A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems

Mariella Kast*, Mengwu Guo, Jan S. Hesthaven

Chair of Computational Mathematics and Simulation Science,
École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

Abstract

We propose a non-intrusive reduced basis (RB) method for parametrized nonlinear partial differential equations (PDEs) that leverages models of different accuracy. The method extracts parameter locations from a collection of low-fidelity (LF) snapshots for the efficient creation of a high-fidelity (HF) reduced basis and employs multi-fidelity Gaussian process regression (GPR) to approximate the combination coefficients of the reduced basis. LF data is assimilated either via projection onto an LF basis or via an interpolation approach inspired by bifidelity reconstruction. The correlation between HF and LF data is modeled with hyperparameters whose values are automatically determined in the regression step. The proposed methods not only leverage the assimilated LF data to reduce the cost of the offline phase, but also allow for a fast evaluation during the online stage, independent of the computational cost of neither the low- nor the high-fidelity solution. Numerical studies demonstrate the effectiveness of the proposed approach on manufactured examples and problems in nonlinear structural mechanics. Clear benefits of using lower resolution models rather than reduced physics models are observed in both the basis selection and the regression step. An active learning scheme is used for additional snapshot selection at locations with high error. The speed-up in the online evaluation and the high accuracy of extracted quantities of interest makes the multifidelity RB method a powerful tool for outer-loop applications in engineering, as exemplified in uncertainty quantification.

Keywords: multifidelity methods, non-intrusive reduced order modeling, Gaussian process regression, vector-valued machine learning, nonlinear structural analysis

1. Introduction

In recent years, computer-aided engineering (CAE) has become an increasingly important tool in the design, assessment and maintenance of engineering systems across a broad range of industries such as aerospace, offshore and automotive engineering. Simulations that rely on parametrized partial differential equations (PDEs) are increasingly used to replace physical experiments in applications for structural or performance optimization [9] and reliability analysis [19]. A large number of model evaluations is generally required to fully explore the parameter space, and engineers often face a trade-off in their design of experiments: accurately resolving the phenomena of interest versus obtaining the desired result with a limited budget of resources and time constraints.

As highly accurate models induce a high cost in terms of both computational time and memory, their repeated evaluation is often infeasible for application domains which involve many queries. Computationally cheaper low-fidelity (LF) models can be generated by reducing either the accuracy of the computational model, e.g. using a coarser mesh, or by simplifying the underlying physical model, e.g. linearizing the

*Corresponding author.
Email addresses: mariella.kast@epfl.ch (Mariella Kast), mengwu.guo@epfl.ch (Mengwu Guo), Jan.Hesthaven@epfl.ch (Jan S. Hesthaven)

governing equations, or using simplifying assumptions such as 1D beam models for frame components [4]. While the gain in speed-up can be significant when the LF model replaces the high-fidelity (HF) model, the accuracy of and confidence in model predictions generally suffer as a consequence of the simplifications [25].

To address this, a significant amount of work has been invested into deriving cheap reduced models based on the original HF setup. Reduced basis (RB) modeling [16, 27], which projects the original governing equations onto a reduced basis extracted from a set of HF snapshots, is one of the most prominent candidates in this category. Such reduced models exploit the intrinsic similarities between the simulation outcomes at different parameter values and operate in an offline-online framework. While the RB schemes have been demonstrated to work well on linear problems, nonlinear problems often require dedicated and intrusive treatment of both nonlinearities and non-affine dependency on the parameters to achieve significant speed-up [8, 23]. Recently, the combination of a reduced basis with a data-driven machine learning model for the reconstruction of the solutions has resulted in a non-intrusive approach [12, 13, 17], which does not require a modification of the original HF solver and can been extended to incorporate physical constraints such as boundary conditions [28] and continuity [31]. However, for general nonlinear problems, these techniques often require a higher number of model evaluations to construct a reliable surrogate in the first place. In this work, we propose an extension to this non-intrusive approach that reduces the number of HF solutions by leveraging a large number of LF solutions.

In the case of a single quantity of interest (QoI), *multifidelity* (MF) methods, which fuse models of different accuracy levels currently, have received substantial attention. The ultimate goal of such MF methods consists in reducing the cost of the outer loop application, while maintaining an acceptable accuracy in the computation of QoIs via a careful combination of information from the different models. MF methods have been successfully applied to importance sampling [24], subset simulation [29] and a variety of Monte Carlo techniques [26]. We refer to [25] for a comprehensive overview of MF techniques.

We conjecture that there can be a significant gain in flexibility by first constructing a general, accurate surrogate model for the parametrized solution field of the governing PDE system before the QoIs are explored. Particularly in situations where the solution field exhibits better continuity properties than the QoI itself, it may be more feasible to reconstruct the smooth solution and then apply the discontinuous operator to extract the QoI. Consider, for example, the location of maximum stress in a structure: while the stress field changes smoothly throughout the parameter domain, the location of the maximum stress might jump between two or several critical points. Access to a general surrogate also allows to explore parametric dependence a posteriori; i.e., for different parameter distributions, the correlation between outputs and sensitivities can be investigated with the same surrogate model. One example of this is *bifidelity reconstruction* [20], which has emerged as a popular technique which has proven its success in the computation of statistical moments of some QoIs [32]. Recent efforts have been extended to construct an estimator for an upper error bound [14]. Yet, the online phase of bifidelity reconstruction continues to scale with the evaluation cost of the LF model. As an alternative, we propose a regression-based approach that uses Gaussian processes (GPs) [30], which decouples the model evaluation in the offline phase and the fast solution recovery in the online phase. We point out the contribution in [2], which first proposes a combination of MF Gaussian process regression (GPR) and a reduced basis to reconstruct the solution field and provides initial results on simulated aerodynamic flow for a small set of design points.

In this paper, we propose techniques for ROMs of parametrized nonlinear PDEs which exploit the multifidelity setup in two ways, first in the reduced basis construction and secondly in the Gaussian process regression for the solution construction in the reduced basis space. In Section 2, we briefly state the problem setup together with a typical case in nonlinear structural analysis. In Section 3, we propose an algorithm for the construction of a MF reduced basis, that leverages a rank revealing QR decomposition on the LF solution snapshots to select salient parameter points for the HF basis. We further discuss the challenges of assimilating LF data so that it can be used in conjunction with the HF basis. Section 4 focuses on the multifidelity regression approach that enables the online-offline decoupling. We use an autoregressive formulation [22] to train GPs on both HF and LF data to learn the mapping between parameters and the RB expansion coefficients. In the online phase, the trained GP models are used to predict the expansion coefficients at unseen parameter combinations, to allow for a fast recovery of an (approximate) reduced order solution. We also discuss the challenges of sparse HF data and cost-concious HF snapshot selection

through an active learning criterion. In our numerical studies in Section 5, we first focus on a manufactured 1D example to illustrate salient features of the developed techniques and compare the performance of single and multifidelity regression. We then proceed to cases from structural solid mechanics and fluid-structure interaction in 2D with two or three parameters. Conclusions are drawn in Section 6.

2. Problem statement

In this work, we consider parametrized nonlinear problems of the following form:

$$\mathcal{N}[\boldsymbol{u}(\boldsymbol{X};\boldsymbol{\mu});\boldsymbol{\mu}] = g(\boldsymbol{X};\boldsymbol{\mu})\,, \quad (\boldsymbol{X},\boldsymbol{\mu}) \in \Omega_X \times \Omega_\mu\,, \tag{1}$$

where $\boldsymbol{\mu}$ is the parameter vector with its variation over the parameter domain Ω_μ, $\boldsymbol{X}$ is the space coordinates, $\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{u}(\boldsymbol{X};\boldsymbol{\mu})$ denotes the parameter-dependent solution field defined in the spatial domain Ω_X, $\mathcal{N}[\cdot;\boldsymbol{\mu}]$ is a nonlinear operator, and g is the source term.

In this work, the discrete solution to (1) for a given parameter value $\boldsymbol{\mu}$ is obtained with a finite element (FE) method on a discrete function space $\mathcal{V}_h$, spanned by basis functions $\boldsymbol{\phi}_i(\boldsymbol{X})$, $i = 1, 2, \cdots, N_h$, with N_h being the number of degree of freedoms (DOF). The proposed methods are also applicable to other computational solvers that rely on domain discretization. In the following, it is assumed that we can compute an approximate FE solution $\boldsymbol{u}_h(\boldsymbol{X};\boldsymbol{\mu}) \in \mathcal{V}_h$ of the nonlinear problem at any location $\boldsymbol{X} \in \Omega_X$ and any parameter sample $\boldsymbol{\mu} \in \Omega_\mu$. The FE solutions $\boldsymbol{u}_h$ are expressed in terms of the basis functions as

$$\boldsymbol{u}_h(\boldsymbol{X};\boldsymbol{\mu}) = \sum_{i=1}^{N_h} (\mathbf{u}_h(\boldsymbol{\mu}))_i \boldsymbol{\phi}_i(\boldsymbol{X}). \tag{2}$$

Note that N_h is also the dimension of the expansion coefficient vector $\mathbf{u}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$.

We are ultimately interested in the reconstruction of specific QoIs that depend on the recovered solution field $\boldsymbol{u}$, defined by a functional f as QoI $= f(\boldsymbol{u})$, such as the maximum stress, displacement at critical points, etc. For reliability analysis, one often computes an expected value $\mathbb{E}[\text{QoI}]$ or a probability to exceed a threshold value t_{QoI}, $\mathbb{P}(\text{QoI} > t_{\text{QoI}})$, for a given parameter distribution. Typically these quantities are estimated with a Monte Carlo (MC) method, which requires a large number of evaluations at distinct parameter values $\boldsymbol{\mu}^{(j)}$, $j = 1, 2, \cdots, N_{\text{MC}}$. We use the standard MC estimator I_{MC} given by

$$I_{\text{MC}} = \frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} f(\boldsymbol{u}_h(\boldsymbol{\mu}^{(j)})). \tag{3}$$

To exemplify the application area for our developed method, we shortly discuss a typical problem from nonlinear structural analysis. As shown in Figure 1, the loading force and stiffness of certain structural components depend on a parameter vector $\boldsymbol{\mu} \in \Omega_\mu$. Applying the principle of virtual work and specifying appropriate Dirichlet boundary conditions leads to the governing equation in the variational form:

$$\int_{\Omega_X} \boldsymbol{S}(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}) : D_{\boldsymbol{v}}\boldsymbol{E}(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}) \mathrm{d}\Omega = \int_{\Omega_X} \boldsymbol{b}(\boldsymbol{\mu})^T \boldsymbol{v} \mathrm{d}\Omega + \int_{\Gamma_{\text{N}}} \boldsymbol{t}(\boldsymbol{\mu})^T \boldsymbol{v} \mathrm{d}\Gamma\,, \quad \forall \boldsymbol{v} \in \mathcal{V}\,. \tag{4}$$

with the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$, the Lagrangian strain tensor $\boldsymbol{E}$, the body forces $\boldsymbol{b}$ and tractions $\boldsymbol{t}$ on the Neumann boundary Γ_{N}. The Lagrangian strain tensor $\boldsymbol{E}(\cdot)$ evaluates the change in length between two physical points in space and is defined as a nonlinear operator as

$$\boldsymbol{E}(\boldsymbol{u}) = \frac{1}{2}\left((\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u})^T + \boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u} + (\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u})^T \boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\right). \tag{5}$$

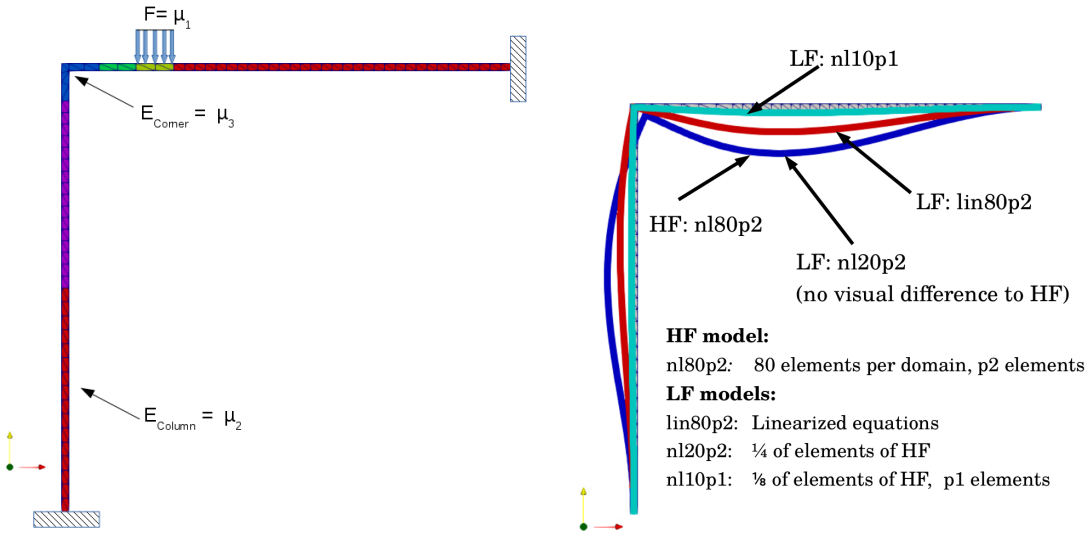


Figure 1: A motivating example from structural mechanics. A frame is loaded from above and experiences large deformations, so that linear strain assumptions no longer hold. The principle of virtual work leads to a nonlinear PDE which is solved numerically with the FE method. The loading force and stiffness of selected domains depend on a parameter vector $\boldsymbol{\mu}$.

The small strain tensor $\boldsymbol{\varepsilon}(\cdot)$,

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left((\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u})^T + \boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\right), \tag{6}$$

replaces $\boldsymbol{E}(\cdot)$ to simplify calculations for the linearized LF model "lin80p2". However, such an approximation only provides valid results for small deformations. To link the strains to the corresponding stresses $\boldsymbol{S}$, we use the hyperelastic constitutive laws that consider nonlinearities in the material response [4, 18], such as the Saint Venant-Kirchhoff material model in terms of the Young modulus E and the Poisson ratio ν:

$$\boldsymbol{S}_{\mathrm{VK}} = \frac{E\nu}{(1+\nu)(1-2\nu)}\mathrm{tr}(\boldsymbol{E})I + \frac{E}{(1+\nu)}\boldsymbol{E}, \tag{7}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator.

We generally use second order Lagrange polynomials ("p2") for the elementwise approximation, whereas the most coarse LF model ("nl10p1") only uses linear basis functions. The governing equations are implemented in FeniCS [3] and the nonlinear variational problem is solved with an iterative Newton method. The HF model ("nl80p2") is simulated by adopting 80 p2 elements per domain. We further have access to solutions of varying accuracy by either linearizing the equations ("lin80p2"), reducing the resolution of the FE mesh ("nl20p2") and/or using a lower polynomial order for the basis functions ("nl10p1"), see Figure 1 for a comparison of the computed deformations.

