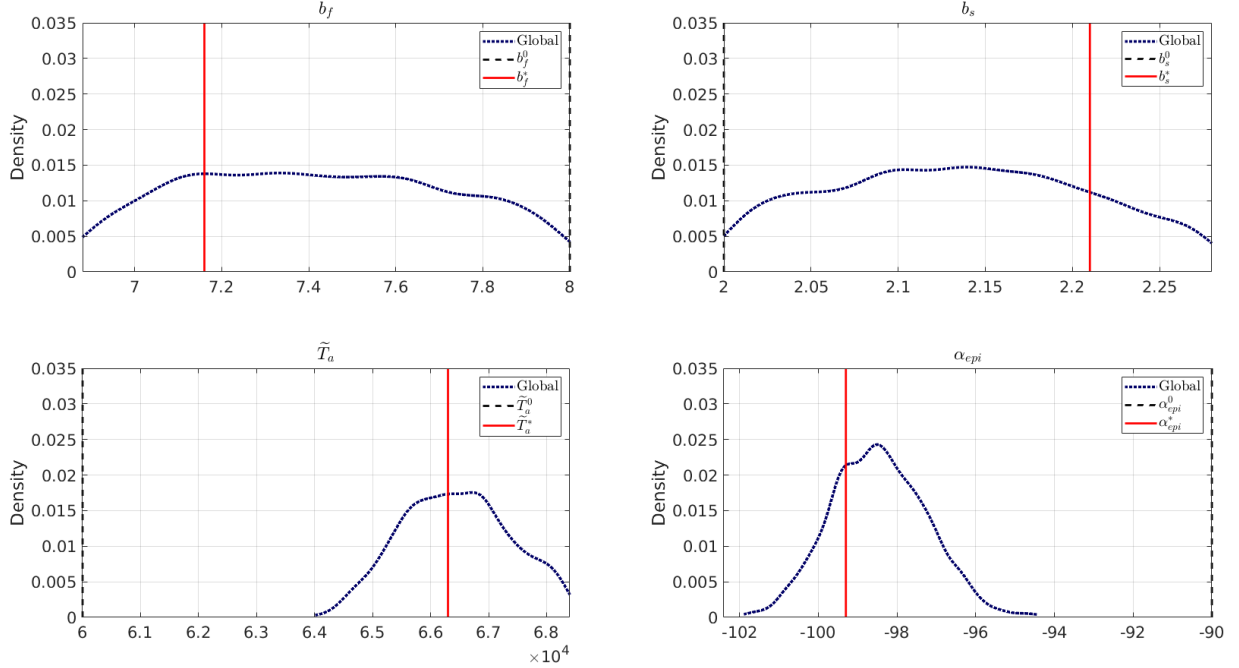


Figure 28: Active contraction of a truncated ellipsoid. Fitted posterior density function obtained from the MCMC samples of two non-influential input parameters (top) and two of the most influential ones (bottom) computed using the global POD-GPR ROM, for $\sigma_\varepsilon^2 = 10^{-7}$. The vertical lines shows the chain starting point (dashed) and the target values (solid).

6 Conclusions

In this work, we have addressed the solution to parameterized, nonlinear, time-dependent problems by means of POD-GPR ROMs, able to capture with sufficient accuracy the state solution dynamics at much lower computational cost with respect to high-fidelity FOMs. In particular, exploiting GP regression to approximate the POD projection coefficient allows to avoid intrusive hyper-reduction techniques. Another advantage of these methods is that, on one hand, they do not need to compute the whole solution dynamics when required only at specific points in space or time, and, on the other hand, one does not need to specify in advance the output quantities, being the training of the GPs independent of them, so that the QoIs can be chosen online according to the specific application.

Regarding the construction of the GPRs, we have compared a global approach and a tensor-decomposition-based one. Different kernels have been tested, as well as scaling techniques for the input/output data to the GPs, showing that a ARD-RBF covariance function and the standardization scaling are to be preferred. Finally, a convergence analysis with respect to the training set size has allowed to find a good trade-off between accuracy and online efficiency.

Both POD-GPR ROMs show low approximation errors with respect to the high-fidelity FOM, being able to accurately predict the system solution for new input values within the parameter space, as well as time instances within the given time range not used during training. Regarding the computational costs,

the tensor-decomposition-based approach is the most efficient during the offline stage, especially as the size of the problem increases. On the other hand, global POD-GPR ROMs shows higher online efficiency, and thus have to be preferred when hundreds or thousands input-output evaluations are required. Finally, the capabilities of the POD-GPR ROMs have been successfully tested on the solution to multi-query problems, namely global sensitivity analysis and parameter estimation.