3. Efficient construction of a reduced basis

3.1. Classic approach via proper orthogonal decomposition (POD)

Ideally, we seek to accurately capture the *analytic* solution manifold $\mathcal{M} = \{\boldsymbol{u}(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \Omega_\mu\}$ of the parametrized problem. This is clearly unavailable and we can only compute approximations of a discretized counterpart $\mathcal{M}_h = \{\boldsymbol{u}_h(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \Omega_\mu\}$. However, even $\mathcal{M}_h$ generally cannot be explored fully, as we can only evaluate a finite number of solutions $\boldsymbol{u}_h(\boldsymbol{\mu}^{(i)})$, often referred to as "snapshots". Given a discrete point-set $\Theta = \{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, ..., \boldsymbol{\mu}^{(N_s)}\}$, which contains $|\Theta| = N_s$ parameter samples, one computes only N_s evaluations

of the form $\boldsymbol{u}_h(\boldsymbol{\mu}^{(i)})$. This defines a subspace of the solution manifold $\mathcal{M}_h$ as

$$\mathcal{M}_\Theta = \text{span}\{\boldsymbol{u}_h(\boldsymbol{\mu}^{(1)}), \boldsymbol{u}_h(\boldsymbol{\mu}^{(2)}), ..., \boldsymbol{u}_h(\boldsymbol{\mu}^{(N_s)})\} \subset \mathcal{M}_h \subset \mathcal{V}_h. \tag{8}$$

For a sufficiently fine sampling set Θ, we conjecture that $\mathcal{M}_\Theta$ accurately represents the solution manifold.

In reduced order modeling, one aims to exploit the low-dimensional structure that the solution manifold $\mathcal{M}$ often exhibits for parametrized PDEs, i.e., we seek a set of basis functions $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, ..., \boldsymbol{\psi}_L\}$, with $L \ll N_h$, which span a subspace of $\mathcal{M}_h$. We aim to exploit this low-rank structure of the snapshot matrix S, that collects the vectors $\mathbf{u}_h(\boldsymbol{\mu}^{(i)})$ in its columns:

$$S = \left[\mathbf{u}_h(\boldsymbol{\mu}^{(1)}) \mid \mathbf{u}_h(\boldsymbol{\mu}^{(2)}) \mid ... \mid \mathbf{u}_h(\boldsymbol{\mu}^{(N_s)})\right]. \tag{9}$$

To construct a low-rank basis, the POD employs the singular-value decomposition (SVD) of the snapshot matrix S, i.e.

$$S = U\Sigma Z^T, \tag{10}$$

where U and Z are unitary matrices, and Σ is diagonal with the singular values $\sigma_1 \geq \sigma_2 \geq \cdots$ in a non-increasing order. For numerical stability and/or to reduce the cost, one may also first construct the Gramian $G = S^T S$ and compute the eigenvalue decomposition [20].

Let $\mathbb{Y}_k = \{W \in \mathbb{R}^{N_h \times k} : W^T W = I_k\}$ be the set of all k-dimensional orthogonal bases of S. We further denote by $\sigma_k(S) = \sigma_k$ the kth largest singular value of S. The Schmidt-Eckart-Young theorem states that the first k columns of U minimize the projection error among all $W \in \mathbb{Y}_k$: $V = U[:, : k]$, i.e., the projection error in the Frobenius norm can further be recovered as

$$\min_{W \in \mathbb{Y}_k} \left\|S - WW^T S\right\|_F^2 = \left\|S - VV^T S\right\|_F^2 = \sum_{i=1}^{N_s} \left\|\mathbf{u}_h(\boldsymbol{\mu}^{(i)}) - VV^T \mathbf{u}_h(\boldsymbol{\mu}^{(i)})\right\|_2^2 = \sum_{i=k+1}^{N_s} \sigma_i^2, \tag{11}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, i.e., $\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^T\mathbf{a}}$, with $\mathbf{a} \in \mathbb{R}^n$ and $n \in \mathbb{N}$. Hence V is the optimal k-rank approximation of $\mathcal{M}_h$ based on the snapshots collected in S.

3.2. Rank revealing QR decomposition for a multifidelity basis

As a single HF snapshot $\mathbf{u}_h(\boldsymbol{\mu}^{(i)})$ of the parametrized problem is costly to evaluate, the optimal basis $V = U[:, : k]$ of size k of the POD is often unavailable since we cannot afford to sample the parameter space finely enough to ensure that $\mathcal{M}_\Theta \approx \mathcal{M}_h$. Instead, we seek a set Θ_k of cardinality $k \leq \text{rank}(S)$, which contains "optimal" parameter locations, so that the corresponding snapshots allow us to construct a sufficiently close approximation to $\mathcal{M}_{\Theta_k} \approx \mathcal{M}_h$ and, consequently, a good basis. To reduce the complexity of choosing the sampling points $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^k$, we restrict the search to the discrete set of candidate parameters Θ. When we have access to the snapshots for all elements in Θ, i.e. the original S, this task is known in the literature as the *Column Subset Selection Problem* (CSSP), which is likely NP-hard [6].

In this work, we will approximately solve the CSSP on a LF snapshots matrix S_{LF}. Such approaches have become popular for bifidelity reconstruction [14, 20], which exploits solutions of LF models to reconstruct interpolatory HF solutions. Instead of assembling HF snapshots S_{HF}, we assemble the cheaper S_{LF} and select the most important parameter locations by approximately solving the CSSP for S_{LF} with a rank revealing QR decomposition (RRQR). Clearly, the quality of such an approach will depend on how well S_{LF} reflects the structure of S_{HF} and our numerical studies will asses how this substitution influences the projection error of the multifidelity basis.

The first algorithm for a rank revealing QR decomposition of a matrix S was proposed in [7]. It is a modified variant of the Householder QR factorization procedure and computes a factorization QR and a

column pivoting P such that

$$S_{\text{pivoted}} = SP = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}, \tag{12}$$

and the absolute values of the diagonal entries of the upper triangular matrix R are non-increasing. Here the matrices Q and P are products of Householder matrices and interchange matrices respectively, i.e., in each QR step, the columns are pivoted so that the first columns are maximally linearly independent. As a side product, this procedure also determines the rank of S via the numbers of pivots. We select $k \leq \text{rank}(S)$ snapshots of S according to the ordering of the pivoting P and assemble a smaller matrix $S_k = Q_1 R_{11}$, which contains the first k columns of S_{pivoted}. Note that $\text{rank}(S_k) = k$.

Only then do we perform an SVD on S_k as

$$S_k = U_k \Sigma_k Z_k^T, \tag{13}$$

where the first k columns of U_k form the best orthonormal basis $U_k[:, : k']$ of S_k in a least squares sense for $k' \leq k$. Taking the reduced basis as $V = U[:, : k]$, it is easy to demonstrate that the projection error of the snapshots S onto the reduced space spanned by V can be given as

$$\left\| S - VV^T S \right\|_\xi = \| R_{22} \|_\xi, \tag{14}$$

with $\xi \in \{2, F\}$ representing either the spectral norm or the Frobenius norm, i.e., the projection error depends on the norm of the the residual triangular factor R_{22} and we do not introduce a projection error for the k snapshots. Algorithm 1 summarizes the proposed algorithm in the bifidelity case. For our numerical experiments, we rely on the linear algebra package of scipy to perform the RRQR (`scipy.linalg.qr(..)`), which provides the desired properties for the considered snapshot matrices.

Algorithm 1 Construction of a bifidelity basis

Input: Parameter range $\mathcal{P} = [\boldsymbol{\mu}_{\min}, \boldsymbol{\mu}_{\max}]$, cut-off tolerance for rank ϵ_{rank}, maximum size k of basis.
Output: An orthonormal basis V_{HF} of size at most k

1: Evaluate LF at snapshot locations $\boldsymbol{\mu} \in \Theta$, with $\Theta \subseteq \mathcal{P}$ sufficiently fine, form the snapshot matrix S_{LF} with $\{\mathbf{u}(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \Theta\}$ as column vectors.
2: Compute a rank revealing QR decomposition: $S_{\text{LF}} P = QR$ with ϵ_{rank} as stopping criterion.
3: Select $\min(k, \text{rank}(S_{\text{LF}}))$ snapshot locations in Θ according to the ordering induced by P: $\Theta_k^{\text{LF}} \subset \Theta$.
4: Evaluate HF for the parameters in Θ_k^{LF} and construct the snapshot matrix S_{HF}.
5: Compute a POD, $S_{\text{HF}} = U \Sigma Z^T$, and set $V_{\text{HF}} = U[:, : \text{rank}(S_{\text{HF}})]$.

3.3. Approximating the solution in the reduced basis space

For a given HF solution $\boldsymbol{u}_{\text{HF}}(\boldsymbol{\mu})$ and a HF basis V_{HF} of rank k, we perform the projection onto the reduced space at an algebraic level and express the reduced order solution $\boldsymbol{u}_{\text{HF}}^{\text{RB}}(\boldsymbol{\mu})$ in functional form:

$$\begin{aligned} \boldsymbol{u}_{\text{HF}}^{\text{RB}}(\boldsymbol{\mu}; \boldsymbol{X}) &= \sum_{i=1}^{N_h} (V_{\text{HF}} V_{\text{HF}}^T \mathbf{u}_{\text{HF}}(\boldsymbol{\mu}))_i \boldsymbol{\phi}_i(\boldsymbol{X}) \\ &= \sum_{l=1}^{k} \left(\left(V_{\text{HF}}^T \mathbf{u}_{\text{HF}}(\boldsymbol{\mu}) \right)_l \sum_{i=1}^{N_h} (V_{\text{HF}})_{il} \boldsymbol{\phi}_i(\boldsymbol{X}) \right) \\ &= \sum_{l=1}^{k} \left(V_{\text{HF}}^T \mathbf{u}_{\text{HF}}(\boldsymbol{\mu}) \right)_l \boldsymbol{\psi}_l(\boldsymbol{X}) = \sum_{l=1}^{k} \left(\mathbf{c}_{\text{HF}}(\boldsymbol{\mu}) \right)_l \boldsymbol{\psi}_l(\boldsymbol{X}) \end{aligned} \tag{15}$$

We observe that (15) leads to a reduced function space spanned by the basis functions $\boldsymbol{\psi}_l = \sum_{i=1}^{N_h} (V_{\mathrm{HF}})_{il} \boldsymbol{\phi}_i$, $1 \leq l \leq k$, and the projection/combination coefficients are collected in $\mathbf{c}_{\mathrm{HF}}(\boldsymbol{\mu}) = V_{\mathrm{HF}}^T \mathbf{u}_{\mathrm{HF}}(\boldsymbol{\mu})$.

Note that the algebraic projection in (15) is not identical to the projection of the original solution $\boldsymbol{u}_{\mathrm{HF}}^{\mathrm{RB}}(\boldsymbol{\mu})$ onto the new basis functions $\boldsymbol{\psi}_l$ with respect to the L^2 inner product, i.e., $V_{\mathrm{HF}}[:, l]^T \mathbf{u}_{\mathrm{HF}} \neq \langle \boldsymbol{\psi}_l, \boldsymbol{u}_{\mathrm{HF}}^{\mathrm{RB}} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product as

$$\langle \boldsymbol{a}(\boldsymbol{X}), \boldsymbol{b}(\boldsymbol{X}) \rangle = \int_\Omega \boldsymbol{a}(\boldsymbol{X}) \boldsymbol{b}(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \,, \quad \boldsymbol{a}, \boldsymbol{b} \in \mathcal{V}_h \,, \tag{16}$$

and the corresponding L^2 norm $\| \cdot \|_{L^2}$ is defined as $\|\boldsymbol{a}\|_{L^2} = \sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle}$, $\boldsymbol{a} \in \mathcal{V}_h$. Moreover, we can also characterize the mass matrix M of the discrete space as

$$M_{ij} = \langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle \,, \quad 1 \leq i, j \leq N_h \,. \tag{17}$$

Following [17], the L^2 error between the truth and the reduced solution functions is bounded as

$$\left\| \boldsymbol{u}_{\mathrm{HF}} - \boldsymbol{u}_{\mathrm{HF}}^{\mathrm{RB}} \right\|_{L^2}^2 = \langle \boldsymbol{u}_{\mathrm{HF}} - \boldsymbol{u}_{\mathrm{HF}}^{\mathrm{RB}}, \boldsymbol{u}_{\mathrm{HF}} - \boldsymbol{u}_{\mathrm{HF}}^{\mathrm{RB}} \rangle = \left(\mathbf{u}_{\mathrm{HF}} - V_{\mathrm{HF}} V_{\mathrm{HF}}^T \mathbf{u}_{\mathrm{HF}} \right)^T M \left(\mathbf{u}_{\mathrm{HF}} - V_{\mathrm{HF}} V_{\mathrm{HF}}^T \mathbf{u}_{\mathrm{HF}} \right) \tag{18}$$

$$= \left\| M^{1/2} \left(\mathbf{u}_{\mathrm{HF}} - \mathbf{u}_{\mathrm{HF}}^{\mathrm{RB}} \right) \right\|_2^2 \leq \|M\|_2 \left\| \mathbf{u}_{\mathrm{HF}} - \mathbf{u}_{\mathrm{HF}}^{\mathrm{RB}} \right\|_2^2 \,, \tag{19}$$

i.e., the spectral norm of the mass matrix influences the quality of the approximated solution in the reduced space. Numerical studies in [17] show that (18) converges in unison with the algebraic projection error.

3.4. Generating LF coefficients for the HF basis

For the MF basis algorithm, we only require that the LF model captures the fundamental parametric dependence of the HF model. We now discuss how the RB solutions $\boldsymbol{u}_{\mathrm{LF}}^{\mathrm{RB}}(\boldsymbol{\mu})$ and the expansion coefficients $\mathbf{c}_{\mathrm{LF}}(\boldsymbol{\mu})$ of the low fidelity are related to the corresponding quantities of the high fidelity, $\boldsymbol{u}_{\mathrm{HF}}^{\mathrm{RB}}(\boldsymbol{\mu})$ and $\mathbf{c}_{\mathrm{HF}}(\boldsymbol{\mu})$. We denote by $\boldsymbol{\varphi}_i$ the LF basis functions, so that $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_{h_{\mathrm{LF}}}}$ span the LF function space $\mathcal{V}_{h_{\mathrm{LF}}}$. Furthermore, let V_{LF} denote the matrix form of the reduced basis of the LF.

3.4.1. Nested function spaces

When $\mathcal{V}_{h_{\mathrm{LF}}} \subseteq \mathcal{V}_{h_{\mathrm{HF}}}$, we can express the link between the HF and LF model via an analytical correction term $\boldsymbol{u}_\delta$, i.e.

$$\boldsymbol{u}_{\mathrm{HF}}(\boldsymbol{\mu}) = \boldsymbol{u}_{\mathrm{LF}}(\boldsymbol{\mu}) + \boldsymbol{u}_\delta(\boldsymbol{\mu}) \,, \tag{20}$$

in which the LF solution can be written in the following form since $\mathcal{V}_{h_{\mathrm{LF}}} \subseteq \mathcal{V}_{h_{\mathrm{HF}}}$:

$$\boldsymbol{u}_{\mathrm{LF}}(\boldsymbol{\mu}) = \sum_{i=1}^{N_{h_{\mathrm{LF}}}} (\mathbf{u}_{\mathrm{LF}}(\boldsymbol{\mu}))_i \boldsymbol{\varphi}_i = \sum_{j=1}^{N_h} (\hat{\mathbf{u}}_{\mathrm{LF}}(\boldsymbol{\mu}))_j \boldsymbol{\phi}_j \,, \tag{21}$$

where $\mathbf{u}_{\mathrm{LF}}$ and $\hat{\mathbf{u}}_{\mathrm{LF}}$ collects the combination coefficients of the LF and HF basis functions, respectively. Thus we can algebraically project the correction term onto the HF reduced space as

$$\boldsymbol{u}_\delta^{\mathrm{RB}} = \sum_{l=1}^{k} (V_{\mathrm{HF}}^T (\mathbf{u}_{\mathrm{HF}} - \hat{\mathbf{u}}_{\mathrm{LF}}))_l \boldsymbol{\psi}_l \,. \tag{22}$$

Moreover, we have that

$$\langle \boldsymbol{u}_{\mathrm{LF}}, \boldsymbol{\phi}_j \rangle = \sum_{i=1}^{N_{h_{\mathrm{LF}}}} (\mathbf{u}_{\mathrm{LF}}(\boldsymbol{\mu}))_i \langle \boldsymbol{\varphi}_i, \boldsymbol{\phi}_j \rangle = \sum_{l=1}^{N_h} (\hat{\mathbf{u}}_{\mathrm{LF}}(\boldsymbol{\mu}))_l \langle \boldsymbol{\phi}_l, \boldsymbol{\phi}_j \rangle \,, \quad 1 \leq j \leq N_h, \tag{23}$$

and we recover a system of linear equations

$$M\hat{\mathbf{u}}_{\mathrm{LF}} = T\mathbf{u}_{\mathrm{LF}}\,, \tag{24}$$

where the entries of $T \in \mathbb{R}^{N_h \times N_{h_{\mathrm{LF}}}}$ correspond to the inner products between the basis functions, i.e. their correlations

$$T_{jk} = \langle \boldsymbol{\phi}_j, \boldsymbol{\varphi}_k \rangle\,, \quad 1 \le j \le N_h\,,\ 1 \le k \le N_{h_{\mathrm{LF}}}\,. \tag{25}$$

Combining (22), (24) with (15) allows to express the correction term explicitly as

$$\boldsymbol{u}_\delta^{\mathrm{RB}}(\boldsymbol{\mu}) = \sum_{l=1}^{k} \left[V_{\mathrm{HF}}^T \left(\mathbf{u}_{\mathrm{HF}}(\boldsymbol{\mu}) - M^{-1}T\mathbf{u}_{\mathrm{LF}}(\boldsymbol{\mu})\right)\right]_l \boldsymbol{\psi}_l = \sum_{l=1}^{k} \left[(\mathbf{c}_{\mathrm{HF}}(\boldsymbol{\mu}))_l - \left(V_{\mathrm{HF}}^T M^{-1} T\mathbf{u}_{\mathrm{LF}}(\boldsymbol{\mu})\right)_l\right] \boldsymbol{\psi}_l\,. \tag{26}$$

If $\boldsymbol{u}_\delta^{\mathrm{RB}}(\boldsymbol{\mu})$ is known for all $\boldsymbol{\mu} \in \Omega_\mu$, it can be used as a "bridge function" between HF and LF. This idea will play an important role in the proposed multifidelity regression scheme.

3.4.2. Generic function spaces

When the LF and HF models belong to two different spaces $\mathcal{V}_h$ and $\mathcal{V}_{h'}$, which may not even represent discretizations on the same physical domain, we loose this direct link between the LF and HF RB coefficients, expressed in (20). A straightforward workaround, as used in [2], consists of projection onto a corresponding LF basis

$$\mathbf{c}_{\mathrm{LF}} = V_{\mathrm{LF}}^T \mathbf{u}_{\mathrm{LF}}\,. \tag{27}$$

It is clear that $\mathbf{c}_{\mathrm{LF}}$ still contains information about $\mathbf{c}_{\mathrm{HF}}$. However, the question remains of how to best extract this information, especially if V_{LF} and V_{HF} do not have the same rank. In [2], the two coefficient vectors are matched entry-wise. However, it is not guaranteed that V_{LF} and V_{HF} actually represent the same RB modes in the same order, so that a mismatch between the expansion coefficients may occur, e.g. if the modes of the LF model do not carry the same energy as the corresponding modes of the HF. In such a case, no useful information can be extracted from the LF model. This is an issue that will be addressed once we introduce our information fusion scheme in the regression step. As an alternative, bifidelity reconstruction [20] provides an excellent tool as a lifting operator between LF and HF data. We propose a two step procedure as follows.

Consider $G = (S_k)_{\mathrm{LF}}^T (S_k)_{\mathrm{LF}}$, the Gramian of the LF snapshot matrix $(S_k)_{\mathrm{LF}}$ of size k selected via rank revealing QR. We use the corresponding HF snapshots $(S_k)_{\mathrm{HF}}$ to perform a bifidelity reconstruction in a least squares sense as follows:

$$\hat{\mathbf{c}}_{\mathrm{LF}} = G^{-1}(S_k)_{\mathrm{LF}}^T \mathbf{u}_{\mathrm{LF}}\,, \tag{28}$$

$$\hat{\mathbf{u}}_{\mathrm{LF}\nearrow\mathrm{HF}} = (S_k)_{\mathrm{HF}} \hat{\mathbf{c}}_{\mathrm{LF}}\,, \tag{29}$$

in which $\hat{\mathbf{c}}_{\mathrm{LF}}$ collects the combination coefficients of the HF snapshots $(S_k)_{\mathrm{HF}}$, and $\hat{\mathbf{u}}_{\mathrm{LF}\nearrow\mathrm{HF}}$ denotes the reconstruction of the solution vector with the HF size. In the second step, one projects onto the HF basis V_{HF}:

$$\mathbf{c}_{\mathrm{LF}\nearrow\mathrm{HF}} = V_{\mathrm{HF}}^T \hat{\mathbf{u}}_{\mathrm{LF}\nearrow\mathrm{HF}}\,. \tag{30}$$

These coefficients provide an alternative to (27) and do not suffer the risk of mismatching the basis modes. In the following, we refer to the strategy in (27) as **"L-proj"** and to the two step lifting approach of (30) as **"LH-interp"**. When either of these strategies are used in conjunction with the HF basis, we recover a bifidelity reconstruction approach as described in [20]. We propose an additional regression step that decouples the online phase from the LF model and automatically determines a correction between HF and LF coefficients in the spirit of the analytic case in (26).

4. Multifidelity solution reconstruction

Let us now discuss how to recover the HF coefficients for fast RB solution construction. In [12] and [17] regression techniques for supervised learning are used to approximate the mapping between parameters and HF coefficients. However, both the Gaussian process regression (GPR) in [12] and the training of the Neural Network in [17] require a large number of HF model evaluations, i.e., if we employ a similar approach, we loose the advantage of having consciously constructed a basis with very few HF evaluations.

Instead, we propose to extend the work in [12] by a second layer of fidelity, so that the GPR can leverage the LF data to efficiently "learn" the trend of the HF data in the offline phase. We thus seek to effectively reduce the number of HF evaluations, while still guaranteeing a desirable accuracy.

4.1. Review of the GPR based RB approach

Offline phase:

As the RB basis is orthogonal, we train an individual regression model for each RB expansion coefficient, i.e., we represent each entry of the mapping $\boldsymbol{\pi}(\boldsymbol{\mu}) : \boldsymbol{\mu} \to (\mathbf{c}_{\mathrm{HF}})$ as an independent GP f with mean function $m(\mathbf{x})$ and kernel $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$ with hyperparameters $\boldsymbol{\theta}$, following [12]:

$$f \sim \mathcal{GP}(m(\mathbf{x}), k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')).$$

To obtain the input training data for the GPR, we evaluate the model at a large set of parameter vectors $X = \{\boldsymbol{\mu}^{(i)}\}_{i=1}^{N_{\mathrm{train}}}$, and assemble the corresponding solution snapshot matrix S_{train}, where X may overlap/coincide with the parameter set Θ of the basis generation. The RB expansion coefficients, as defined in (15), can then efficiently be computed in matrix form:

$$C_{\mathrm{HF}} = \begin{bmatrix} \mathbf{c}_{\mathrm{row}}^{(1)} \\ \dots \\ \mathbf{c}_{\mathrm{row}}^{(k)} \end{bmatrix} = V_{\mathrm{HF}}^T S_{\mathrm{train}}. \tag{31}$$

Each GPR uses one row of C_{HF} as training input $\mathbf{f}(X)^{(j)} = \left(\mathbf{c}_{\mathrm{row}}^{(j)}\right)^T$. For a reduced basis of size k, one thus trains exactly k GPR models $\pi^{(j)} := f^{(j)}$, $1 \leq j \leq k$. The corresponding kernel hyperparameters $\boldsymbol{\theta}^{(j)}$ are found with a maximum likelihood estimate (MLE) of the data [12, 30]. We use the mean of the GP, conditioned on the observed data $\{X, \mathbf{f}(X)\}$, to predict $\pi(\boldsymbol{\mu})^{(j)}$ at unobserved parameter locations $X^* = \{\boldsymbol{\mu}^{(l)}\}_{l=1}^{N_{\mathrm{new}}}$:

$$\mathbf{f}(X^*)|X, \mathbf{f}(X) \sim \mathcal{GP}\left(\mathbf{m}^*(X^*), K^*(X^*, X^*)\right), \tag{32}$$

$$\mathbf{m}^*(X^*) = K_{\boldsymbol{\theta}}(X^*, X)^T K_{\boldsymbol{\theta}}^{(j)}(X, X)^{-1} \mathbf{f}(X)^{(j)} = K_{\boldsymbol{\theta}}(X^*, X)^T \boldsymbol{\alpha}. \tag{33}$$

For efficient prediction in the online phase, we save only the factors $\boldsymbol{\alpha}^{(j)}$

$$\boldsymbol{\alpha}^{(j)} = K_{\boldsymbol{\theta}}^{(j)}(X, X)^{-1} \mathbf{f}(X)^{(j)}, \text{ for } 1 \leq j \leq k. \tag{34}$$

Online phase:

To reconstruct the full RB solution $\boldsymbol{u}_{\mathrm{RB}}(\boldsymbol{\mu}^*)$ online, we first evaluate the approximate mapping at the new location $\boldsymbol{\mu}^*$

$$\hat{\pi}(\boldsymbol{\mu}^*)^{(j)} = K^{(j)}(\{\boldsymbol{\mu}^*\}, X)^T \boldsymbol{\alpha}^{(j)}, \quad \text{for } 1 \leq j \leq k, \tag{35}$$

and then employ the RB expansion coefficients to recover the discrete solution vector of the original function

space $\hat{\mathbf{u}}_{\mathrm{RB}}(\boldsymbol{\mu}^*) = V\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*)$, so that the RB-GPR solution is found as

$$\boldsymbol{u}_{\mathrm{RB}}(\boldsymbol{\mu}^*) \approx \hat{\boldsymbol{u}}_{\mathrm{RB}}(\boldsymbol{\mu}^*) = \sum_{i=1}^{N_h} \hat{\mathbf{u}}_{\mathrm{RB}}(\boldsymbol{\mu}^*)\boldsymbol{\phi}_i = \sum_{i=1}^{N_h}\sum_{j=1}^{k} V_{ij}\hat{\pi}(\boldsymbol{\mu}^*)^{(j)}. \tag{36}$$

4.2. Multifidelity Gaussian process regression

GPR with inputs from different levels of accuracy is known in the literature as cokriging, multi-output or vector-valued learning [1, 5]. Since HF data tends to be more sparse than LF data, we consider the heterotopic data case, i.e., we do not necessarily observe the same locations for both fidelity levels. We continue to train a separate regression model for each mapping entry, and limit our discussion to learning a single HF mapping entry $f_1 = f_{\mathrm{HF}}^{(j)} = \hat{\boldsymbol{\pi}}_{\mathrm{HF}}(\boldsymbol{\mu})^{(j)}$, for which we have access to a LF coefficient mapping $f_2 = f_{\mathrm{LF}}^{(j)} = \hat{\boldsymbol{\pi}}_{\mathrm{LF}}(\boldsymbol{\mu})^{(j)}$ for a given j. Once each $f_{\mathrm{HF}}^{(j)}$ is trained, solution reconstruction follows (36), i.e., the online phase remains essentially identical.