Acknowledgment

LC and PZ acknowledge the partial support from Regione Lombardia project NEWMED under Grant No. POR FESR 2014–2020. AM, SF, PZ are members of Gruppo Nazionale per il Calcolo Scientifico (GNCS) of Istituto Nazionale di Alta Matematica (INdAM). AM and PZ acknowledge the partial support of the project ‘Sviluppo di sinergie fra Calcolo Scientifico e Machine Learning per applicazioni biomediche’ funded by GNCS. MG acknowledges the financial support from *Sectorplan Bèta* (the Netherlands) under the focus area *Mathematics of Computational Science*. The present research has been partially supported by FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence) and by MUR, grant Dipartimento di Eccellenza 2023-2027.

References

- [ABD⁺20] B Adams, W Bohnhoff, K Dalbey, M Ebeida, J Eddy, M Eldred, R Hooper, P Hough, K Hu, J Jakeman, M Khalil, K Maupin, JA Monschke, E Ridgway, A Rushdi, D Seidl, J Stephens, and J Winokur. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.13 user’s manual. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.
- [ADVN13] C Audouze, F De Vuyst, and PB Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.
- [AGS20] A Alexanderian, PA Gremaud, and RC Smith. Variance-based sensitivity analysis for time-dependent processes. *Reliability Engineering & System Safety*, 196:106722, 2020.
- [APLK19] F Alsayyari, Z Perkó, D Lathouwers, and JL Kloosterman. A nonintrusive reduced order modelling approach using proper orthogonal decomposition and locally adaptive sparse grids. *Journal of Computational Physics*, 399:108912, 2019.
- [AZF12] D Amsallem, MJ Zahr, and C Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.
- [BGB22] N Botteghi, M Guo, and C Brune. Deep kernel learning of dynamical models from high-dimensional noisy data. *Scientific Reports*, 12(1):21530, 2022.
- [BGW15] P Benner, S Gugercin, and K Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.
- [BHKE20] A Bērziņš, J Helmig, F Key, and S Elgeti. Standardized non-intrusive reduced order modeling using different regression models with application to complex flow problems. *arXiv preprint arXiv:2006.13706*, 2020.
- [BMNP04] M Barrault, Y Maday, NC Nguyen, and AT Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667–672, 2004.

- [BMQ17] D Bonomi, A Manzoni, and A Quarteroni. A matrix deim technique for model reduction of nonlinear parametrized problems in cardiac mechanics. *Computer Methods in Applied Mechanics and Engineering*, 324:300–326, 2017.
- [CFM22] L Cicci, S Fresca, and A Manzoni. Deep-hyROMnet: A deep learning-based operator approximation for hyper-reduction of nonlinear parametrized PDEs. *Journal of Scientific Computing*, 93(2):57, Oct 2022.
- [CFMQ22] L Cicci, S Fresca, A Manzoni, and A Quarteroni. Efficient approximation of cardiac mechanics through reduced order modeling with deep learning-based operator approximation. *arXiv preprint arXiv:2202.03904*, 2022.
- [CGMH23] P Conti, M Guo, A Manzoni, and JS Hesthaven. Multi-fidelity surrogate modeling using long short-term memory networks. *Computer Methods in Applied Mechanics and Engineering*, 404:115811, 2023.
- [CHJ⁺18] W Chen, JS Hesthaven, B Junqiang, Y Qiu, Z Yang, and Y Tihao. Greedy nonintrusive reduced order model for fluid dynamics. *AIAA Journal*, 56(12):4927–4943, 2018.
- [CLLZ20] K Cheng, Z Lu, C Ling, and S Zhou. Surrogate-assisted global sensitivity analysis: an overview. *Structural and Multidisciplinary Optimization*, 61(3):1187–1213, 2020.
- [CMW06] K Campbell, MD McKay, and BJ Williams. Sensitivity analysis when model outputs are functions. *Reliability Engineering & System Safety*, 91(10):1468–1472, 2006. The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004).
- [CMW15] T Cui, YM Marzouk, and KE Willcox. Data-driven model reduction for the bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.
- [CS10] S Chaturantabut and DC Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [DHO12] M Drohmann, B Haasdonk, and M Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing*, 34(2):A937–A969, 2012.
- [FDM21] S Fresca, L Dede’, and A Manzoni. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. *Journal of Scientific Computing*, 87(2):1–36, 2021.
- [FM22] S Fresca and A Manzoni. POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114181, 2022.
- [FMWvBW10] M Frangos, Y Marzouk, K Willcox, and B van Bloemen Waanders. Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems. *Large-Scale Inverse Problems and Quantification of Uncertainty*, pages 123–149, 2010.
- [FMZ23] N Franco, A Manzoni, and P Zunino. A deep learning approach to reduced order modelling of parameter dependent partial differential equations. *Mathematics of Computation*, 92(340):483–524, 2023.
- [GCM95] JM Guccione, KD Costa, and AD McCulloch. Finite element stress analysis of left ventricular mechanics in the beating dog heart. *Journal of biomechanics*, 28(10):1167–1177, 1995.
- [GH18] M Guo and JS Hesthaven. Reduced order modeling for nonlinear structural analysis using gaussian process regression. *Computer methods in applied mechanics and engineering*, 341:807–826, 2018.
- [GH19] M Guo and JS Hesthaven. Data-driven reduced order modeling for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 345:75–99, 2019.