4.2.1. The Linear Model of Coregionalization as a generalization of AR(1)-cokriging

In the simplest form of AR(1)-cokriging [22], the Gaussian process f_1 of the HF data split into one part that is correlated with f_2 via a constant correlation parameter ρ and a correction part, sometimes also called a "bridge function" [15]:

$$f_1(\boldsymbol{\mu}) = u_1(\boldsymbol{\mu}) + \rho u_2(\boldsymbol{\mu}) \tag{37}$$

$$f_2(\boldsymbol{\mu}) = u_2(\boldsymbol{\mu}). \tag{38}$$

We refer to [22] for a precise discussion of the assumptions on the data that leads to this hierarchical form of the correlation. To generalize to other forms of correlation between multiple levels, we write the equations of cokriging in terms of the "linear model of coregionalization" (LMC) [1]. In the LMC with D components, each component f_d is expressed as a linear combination of independent random functions u_q:

$$f_d(\boldsymbol{\mu}) = \sum_{q=1}^{Q} a_{d,q} u_q(\boldsymbol{\mu}). \tag{39}$$

For the class of "sum of separable" (SoS) kernels, the cross-covariance of two composite functions f_d and f_d' is defined by a matrix valued kernel of the form $\mathbf{K}(\boldsymbol{\mu}, \boldsymbol{\mu}') = \sum_{q=1}^{Q} B_q k_q(\boldsymbol{\mu}, \boldsymbol{\mu}')$, whose entries can be constructed from the definition in (39):

$$\mathrm{cov}[f_d(\boldsymbol{\mu}), f_{d'}(\boldsymbol{\mu}')] = (\mathbf{K}(\boldsymbol{\mu}, \boldsymbol{\mu}'))_{d,d'} = \sum_{q=1}^{Q} a_{d,q} a_{d',q} k_q(\boldsymbol{\mu}, \boldsymbol{\mu}'), \tag{40}$$

Nearly all results of the scalar GP case have a direct multidimensional equivalent, found by introducing the notion of a matrix valued kernel:

$$\mathbf{f} \sim \mathcal{GP}(\mathbf{m}(\boldsymbol{\mu}), \mathbf{K}(\boldsymbol{\mu}, \boldsymbol{\mu}')). \tag{41}$$

Given N data points, collected into an information set $\mathbf{S}$ where we have the full information of all D components, we collect the data in an ND output vector $\bar{\mathbf{y}}$. Inference at a new point $\boldsymbol{\mu}^*$ can be performed in vector form, i.e., simultaneously for all components:

$$\mathbf{f}(\boldsymbol{\mu}^*)|\mathbf{S}, \mathbf{f} \sim \mathcal{N}\left(\mathbf{m}^*(\boldsymbol{\mu}^*), \mathbf{K}^*(\boldsymbol{\mu}^*, \boldsymbol{\mu}^*)\right), \tag{42}$$

with the posterior mean vector and covariance matrix given by

$$\mathbf{m^*}(\boldsymbol{\mu}^*) = \mathbf{K}(\{\boldsymbol{\mu}^*\}, \mathbf{X})^T \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}\right)^{-1} \bar{\mathbf{y}}, \tag{43}$$

$$\mathbf{K}^*(\boldsymbol{\mu}^*, \boldsymbol{\mu}^*) = \mathbf{K}(\boldsymbol{\mu}^*, \boldsymbol{\mu}^*) - \mathbf{K}(\{\boldsymbol{\mu}^*\}, \mathbf{X})^T \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}\right)^{-1} \mathbf{K}(\{\boldsymbol{\mu}^*\}, \mathbf{X}). \tag{44}$$

Here $\boldsymbol{\Sigma} = \Sigma \circ I_N$ incorporates the component-wise noise and $\mathbf{K}(\{\boldsymbol{\mu}^*\}, \mathbf{X}) \in \mathbb{R}^{D \times ND}$ has entries $\mathbf{K}(\boldsymbol{\mu}^*, \boldsymbol{\mu}_j)_{d,d'}$ for $1 \leq j \leq$ and $1 \leq d, d' \leq D$. Note that for a concise notation, we dropped the dependence on the hyperparameters and give the expression for a single data point $\boldsymbol{\mu}^*$, while (33) considered a set of data points X^*. For the implementation of (43) and (44), as well as the precise formulations of the heterotopic case, we refer to [1] and [11].

4.2.2. The two-level and three-level case

A two-level hierarchy:

Setting $Q = 2$, $a_{1,1} = 1$, $a_{1,2} = \rho$, $a_{2,1} = 0$, $a_{2,2} = 1$, we recover (37), where u_1 and u_2 denote two independent Gaussian processes, each with their own kernel and hyperparameters. The cross-covariance for the AR(1)-cokriging is simply $(\mathbf{K}(\boldsymbol{\mu}, \boldsymbol{\mu}'))_{1,2} = \rho k_2(\boldsymbol{\mu}, \boldsymbol{\mu}')$, i.e., it is described by the correlation parameter ρ and the properties of the LF GP kernel. The matrices B_q are thus:

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} \rho^2 & \rho \\ \rho & 1 \end{bmatrix}. \tag{45}$$

We use the software package GPy [11] as it conveniently implements the co-regionalization models. GPy prescribes the B matrices in a particular way to ensure that the MLE of the hyperparameters yields positive definite covariance matrices:

$$B = WW^T + \boldsymbol{\kappa} I, \tag{46}$$

where WW^T is a low-rank matrix, usually of rank 1. For $D = 2$, we thus have $W = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T$. To create the structure of the matrix B_1, we fix $W_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, $\boldsymbol{\kappa}_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, which leads to the desired positive semidefinite matrix. For B_2, the construction is similar by setting $W_2 = \begin{bmatrix} \rho & 1 \end{bmatrix}^T$, $\boldsymbol{\kappa}_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, so that the autoregressive approach effectively introduces only one additional hyperparameter for the correlation, which is an advantage when HF data is sparse. To propose a true multifidelity method, one must be able to extend the techniques to more than two levels and we consider two ways of extending the framework to three models.

A three-level hierarchy:

For three levels of fidelity, the methodology of O'Hagan and Kennedy [22] imposes a hierarchy and a sum of GPs as follows:

$$\begin{aligned} f_1(\boldsymbol{\mu}) &= u_1(\boldsymbol{\mu}) + \rho_1 f_2(\boldsymbol{\mu}) = u_1(\boldsymbol{\mu}) + \rho_1 u_2(\boldsymbol{\mu}) + \rho_1 \rho_2 u_3(\boldsymbol{\mu}), \\ f_2(\boldsymbol{\mu}) &= u_2(\boldsymbol{\mu}) + \rho_2 f_3(\boldsymbol{\mu}) = u_2(\boldsymbol{\mu}) + \rho_2 u_3(\boldsymbol{\mu}), \\ f_3(\boldsymbol{\mu}) &= u_3(\boldsymbol{\mu}). \end{aligned} \tag{47}$$

This corresponds to a covariance kernel $\mathbf{K}$ of the form:

$$\mathbf{K}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = W_1 W_1^T k_1(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + W_2 W_2^T k_2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + W_3 W_3^T k_3(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j), \tag{48}$$

where k_i is the kernel function assigned to each of the Gaussian processes u_i and the respective covariance

matrices are defined via W_i as

$$W_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, W_2 = \begin{bmatrix} \rho_1 \\ 1 \\ 0 \end{bmatrix}, W_3 = \begin{bmatrix} \rho_2\rho_1 \\ \rho_2 \\ 1 \end{bmatrix}. \tag{49}$$

In the appendix we discuss how to enforce this exact correlation structure in GPy.

Non-hierarchical coregionalization:

We can also exploit the flexibility of the LCM to exploit other kinds of model dependencies. In some cases, we may have access to several models with lower fidelity without a clear hierarchy between them. For nonlinear structural mechanics, one might consider a lower resolution and a reduced physics model as LF models. Instead of imposing a hierarchy, we now model the two lower fidelity levels by independent GPs and only consider their respective correlation with the high fidelity:

$$\begin{aligned} f_1(\boldsymbol{\mu}) &= u_1(\boldsymbol{\mu}) + \rho_{12} f_2(\boldsymbol{\mu}) + \rho_{13} f_3(\boldsymbol{\mu}) \\ f_2(\boldsymbol{\mu}) &= u_2(\boldsymbol{\mu}) \\ f_3(\boldsymbol{\mu}) &= u_3(\boldsymbol{\mu}) \end{aligned} \tag{50}$$

This structure could allow us to combine the best of two worlds at the cost of fitting an additional hyper-parameter at the HF level f_1. With the formulation in (48), the matrices W_i are

$$W_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, W_2 = \begin{bmatrix} \rho_{12} \\ 1 \\ 0 \end{bmatrix}, W_3 = \begin{bmatrix} \rho_{13} \\ 0 \\ 1 \end{bmatrix}. \tag{51}$$

4.3. A concrete algorithm for MF-ROM with GPR

For a concrete set-up, we consider a given number of LF and HF evaluations (n_{LF}, n_{HF}) and a set of candidate locations $\Theta_c = \{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, ..., \boldsymbol{\mu}^{(N_c)}\}$, which may be randomly ordered or according to the RRQR pivoting. We propose the following approach: Set Θ_{LF} and Θ_{HF} to the first n_{LF}, n_{HF} elements of Θ_c respectively. For both fidelity levels, evaluate the snapshots at the locations given by Θ and assemble S_{LF} and S_{HF}. In subsection 3.4, we proposed two strategies for assimilating LF data in the general case and we now focus on the implications of each for the MF regression step.

- **"L-proj":** We set $\mathbf{f}(X_{\mathrm{LF}})^{(j)} = \left(V_{\mathrm{LF}}^T S_{\mathrm{LF}}\right)^{(j)}$, according to (27). In the case where $k < k'$, we set the $\mathbf{f}(X_{\mathrm{LF}})^{(k')} = \text{None}$, for $k < j \leq k'$, i.e., we perform single (high) fidelity GPR in the absence of LF data.
- **"LH-interp":** We set $\mathbf{f}(X_{\mathrm{LF}})^{(j)} = V_{\mathrm{HF}}^T S_{\mathrm{LF}\nearrow\mathrm{HF}}^{(j)}$, via the two step bifidelity reconstruction-projection approach described in (30).

The full approach of the offline phase, in which the mapping $\boldsymbol{\pi}(\boldsymbol{\mu}) : \boldsymbol{\mu} \to \mathbf{c}_{\mathrm{HF}}$ is learned, is more compactly described in Algorithm 2. The reconstruction of the actual solutions in the online phase proceeds identically to the single fidelity GPR, c.f. (36), by using the HF predictions.

We already cautioned that the strategy **"L-proj"** may provide unsatisfactory result if the reduced basis space of the LF and HF are too different, whereas **"LH-interp"** avoids this scenario. Unfortunately, **"LH-interp"** may introduce a bias towards the LF trend, when HF solutions are sparse. It is important to bear in mind that we need to use exactly the same k coinciding LF and HF snapshots at locations Θ_k^{HF} for (29) to be applicable. This also implies that for the lifted snapshots $\hat{\mathbf{u}}_{\mathrm{LF}\nearrow\mathrm{HF}}(\boldsymbol{\mu}_i)$ with $\boldsymbol{\mu}_i \in \Theta_k^{\mathrm{HF}}$, i.e., the coinciding snapshots, the interpolation coefficients $\hat{\mathbf{c}}_{\mathrm{LF}}$ are equal to one and the HF and LF input data has identical values:

$$\hat{\mathbf{u}}_{\mathrm{LF}\nearrow\mathrm{HF}}(\boldsymbol{\mu}_i) = \mathbf{u}_{\mathrm{HF}}(\boldsymbol{\mu}_i) \Rightarrow \mathbf{f}_{\mathrm{LF}}(\Theta_k^{\mathrm{HF}})^{(j)} = \mathbf{f}_{\mathrm{HF}}(\Theta_k^{\mathrm{HF}})^{(j)}. \tag{52}$$

Algorithm 2 Two-fidelity GPR for learning the mapping $\boldsymbol{\pi}(\boldsymbol{\mu}) : \boldsymbol{\mu} \to \mathbf{c}_{\mathrm{HF}}$

Input: An ordered set of parameter locations Θ_c, budgets n_{LF} and n_{HF}, the choice of strategy, a suitable covariance kernel.

Output: A GPR model that can be used to predict $\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*) : \boldsymbol{\mu}^* \to \mathbf{c}_{\mathrm{HF}}$

1: Set $\Theta_{\mathrm{LF}} = \Theta_c[: n_{\mathrm{LF}}]$ and $\Theta_{\mathrm{HF}} = \Theta_c[: n_{\mathrm{HF}}]$. Compute the required solutions at Θ_{LF}, Θ_{HF} and assemble S_{LF} and S_{HF}.

2: Compute the respective basis matrices V_{LF} and V_{HF} via POD.

3: Set $X_{\mathrm{LF}} = \Theta_{\mathrm{LF}}$, $X_{\mathrm{HF}} = \Theta_{\mathrm{HF}}$. Set the input data for the GP as $\mathbf{f}(X_{\mathrm{HF}})^{(j)} = \left(V_{\mathrm{HF}}^T S_{\mathrm{HF}}\right)^{(j)}$ and $\mathbf{f}(X_{\mathrm{LF}})^{(j)}$ according to strategy **"L-proj"** or **"LH-interp"**.

4: Train 2-level GPR models $\mathcal{GP}f^{(j)}$ with input data $\mathcal{D}_j$, $\mathcal{D}_j = ([X_{\mathrm{HF}}, X_{\mathrm{LF}}], [\mathbf{f}(X_{\mathrm{HF}})^{(j)}, \mathbf{f}(X_{\mathrm{LF}})^{(j)}])$, for $1 < j \leq \mathrm{rank}(V_{\mathrm{HF}})$, by maximizing the MLE of the hyperparameters.

In the case of $k = n_{\mathrm{HF}} < \mathrm{rank}(S_{\mathrm{LF}})$, we have that $X_{\mathrm{HF}} \cap X_{\mathrm{LF}} = \Theta_k^{\mathrm{HF}}$, i.e., all HF snapshots are part of the interpolation approach and thus:

$$\mathbf{f}_{\mathrm{LF}}(X_{\mathrm{HF}} \cap X_{\mathrm{LF}})^{(j)} = \mathbf{f}_{\mathrm{HF}}(X_{\mathrm{HF}} \cap X_{\mathrm{LF}})^{(j)} \,. \tag{53}$$

This implies that wherever we have HF data, it coincides perfectly with the LF data **by construction** and independently of the actual usefulness of the LF data. This is especially important as the authors of [2] observe that the coinciding parameter locations Θ_k^{HF} are essential for a successful cokriging setup. As a consequence, predictions for the HF model at new parameter locations may be heavily biased towards the LF trend and there may be no correcting effect of the HF data. We thus assess both strategies in our numerical experiments. We further follow these measures to facilitate and stabilize the training process:

- We scale each entry of $\mathbf{c}_{\mathrm{HF}}$ and $\mathbf{c}_{\mathrm{LF}}$ with the corresponding singular value σ from the SVD for the GP input. We thus expect the scaled inputs $\tilde{\mathbf{f}}^{(j)} = \mathbf{f}^{(j)}/\sigma_j$ to be of a magnitude around one.
- Since we use simulated data, we fix $\sigma^2_{\mathrm{noise}}$ of the GP to a value close to machine precision.
- We use isotropic kernels for the HF data, when n_{HF} is small as to avoid under-determination of the hyperparameters and otherwise an automatic relevance determination (ARD) kernel with distinct lengthscales for each parameter dimension. See [22], for a more elaborate discussion of suitable kernel properties for different fidelity levels.

4.4. Active learning

In this section, we propose an active learning procedure that can extend the selection of the HF snapshots, beyond those chosen for the basis via RRQR. We seek to characterize the error of RB-GPR $\mathbf{u}_{\text{RB-GP}}(\boldsymbol{\mu}) = V\boldsymbol{\pi}_{\mathrm{GP}}(\boldsymbol{\mu})$:

$$\varepsilon_{\text{RB-GP}}(\boldsymbol{\mu}) = \|\mathbf{u}_{\mathrm{true}}(\boldsymbol{\mu}) - \mathbf{u}_{\text{RB-GP}}(\boldsymbol{\mu})\|_2 \,. \tag{54}$$

Exploiting the Galerkin orthogonality, we split this error into a contribution of the projection error and a contribution of the regression error:

$$\varepsilon_{\text{RB-GP}}(\boldsymbol{\mu})^2 = \underbrace{\|\mathbf{u}_{\mathrm{true}}(\boldsymbol{\mu}) - \mathbf{u}_{\mathrm{RB}}(\boldsymbol{\mu})\|_2^2}_{\varepsilon^2_{\mathrm{RB}}(\boldsymbol{\mu})} + \underbrace{\|\mathbf{u}_{\mathrm{RB}}(\boldsymbol{\mu}) - \mathbf{u}_{\text{RB-GP}}(\boldsymbol{\mu})\|_2^2}_{\varepsilon^2_{\mathrm{GP}}(\boldsymbol{\mu})} \,. \tag{55}$$

We note that $\varepsilon_{\mathrm{RB}}(\boldsymbol{\mu})$ cannot easily be interpolated via a Gaussian process as it is nonsmooth, zero at selected snapshots for $k \leq \mathrm{rank}(S)$ and strictly positive otherwise. We can, however, construct an error indicator for

the regression error by using the properties of the Gaussian process.

$$\begin{aligned}\varepsilon_{\mathrm{GP}}(\boldsymbol{\mu}) &= \|\mathbf{u}_{\mathrm{RB}}(\boldsymbol{\mu}) - \mathbf{u}_{\text{RB-GP}}\|_2 \\ &= \left\|VV^T\mathbf{u}_{\mathrm{true}}(\boldsymbol{\mu}) - V\boldsymbol{\pi}_{\mathrm{GP}}(\boldsymbol{\mu})\right\|_2 \\ &= \|V(\boldsymbol{\pi}_{\mathrm{RB}}(\boldsymbol{\mu}) - \boldsymbol{\pi}_{\mathrm{GP}}(\boldsymbol{\mu}))\|_2 \\ &= \|\boldsymbol{\pi}_{\mathrm{RB}}(\boldsymbol{\mu}) - \boldsymbol{\pi}_{\mathrm{GP}}(\boldsymbol{\mu})\|_2 \end{aligned} \tag{56}$$

We use a confidence interval to estimate (56) as the GP allows us to quantify the error $\boldsymbol{\varepsilon} = \boldsymbol{\pi}_{\mathrm{RB}}(\boldsymbol{\mu}) - \boldsymbol{\pi}_{\mathrm{GP}}(\boldsymbol{\mu})$ in terms of the variance since the error is normally distributed. We can thus use the standard deviations of the GPs, which are collected in the vector $\boldsymbol{\sigma}_{\mathrm{GP}}$, to indicate the regression error as

$$\varepsilon_{\mathrm{GP}}(\boldsymbol{\mu}) \approx c\|\boldsymbol{\sigma}_{\mathrm{GP}}(\boldsymbol{\mu})\|_2. \tag{57}$$

We note that (57) does not depend on the basis and thus can be evaluated at a cost independent of the underlying LF and HF models. This error indicator can now be used for an active learning scheme, following [12], see Algorithm 3.

Algorithm 3 Active learning scheme

Input: Parameter range $P = [\boldsymbol{\mu}_{\min}, \boldsymbol{\mu}_{\max}]$ and associated parameter pool $\mathcal{P}_c$ of candidate locations, initial guess of l snapshot locations $\Theta_l^{(0)} = \{\boldsymbol{\mu}_i\}_{i=1}^l \subset \mathcal{P}_c$, a budget of LF and HF evaluations n_{LF} and n_{HF}, a budget of additional evaluations n_{add}.

Output: A trained GPR model $\mathcal{GP}f^{(j)}$, $1 \le j \le k$ with a reduced basis V, augmented snapshot locations $\Theta_p^{n_{\mathrm{add}}}$.

1: Build a GPR model using Algorithm 2 with inputs ($\Theta_l^{(0)}$, n_{LF} and n_{HF}). Algorithm 2 also computes initial snapshot matrices $S_{\mathrm{HF}}^{(0)}$ and $S_{\mathrm{LF}}^{(0)}$.
2: **for** i=1 to n_{add} **do**
3: Evaluate the error indicator $\varepsilon_{\mathrm{GP}}(\boldsymbol{\mu})$ in (57) for all candidates $\boldsymbol{\mu}_c \in \mathcal{P}_c$.
4: Choose the location of highest error: $\boldsymbol{\mu}_i = \mathrm{argmax}_{\boldsymbol{\mu}\in\mathcal{P}_c}\varepsilon_{\mathrm{GP}}(\boldsymbol{\mu})$ and add it to the parameter locations $\Theta_{l+i}^{(i)} = \Theta_{l+i-1}^{(i-1)} \cup \{\boldsymbol{\mu}_i\}$.
5: Compute the solution vector $\mathbf{u}_h(\boldsymbol{\mu}_i)$ and add it to the snapshot matrix: $S_{\mathrm{HF}}^{(i)} = \left[S_{\mathrm{HF}}^{(i-1)}, \mathbf{u}_h(\boldsymbol{\mu}_i)\right]$.
6: *Optional: Update the HF basis V_{HF} by computing an SVD on $S_{HF}^{(i)}$: $S_{HF}^{(i)} = U\Sigma Z^T$, $V_{HF} = U[:, :\mathrm{rank}(S_{HF}^{(i)})]$.*
7: Retrain the GPR model with the updated basis V_{HF} and data $\left(\Theta_{l+i}^{(i)}, S_{\mathrm{HF}}^{(i)}\right)$ using Algorithm 2.
8: **end for**

Due to the representer theorem [1, 30], the regression error equals zero at already explored points: $\varepsilon_{\mathrm{GP}}(\boldsymbol{\mu}) = 0$, which this error indicator accurately reflects, so that a new point is explored in each step. For a stable approach, we propose to initialize Algorithm 3 with $\Theta^{(0)} = \Theta_k$, i.e., the anchoring points of the basis snapshots. As the projection error tends to decay much faster than the regression error [12, 17], it is reasonable to choose k small and update the HF basis on the fly as proposed in Algorithm 3, expecting that the basis will converge with the regression. This idea proves straightforward in the single fidelity setup. However, it does not carry over as easily for the MF setup with **"LH-interp"**: Since the error indicator is not sensitive to the actual structure of the underlying PDE, it does not guarantee that the snapshots of the selected points are actually linearly independent. As a consequence, the Gramian in (28) might not be invertible. To avoid this situation, we choose $k = \min\left(\mathrm{rank}(S_{\mathrm{HF}}), \mathrm{rank}(S_{\mathrm{LF}})\right)$.

5. Numerical examples

In this section, we assess the performance of the developed methods on typical examples. We begin with manufactured examples, consider large deformations in structural dynamics, and finally analyze a fluid-structure interaction problem in its steady-state.

5.1. Example 1: A manufactured case

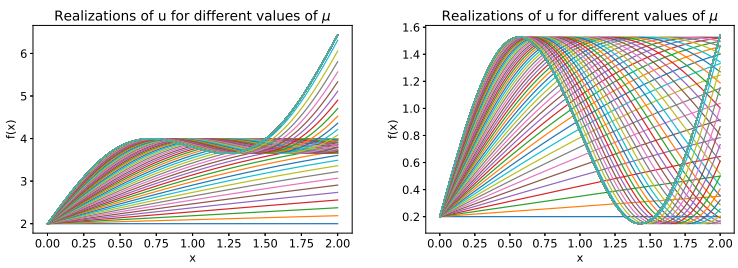
To test the proposed algorithms, we introduce a manufactured example with a limited region of importance in the parameter space. To visualize the results, we define a one-dimensional problem with one parameter:

$$\boldsymbol{u}_{\mathrm{LF}}(\mu; x) = \sin(g(\mu)2\pi x) + g(\mu)x + 0.2 \cdot \exp(g(\mu)x) \tag{58}$$

$$\boldsymbol{u}_{\mathrm{HF}}(\mu; x) = \sin(g(\mu)2\pi x) + g(\mu)x + 2 \cdot \exp(g(\mu)x) \tag{59}$$

$$g(\mu) = \left\{ \begin{array}{ll} \mu, & \text{if } 0 \leq \mu \leq 0.5 \\ 0.5, & \text{otherwise} \end{array} \right\} \tag{60}$$

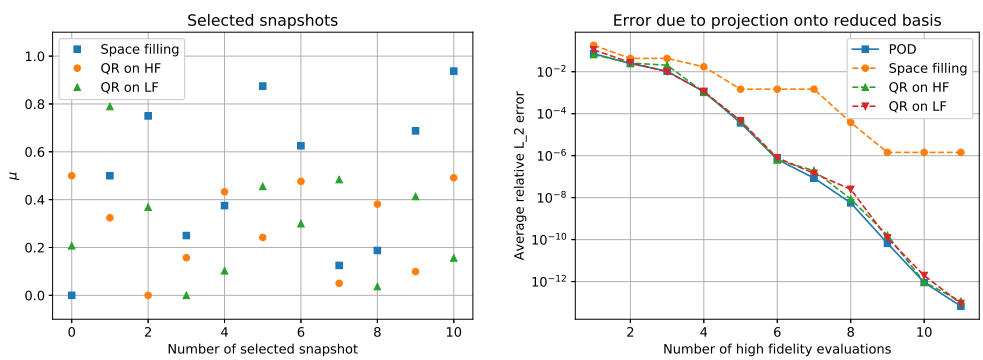
for $x \in \Omega_X = [0, 2]$ and $\mu \in \Omega_\mu = [0, 1]$. Figure 2 shows some realizations of the functions for different



(a) high-fidelity: $\mathbf{u}_{\mathrm{HF}}$ (b) low-fidelity: $\mathbf{u}_{\mathrm{LF}}$

Figure 2: Plots of possible solutions for **Example 1**.

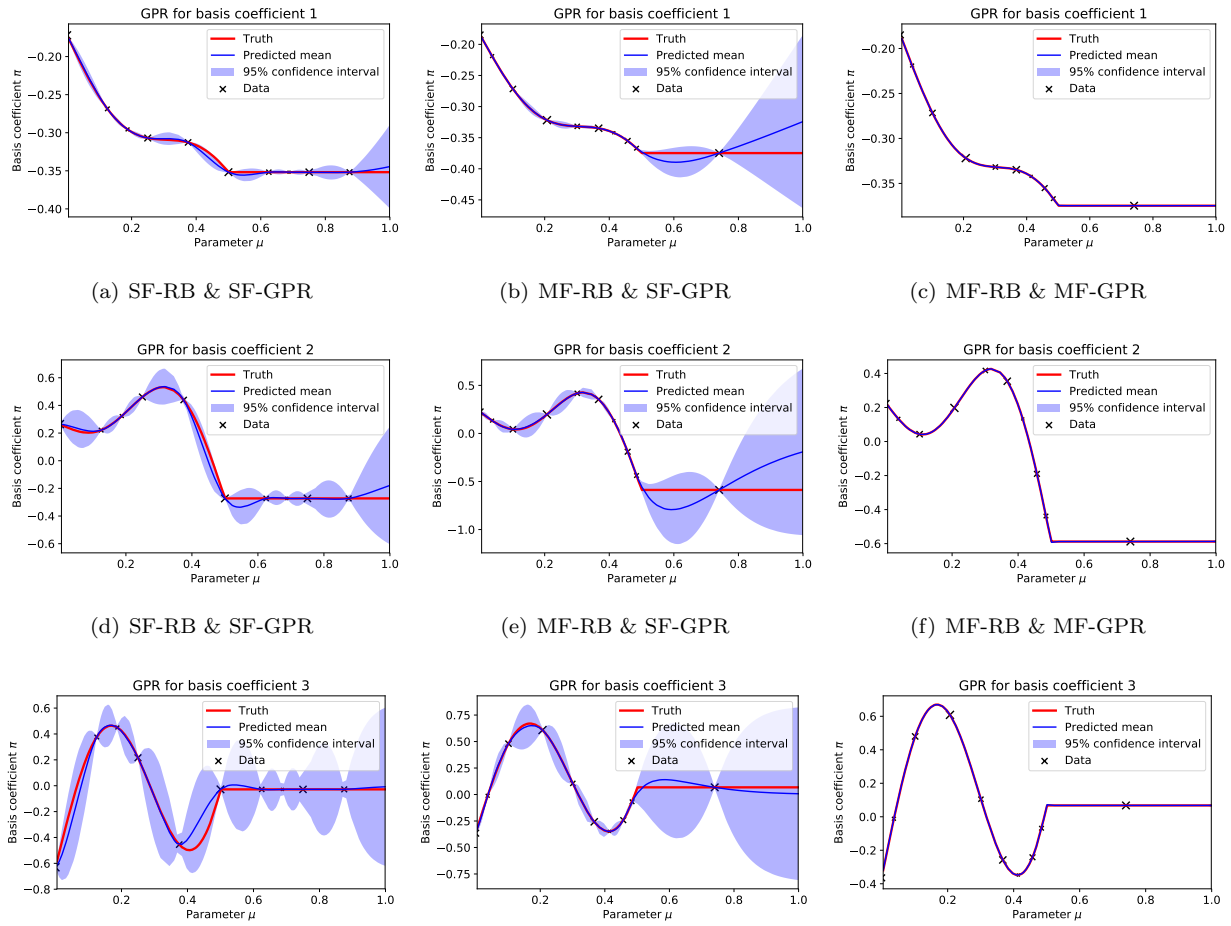
parameter values μ. In Figure 3, we observe that the RRQR algorithm correctly chooses snapshots in the critical region $\mu = [0, 0.5]$ for both the HF and the LF model.



(a) Selected snapshots (b) Projection error

Figure 3: Performance of the bifidelity basis for **Example 1**: Different snapshots are selected with the RRQR on the LF and HF model, however both selections allow for a reduced basis with similarly fast decay in the projection error. The projection error of a basis constructed from space filling snapshots decays much more slowly.

We confirm that the bifidelity basis with the RRQR on the LF performs much better than randomly choosing points in the parameter domain. In Figure 4, we plot the GPR results for the RB coefficients. We notice that the snapshots selected with the RRQR are concentrated where the RB coefficient varies, so that the constant part is not approximated well for the single fidelity regression, whereas the space-filling points lead to an overall average approximation. The MF-GPR with "LH-interp" and 160 LF snapshots captures the parametric dependence perfectly. The error convergence (Figure 5) confirms this trend. We note that the strategy "L-proj" does not improve the results over single fidelity (SF)-GPR, whereas the active learning scheme succeeds in reducing the error much faster than randomly chosen points.