- [GJKL14] F Gamboa, A Janon, T Klein, and A Lagnoux. Sensitivity analysis for multidimensional and functional outputs. *Electronic Journal of Statistics*, 8(1):575–603, 2014.
- [GLH⁺21] Z Gao, Q Liu, JS Hesthaven, BS Wang, WS Don, and X Wen. Non-intrusive reduced order modeling of convection dominated flows using artificial neural networks with application to rayleigh-taylor instability. *Commun. Comput. Phys.*, 30(1):97–123, 2021.
- [GMA⁺22] M Guo, A Manzoni, M Amendt, P Conti, and JS Hesthaven. Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities. *Computer methods in applied mechanics and engineering*, 389:114378, 2022.
- [GMW22] M Guo, SA McQuarrie, and K Willcox. Bayesian operator inference for data-driven reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 402:115336, 2022.
- [GPy12] GPy. GPy: A gaussian process framework in python. <http://github.com/SheffieldML/GPy>, since 2012.
- [GTS17] F Ghavamian, P Tiso, and A Simone. POD–DEIM model order reduction for strain-softening viscoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 317:458–479, 2017.
- [Guo21] M Guo. A brief note on understanding neural networks as gaussian processes. *arXiv preprint arXiv:2107.11892*, 2021.
- [HO03] GA Holzapfel and RW Ogden. *Biomechanics of Soft Tissue in Cardiovascular Systems*. CISM International Centre for Mechanical Sciences. Springer Vienna, 2003.
- [HPR22] JS Hesthaven, C Pagliantini, and G Rozza. Reduced basis methods for time-dependent problems. *Acta Numerica*, 31:265–345, 2022.
- [HRS16] JS Hesthaven, G Rozza, and B Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.
- [HU18] JS Hesthaven and S Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
- [KGH20] M Kast, M Guo, and JS Hesthaven. A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 364:112947, 2020.
- [LBN⁺17] J Lee, Y Bahri, R Novak, SS Schoenholz, J Pennington, and J Sohl-Dickstein. Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.
- [LGA⁺15] S Land, V Gurev, S Arens, CM Augustin, L Baron, R Blake, C Bradley, S Castro, A Crozier, M Favino, TE Fastl, T Fritz, H Gao, A Gizzi, BE Griffith, DE Hurtado, R Krause, X Luo, MP Nash, S Pezzuto, G Plank, S Rossi, D Ruprecht, G Seemann, NP Smith, J Sundnes, JJ Rice, NA Trayanova, D Wang, ZJ Wang, and SA Niederer. Verification of cardiac mechanics software: benchmark problems and solutions for testing active and passive material behaviour. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2184):20150641, 2015.
- [LGMS17] L Le Gratiet, S Marelli, and B Sudret. Metamodel-based sensitivity analysis: polynomial chaos expansions and gaussian processes. In *Handbook of uncertainty quantification*, pages 1289–1325. Springer, 2017.
- [LM14] J Li and YM Marzouk. Adaptive construction of surrogates for the bayesian solution of inverse problems. *SIAM Journal on Scientific Computing*, 36(3):A1163–A1186, 2014.
- [LMM11] M Lamboni, H Monod, and D Makowski. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliability Engineering & System Safety*, 96(4):450–459, 2011.