(a) SF-RB & SF-GPR (b) MF-RB & SF-GPR (c) MF-RB & MF-GPR

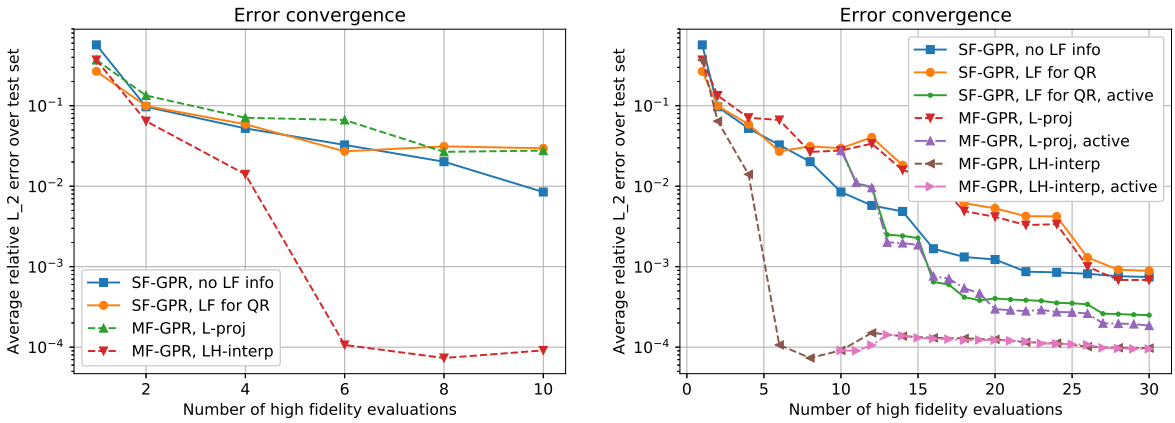
(d) SF-RB & SF-GPR (e) MF-RB & SF-GPR (f) MF-RB & MF-GPR

(g) SF-RB & SF-GPR (h) MF-RB & SF-GPR (i) MF-RB & MF-GPR

Figure 4: GPR for the RB coefficients for **Example 1** with 10 HF snapshots. An ARD Matern kernel with smoothness parameter $\nu = 3/2$ is used. For the results on the left [(a), (d), (g)] a pure HF approach with snapshots by a space-filling sequence are used to construct the basis and data for regression: SF-RB & SF-GPR. For the results in the middle [(b),(e), (h)], HF snapshots and the basis are chosen according to Algorithm 1 and used in single fidelity GPR: MF-RB & SF-GPR. The results on the right [(b),(e), (h)] use the both the MF basis and MF GPR, i.e., both Algorithm 1 and 2, with "LH-interp" and 160 LF snapshots: MF-RB & MF-GPR.

5.2. Example 2: 2D - structural analysis on a frame

Next, we consider the previously introduced frame problem with three parameters. Figure 6 shows the considered parameter domain and the first three HF RB basis functions for this setup. The Young modulus in the remaining components is fixed to 210 GPa, and the Poisson ratio is $\nu = 0.3$.



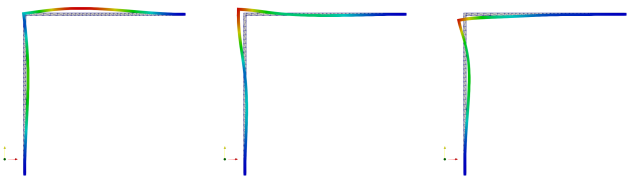
(a) First part: Snapshots chosen via QR

(b) Second part: Extension beyond QR snapshots

Figure 5: Convergence of the error for **Example 1**, computed over 200 randomly chosen test snapshots. To select HF snapshots beyond the rank of the LF, we choose either a space-filling sequence or an active learning scheme with an error indicator.

Parameter	**Range** $[\mu_{\min}, \mu_{\max}]$	**Unit**
$\mu_1 = F$	[0.3, 1.0]× 6.5e6	N/m
$\mu_2 = E_{\text{Column}}$	[0.3, 1.3] × 210	GPa
$\mu_3 = E_{\text{Corner}}$	[0.7,1.3] × 210	GPa

(a) Parameters



(b) Basis 1 (c) Basis 2 (d) Basis 3

Figure 6: Parameter range and the first three corresponding RB functions for the 2D frame problem.

For UQ applications, we focus on two QoIs:

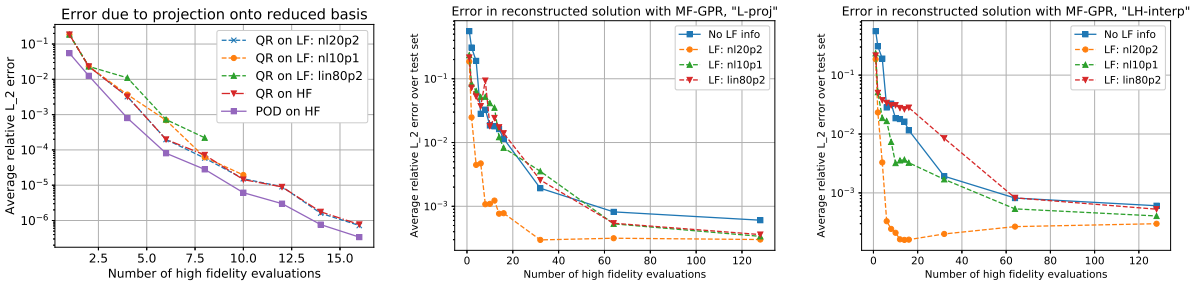
- The displacement in the y direction at the location where the force pushes onto the frame.
- The maximum von Mises stress in the structure.

To simulate a realistic scenario, we sample from a truncated normal distribution, centred on the parameter space under consideration. Even though the proposed method allows it, we do not consider correlations between the parameters. The parameters are sampled according to

$$\mu_i \sim \mathcal{N}(m, \sigma, \text{cut-off}) \quad \text{for } 1 \leq i \leq 3, \tag{61}$$

with mean $m = \mu_{\min} + 0.5\mu_{\max}$, standard deviation $\sigma = 0.5(m - \mu_{\min})$ and a cut-off of 2 standard deviations, so that $\mu_i \in [\mu_{\min}, \mu_{\max}]$ as defined in Figure 6.

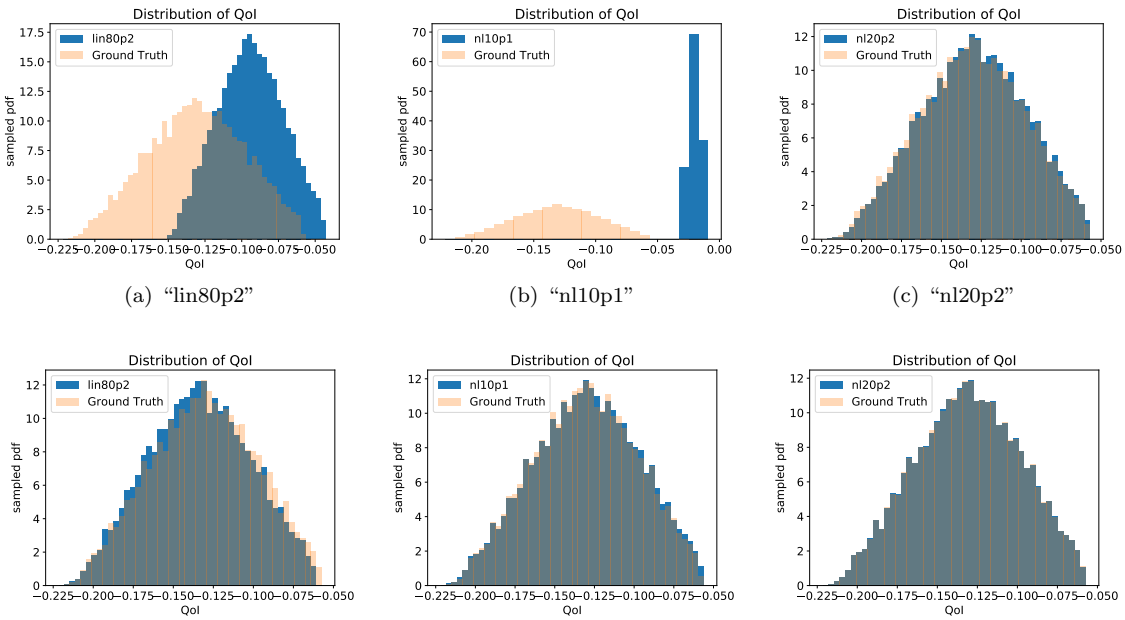
In Figure 7 (a), we observe convergence of the projection error with a gap between the POD and the RRQR basis. We note that the LF models exhibit lower rank, the linearized model ("lin80p2") in particular has half as many distinct modes as the HF model. Figure 7 (b),(c) depict the error in solution recovery for both proposed strategies. Unsurprisingly, the best LF model gives the most improvement over SF-GPR, while the other two LF models provide much smaller gains and even lead to worse approximations in one case ("lin80p2" with "LH-interp"). Interestingly, the strategy "L-proj" seems capable of leveraging all three LF models in the presence of a high number of HF snapshots, i.e., it may also provide benefits outside of the "small data" regime.



(a) Projection error (b) Error of MF-GPR (c) Error of MF-GPR

Figure 7: Error of the reduced basis and the 2-level GPR for **Example 2**. A RBF kernel with ARD properties is used when more than 6 HF snapshots are available, otherwise the kernel is isotropic. 160 LF snapshots are used for each MF model.

Figure 8 assesses the match in the distribution of the y-displacement and compares the direct use of the LF models with a bifidelity approach on 6 HF snapshots. The very coarse model "nl10p1" shows most improvements, whereas the linearized model is biased in both direct and bifidelity use. For succesfull MF-GPR, it thus seems more important that the LF model incorporates the correct physics than recovering a close solution in an engineering sense.

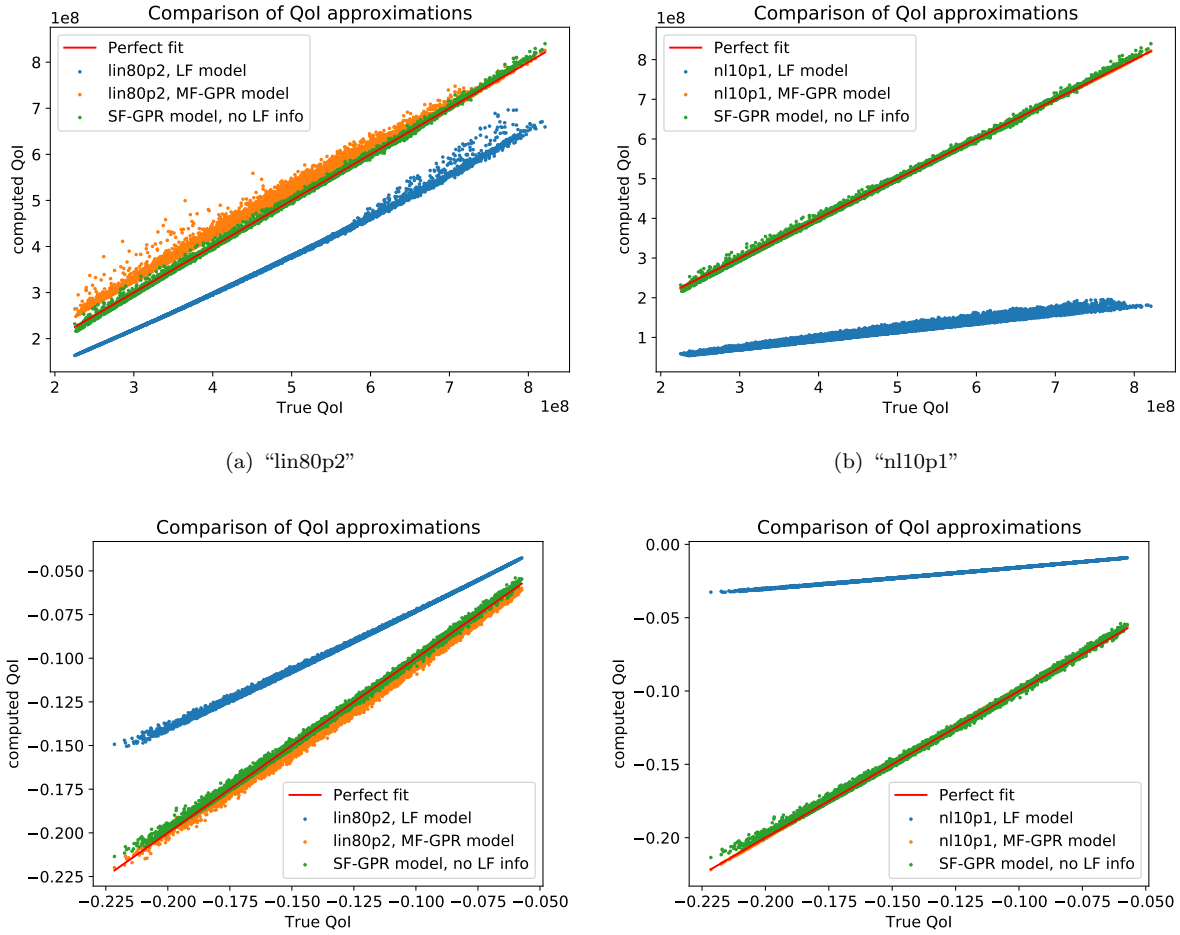


(a) "lin80p2" (b) "nl10p1" (c) "nl20p2"

(d) "lin80p2" (e) "nl10p1" (f) "nl20p2"

Figure 8: Distributions of the QoI "y-displacement" for **Example 2**. For the top row (a), (b), (c) the solution is obtained from the LF model directly, the bottom row (d), (e), (f) uses a surrogate model with MF -GPR regression on 6 HF and 160 LF snapshots with the strategy "LH-interp" and an RBF kernel. The ground truth is obtained with direct simulation on the HF model.

Figure 9 confirms this observation for both the maximum stress and the y-displacement. While the linearized model is closer to the true QOI than the coarse model, its use in MF-GPR does not bring benefits over SF-GPR. In contrast, the MF-GPR with "nl10p1" gives a perfect fit and outperforms SF-GPR on 6 HF snapshots.



(a) "lin80p2"

(b) "nl10p1"

(c) "lin80p2"

(d) "nl10p1"

Figure 9: Recovery of the QoIs for **Example 2** and fit with the ground truth of the HF model computed for different LF and surrogate models. The plots on the left use the linearized LF model, the plots on the right use the coarse nonlinear LF model. Top: maximum stress, bottom: y-displacement at the attack point of the force. 160 LF snapshots are used for the MF-GPR with the strategy "LH-interp".

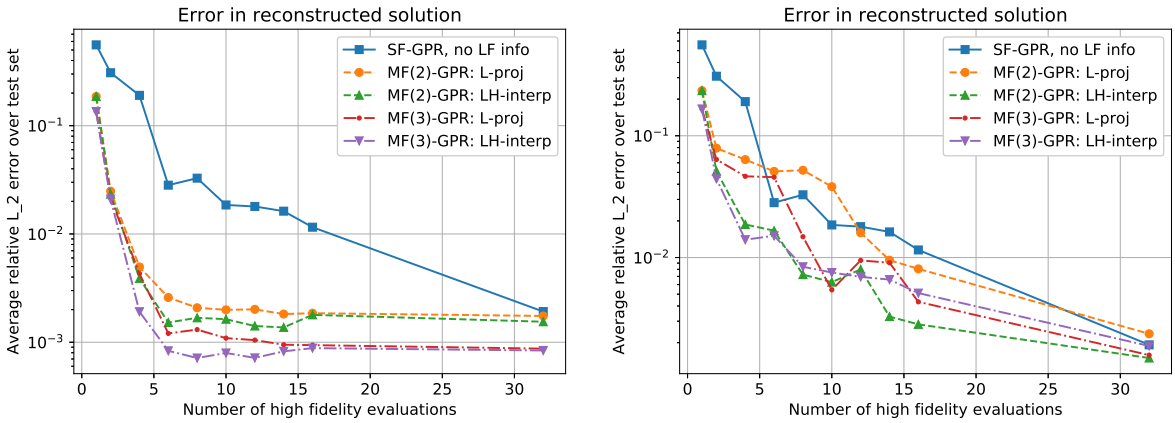
Three-fidelity levels

As the best LF model "nl20p2" is also the most expensive, this section explores how a 3-level model can further reduce simulation costs. We consider

- a hierarchical model which uses 40 snapshots from "nl20p2" as a medium fidelity data and 300 snapshots from "nl10p1" as LF data and
- a semi-hierarchical model which uses 40 snapshots from "nl10p1" and 300 snapshots from "lin80p2" as LF data. This model thus combines the linear high resolution model and the nonlinear low resolution model without imposing a hierarchy between the LF models.

Figure 10 (a) shows the results of the hierarchical model ("MF(3)-GPR"), compared to a single fidelity model ("SF-GPR, no LF info") and a bifidelity model ("MF(2)-GPR"), which only uses the 40 medium fidelity snapshots.

For both assimilation strategies, we observe a significant improvement of the 3-level model as compared to the 2-level model. The semi-hierarchical model does not combine the best of two worlds as hoped for the



(a) Hierarchical 3-level model

(b) Semi-hierarchical 3-level model

Figure 10: Performance of the 3-level models for solution recovery for **Example 2** as compared to SF-GPR and 2-level models.

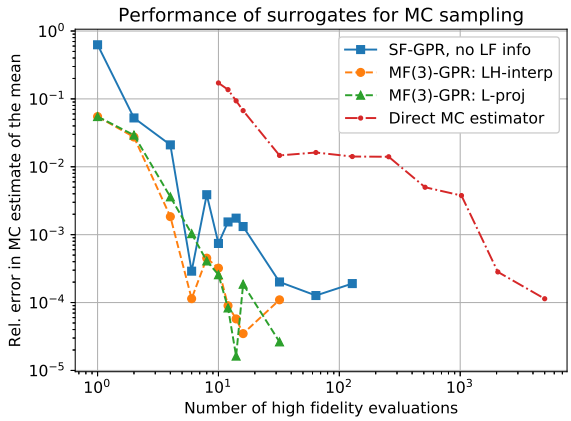
strategy "LH-interp", see Figure 10 (b). While the strategy "L-proj" appears to benefit slightly from the additional information, it gives worse approximations than the bifidelity approach "LH-interp". We thus focus on the purely hierarchical model for the UQ applications. Figure 11 shows much faster error decay for MF-GPR compared to a direct MC estimator of the mean. We also observe that the surrogate models can give stable estimates of event probabilities with only 6 HF snapshots, while the MC estimate requires up to 100 HF snapshots to reach a rough approximation.

5.3. Example 3: 2D - fluid-structure interaction

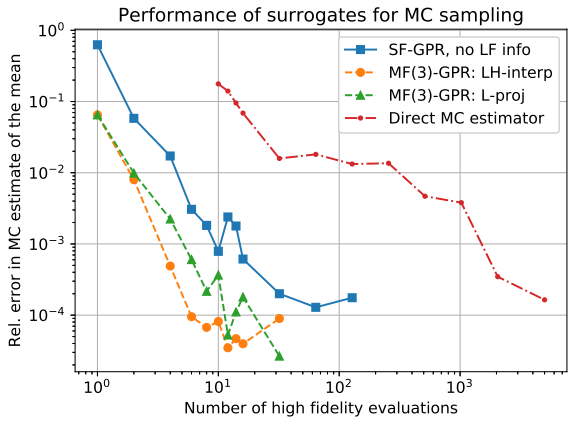
As a third example, we consider a benchmark fluid-structure interaction (FSI) between an elastic object and a laminar incompressible flow. We refer to [10] for a precise description of the problem setup. We use the implementation in [21], which uses a monolithic geometric convective explicit approach with a semi-implicit fluid and a nonlinear structural implementation, and parametrize the in-flow velocity and the stiffness of the beam. The parameter domain Ω_μ is chosen so that the FSI solution converges to a steady state within 10 seconds of simulation time. The benchmark focuses on 4 QoIs: The x- and y-displacement at the beam tip and the lift and drag on the beam. The HF model has 82169 degrees of freedom (dofs) in total, which can be compared to level 3 of the benchmark, which exhibits relative errors in the range of 1e-4 to 1e-3 as compared to the highest resolution benchmark. We create a LF model with only 9847 dofs by using a coarser mesh and p1 elements for the fluid part. This reduction in DOFs with a factor ≈ 10 significantly reduces the simulation time. As demonstrated during our numerical studies, the LF model yields average errors of around 50% for the y-displacement and around 10% for the other 3 QoIs, compared to the HF model. In spite of this, the proposed MF-GPR algorithm can benefit from the LF information.

As the magnitudes of the different solutions fields do not differ greatly, we stack the velocity, displacement and pressure field to obtain exactly one (steady-state) snapshot per parameter vector $\boldsymbol{\mu}$. An ARD Matérn kernel with smoothness parameter $\nu = 3/2$ and 80 LF snapshots are used for the GPR if not stated otherwise.

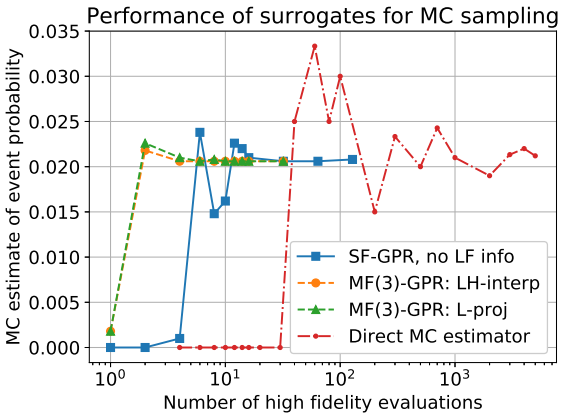
In Figure 12 (a)-(d), MF-GPR with a single HF snapshot suffices to approximate the solution field so closely that there are no visual differences. Figure 12 (e)-(g) show the first basis vectors for the velocity in x-direction for the parameter domain in Figure 12 (h). In Figure 13 (a), we see slight improvement in the projection error, when a snapshot selection scheme is used as compared to a space filing sequence. Regarding error in the solution field, Figure 14 (b), we note faster error convergence for both MF strategies when compared to SF GPR. The active learning scheme does not yield improvement as the error has already stagnated at a low level below 1e-4. In Figure 14, we observe that the error in specific QoIs decays slower than the error in the solution field. Consequently more HF snapshots are required for satisfactory results.



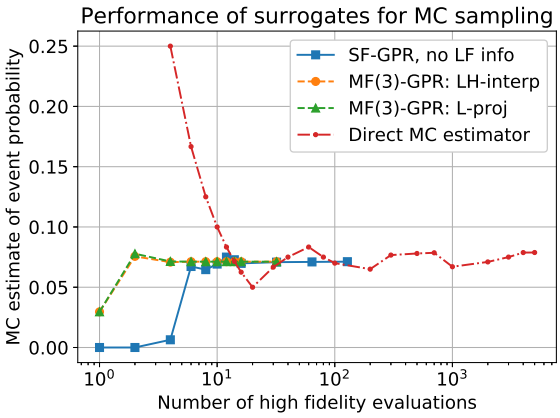
(a) QoI: maximum stress



(b) QoI: y-displacement at the applied force



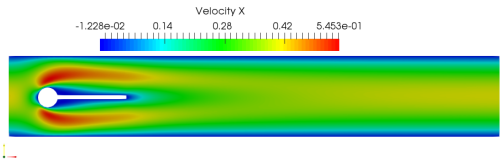
(c) P(maximum stress> 7.5e8Pa)



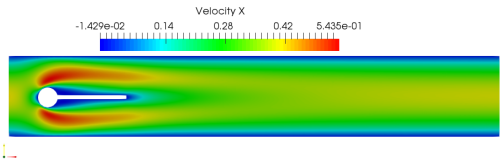
(d) P(y-displacement $>$ 0.18m)

Figure 11: UQ application for the two QoIs of **Example 2**. The top row shows the error in the MC estimate of the mean of the maximum stress (a) and the y-displacement (b), the reference MC estimate was obtained on 10000 i.i.d samples. The bottom row shows the MC events for the given event probabilities (c), (d).

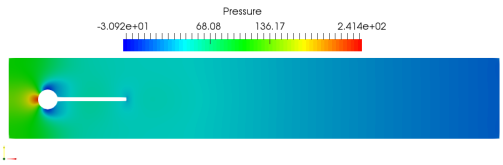
We note a benefit of MF-GPR over SF-GPR for all four QoIs, notably, the strategy "LH-interp" can leverage the LF information more than "L-proj", which aligns with our theoretical observations. Figure 15 shows that MF-GPR with only 6 and 10 HF snapshots performs better than both SF-GPR and the direct use of the LF model. This demonstrates that the proposed technique leverages LF models that are too inaccurate for direct use to improve the HF surrogate model efficiently.



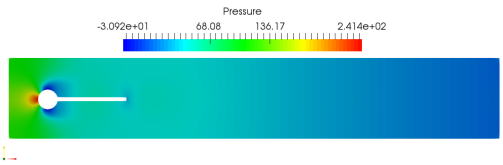
(a) HF solution



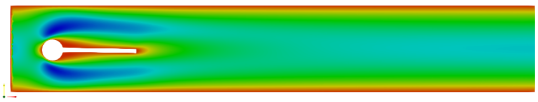
(b) MF-GPR solution with 1 HF snapshot and 80 LF snapshots



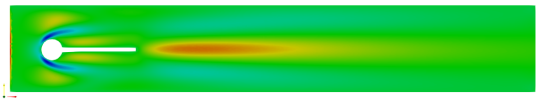
(c) HF solution



(d) MF-GPR solution with 1 HF snapshot and 80 LF snapshots



(e) Basis 1



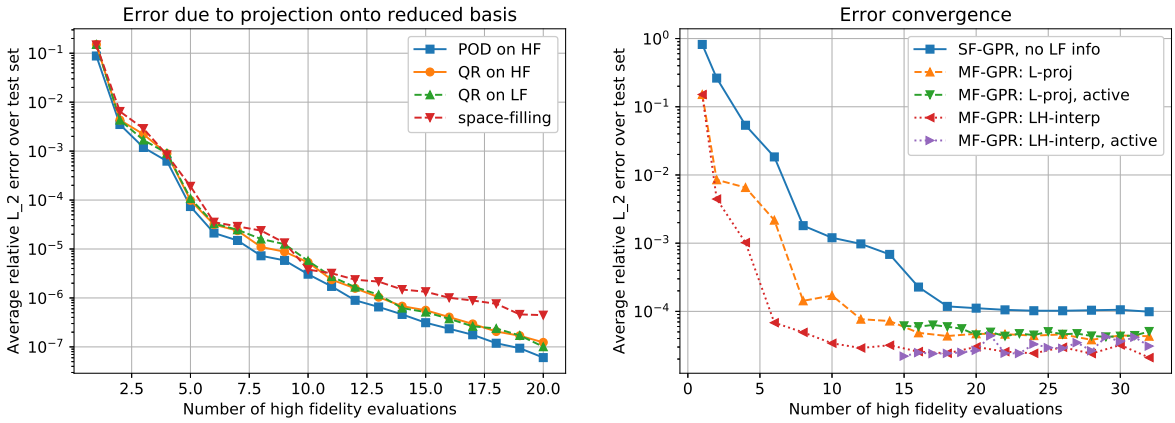
(f) Basis 2

(g) Basis 3

Parameter	**Range** $[\mu_{\min}, \mu_{\max}]$
$\mu_1 = u_{\text{inlet}}$	$[0.5,\ 1.5]\times\ 0.2$
$\mu_2 = E_{\text{Beam}}$	$[0.5,\ 1.5]\ \times\ 1.4\text{e}6$

(h) Parameters

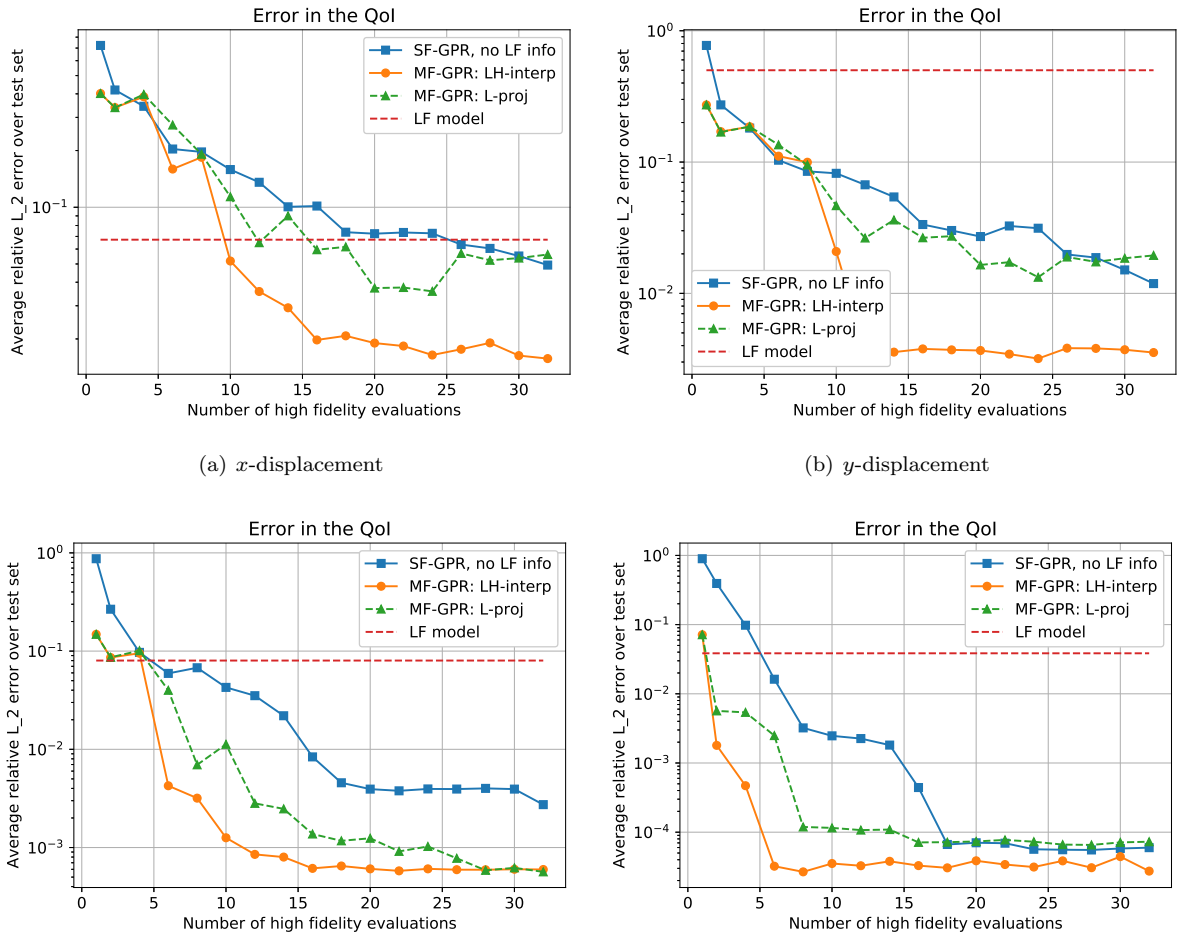
Figure 12: Visualization of **Example 3**. (a) and (c) show the HF solution for the pressure and x-velocity for $\boldsymbol{\mu} = [0.27, 1.90]^T$, (b) and (d) show the MF-GPR reconstruction of the same fields. (e),(f), (g) depict the first three basis vectors for the velocity field and (h) the parameter domain.