- [LMQR14] T Lassila, A Manzoni, A Quarteroni, and G Rozza. *Model Order Reduction in Fluid Dynamics: Challenges and Perspectives*, pages 235–273. Springer International Publishing, Cham, 2014.
- [Mor91] MD Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, (33):161–174, 1991.
- [Mur12] KP Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [NMA15] F Negri, A Manzoni, and D Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.
- [ODR20] G Ortali, N Demo, and G Rozza. Gaussian process approach within a data-driven pod framework for fluid dynamics engineering problems. *arXiv preprint arXiv:2012.01989*, 2020.
- [PW16] B Peherstorfer and K Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.
- [QMN16] A Quarteroni, A Manzoni, and F Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*. 2016.
- [QPO⁺18] E Qian, B Peherstorfer, D O’Malley, VV Vesselinov, and K Willcox. Multifidelity monte carlo estimation of variance and sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):683–706, 2018.
- [RR16] A Radermacher and S Reese. POD-based model reduction with empirical interpolation applied to nonlinear elasticity. *International Journal for Numerical Methods in Engineering*, 107(6):477–495, 2016.
- [Sal02] A Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer physics communications*, 145(2):280–297, 2002.
- [SDM21] M Salvador, L Dede’, and A Manzoni. Non intrusive reduced order modeling of parametrized pdes by kernel pod and neural networks. *Computers & Mathematics with Applications*, 104:1–13, 2021.
- [SMPW19] R Swischuk, L Mainini, B Peherstorfer, and K Willcox. Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 179:704–717, 2019.
- [Sob90] IM Sobol’. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe modelirovanie*, 2(1):112–118, 1990.
- [Sud08] B Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979, 2008.
- [TDR13] P Tiso, R Dedden, and D Rixen. A modified discrete empirical interpolation method for reducing non-linear structural finite element models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55973, page V07BT10A043. American Society of Mechanical Engineers, 2013.
- [TR13] P Tiso and DJ Rixen. Discrete empirical interpolation method for finite element structural dynamics. In *Topics in Nonlinear Dynamics, Volume 1*, pages 203–212. Springer, 2013.
- [WHM13] S Walton, O Hassan, and K Morgan. Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. *Applied Mathematical Modelling*, 37(20-21):8930–8945, 2013.
- [WHR19] Q Wang, JS Hesthaven, and D Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of computational physics*, 384:289–307, 2019.

- [WR06] CKI Williams and CE Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [XFPN17] D Xiao, F Fang, CC Pain, and IM Navon. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering*, 317:868–889, 2017.
- [YMK21] L Yang, X Meng, and GE Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.
- [YYG19] J Yu, C Yan, and M Guo. Non-intrusive reduced-order modeling for fluid problems: A brief review. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(16):5896–5912, 2019.
- [ZGH19] Z Zhang, M Guo, and JS Hesthaven. Model order reduction for large-scale structures with local nonlinearities. *Computer Methods in Applied Mechanics and Engineering*, 353:491–515, 2019.
- [ZLL21] Y Zhao, L Li, and K Li. Non-intrusive reduced order modeling of parametric electromagnetic scattering problems through gaussian process regression. *arXiv preprint arXiv:2103.12472*, 2021.

MOX Technical Reports, last issues

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

- 104/2023** Possenti, L.; Gallo, A.; Vitullo, P.; Cicchetti, A.; Rancati, T.; Costantino, M.L.; Zunino, P.
A computational model of the tumor microenvironment applied to fractionated radiotherapy
- 103/2023** Dimola N.; Kuchta M.; Mardal K.A.; Zunino P.
Robust Preconditioning of Mixed-Dimensional PDEs on 3d-1d domains coupled with Lagrange Multipliers
- 101/2023** Formaggia, L.; Zunino, P.
Hybrid dimensional models for blood flow and mass transport
- 98/2023** Lespagnol, F.; Grandmont, C.; Zunino, P.; Fernandez, M.A.
A mixed-dimensional formulation for the simulation of slender structures immersed in an incompressible flow
- 100/2023** Vitullo, P.; Cicci, L.; Possenti, L.; Coclite, A.; Costantino, M.L.; Zunino, P.
Sensitivity analysis of a multi-physics model for the vascular microenvironment
- 96/2023** Bonetti, S.; Botti, M.; Antonietti, P.F.
Robust discontinuous Galerkin-based scheme for the fully-coupled non-linear thermo-hydro-mechanical problem
- 95/2023** Barnafi, N. A.; Regazzoni, F.; Riccobelli, D.
Reconstructing relaxed configurations in elastic bodies: mathematical formulation and numerical methods for cardiac modeling
- 93/2023** Andrini, D.; Magri, M.; Ciarletta, P.
Optimal surface clothing with elastic nets
- 92/2023** Burzacchi, A.; Rossi, L.; Agasisti, T.; Paganoni, A. M.; Vantini, S.
Commuting time as a determinant of higher education students' performance: the case of Politecnico di Milano
- Burzacchi, A.; Rossi, L.; Agasisti, T.; Paganoni, A. M.; Vantini, S.
Commuting time as a determinant of higher education students' performance: the case of Politecnico di Milano