(a) Projection error

(b) Error in solution field

Figure 13: Convergence of the error for **Example 3** computed over a grid of 225 test snapshots. To select HF snapshots beyond the rank of the LF, we choose either a space-filling sequence or an active learning scheme with an error indicator.



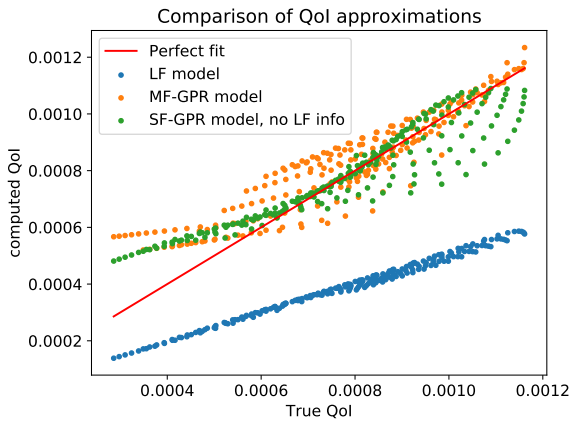
(a) x-displacement

(b) y-displacement

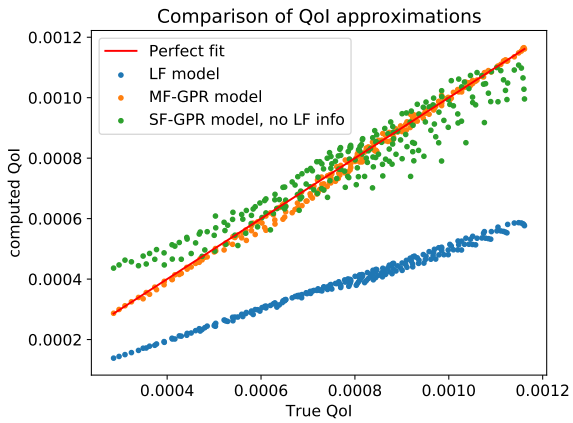
(c) Lift

(d) Drag

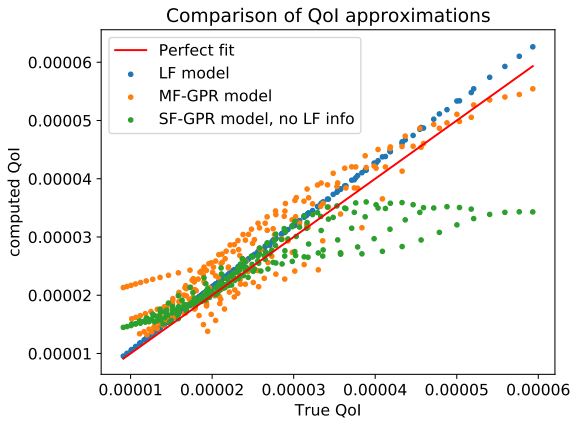
Figure 14: Convergence of the error for the four QoIs in **Example 3**, computed over a grid of 225 test snapshots.



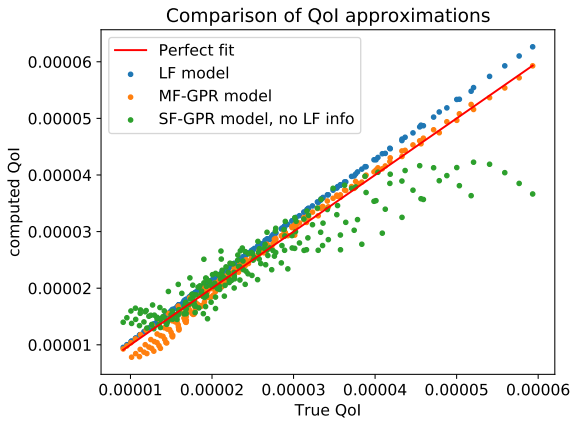
(a) y-displacement, 6 HF snapshots



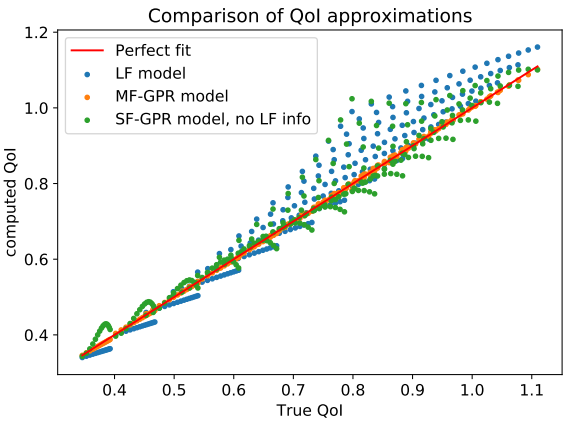
(b) y-displacement, 10 HF snapshot



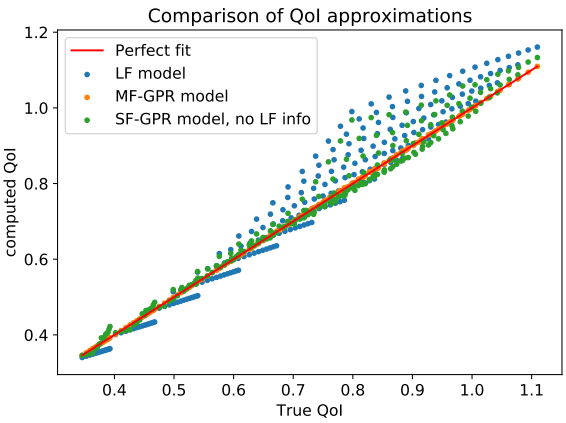
(c) x-displacement, 6 HF snapshots



(d) x-displacement, 10 HF evaluations



(e) Drag, 6 HF snapshots



(f) Drag, 10 HF snapshots

Figure 15: Fit between the approximated QoIs and the HF QoIs for **Example 3**. 6 HF snapshots are used in the models for the plots on the left, 10 HF snapshots on the right. From top to bottom, we consider the y-displacement of the tip, the corresponding x-displacement and the drag on the beam.

6. Conclusions

We have demonstrated the feasibility of a non-intrusive RB method for parametrized nonlinear PDEs that can leverage models of different fidelity to accurately recover both the full solution field and specific QoIs for UQ applications. The method extracts parameter locations from a collection of low-fidelity (LF) snapshots for the efficient creation of a high-fidelity (HF) reduced basis and employs multifidelity Gaussian process regression (MF-GPR) to approximate the coefficients of the reduced model. The basis selection relies on a rank revealing QR decomposition of a low fidelity snapshot matrix.

For the three examples under consideration, we confirm that the proposed MF surrogate approach yields performance gains over a single fidelity (SF) surrogate and methods that directly use the LF model. For all solution fields, we achieve relative errors below 1e-3 with very few HF snapshots. Overall, the construction of a good RB space appears to require less snapshots than a reliable MF-GPR regression. We also observe greater benefit for low-resolution LF models as compared to reduced-physics LF models. It appears that, for a useful LF model, accurately incorporating the properties of the PDE is more important than accurately recovering the values of the solution field: even underresolved, low accuracy LF models can greatly improve the MF surrogate, as long as the properties of the PDE are preserved. It would be interesting to investigate if such conclusions hold for other applications, e.g. in fluid dynamics, where reduced physics models are commonly used.

The proposed active learning scheme discovers snapshot locations that further reduce the error of the MF surrogate and thus enables the addition of HF data points beyond those chosen on the LF model. Interestingly, MF-GPR models can give benefits over SF-GPR models, even when a significant amount of HF data is available. Future work could thus leverage this additional HF information to exploit more complicated correlation structures, which use e.g. nonlinear or spatially varying correlation kernels, with the goal of obtaining an even more accurate MF surrogate model.

Acknowledgment

M. Guo and J. S. Hesthaven acknowledge the support of the Swiss Commission for Technology and Innovation (CTI) under Grant No. 25964.2PFIW-IW.

References

[1] M. A. Alvarez, L. Rosasco, N. D. Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

[2] A. Bertram, C. Othmer, and R. Zimmermann. Towards real-time vehicle aerodynamic design via multi-fidelity data-driven reduced order modeling. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0916, 2018.

[3] J. Bleyer. Numerical tours of continuum mechanics using FEniCS. 2018.

[4] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge university press, 2nd edition, 2008.

[5] E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

[6] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. SIAM, 2009.

[7] P. Businger and G. H. Golub. Handbook Series Linear Algebra. *Numerische Mathematik*, 7(044):269–276, 1965.

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

[9] K. K. Choi and N.-H. Kim. *Structural sensitivity analysis and optimization 1: linear systems*. Springer Science & Business Media, 2006.

[10] FeatFlow. Featflow website. `http://www.featflow.de/en/benchmarks/cfdbenchmarking/fsi_benchmark.html`, 2019.

[11] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

[12] M. Guo and J. S. Hesthaven. Reduced order modeling for nonlinear structural analysis using gaussian process regression. *Computer Methods in Applied Mechanics and Engineering*, 341:807–826, 2018.

[13] M. Guo and J. S. Hesthaven. Data-driven reduced order modeling for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 345:75–99, 2019.

[14] J. Hampton, H. R. Fairbanks, A. Narayan, and A. Doostan. Practical error bounds for a non-intrusive bi-fidelity approach to parametric / stochastic model reduction. *Journal of Computational Physics*, 368:315–332, 2018.
[15] Z.-H. Han, S. Görtz, and R. Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, 25(1):177–189, 2013.
[16] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations.* Springer, 2016.
[17] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
[18] W. M. Lai, D. H. Rubin, E. Krempl, and D. Rubin. *Introduction to continuum mechanics.* Butterworth-Heinemann, 2009.
[19] R. E. Melchers and A. T. Beck. *Structural reliability analysis and prediction.* John Wiley & Sons, 2018.
[20] A. Narayan, C. Gittelson, and D. Xiu. A stochastic collocation algorithm with multifidelity models. *SIAM Journal on Scientific Computing*, 36(2):A495–A521, 2014.
[21] F. Negri. redbKIT Version 2.2. `http://redbkit.github.io/redbKIT/`, 2016.
[22] A. O'Hagan and M. C. Kennedy. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.
[23] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.
[24] B. Peherstorfer, T. Cui, Y. Marzouk, and K. Willcox. Multifidelity importance sampling. *Computer Methods in Applied Mechanics and Engineering*, 300:490–509, 2016.
[25] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.
[26] E. Qian, B. Peherstorfer, D. O'Malley, V. Vesselinov, and K. Willcox. Multifidelity monte carlo estimation of variance and sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):683–706, 2018.
[27] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations.* Springer, 2016.
[28] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox. Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 2018.
[29] E. Ullmann and I. Papaioannou. Multilevel estimation of rare events. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):922–953, 2015.
[30] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press, 2006.
[31] Z. Zhang, M. Guo, and J. S. Hesthaven. Model order reduction for large-scale structures with local nonlinearities. *Infoscience preprint*, 2018.
[32] X. Zhu, E. M. Linebarger, and D. Xiu. Multi-fidelity stochastic collocation method for computation of statistical moments. *Journal of Computational Physics*, 341:386–396, 2017.

Appendix: GPy implementation of the three level autoregressive model

The implementation of the LCM in the Python package GPy considers each entry of the W matrices as independent. For the 2-level cokriging case, we can easily impose the structure by fixing parts of the matrices W to a certain value. However, the interdependence between W_2 and W_3 given in (49) is more complex and the implementation in GPy does not simply allow to link the parameters of different kernels together, i.e., for the formulation in (48), GPy treats the factor $\rho_1\rho_2$ in W_3 as an independent hyperparameter, which introduces unnecessary complexity to the model. To impose the nested structure of 3-level cokriging in the GPy package, which only has two degrees of freedom ρ_1 and ρ_2, we rewrite the formulation in (48). Let

$$M = \begin{bmatrix} \rho_1^2 & \rho_1 & \rho_1 \\ \rho_1 & 1 & 1 \\ \rho_1 & 1 & 1 \end{bmatrix} = W_{\text{nested}} W_{\text{nested}}^T, \text{ so that } W_{\text{nested}} = \begin{bmatrix} \rho_1 \\ 1 \\ 1 \end{bmatrix}. \tag{A.1}$$

We then factor out the ρ_1 term to obtain $\tilde{W}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, $\tilde{W}_3 = \begin{bmatrix} \rho_2 \\ \rho_2 \\ 1 \end{bmatrix}$. This finally allows us to rewrite (48) as

$$K(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = W_1 W_1^T k_1(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + M \circ \left(\tilde{W}_2 \tilde{W}_2^T k_2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + \tilde{W}_3 \tilde{W}_3^T k_3(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) \right), \tag{A.2}$$

where $\circ$ denotes the entrywise matrix Hadamard product. In this way, the hyperparameter ρ_1 only appears in W_{nested} and ρ_2 only appears in $\tilde{W}_3$. It is then straightforward to extend the original coregionalization kernel class to model the kernel $\tilde{W}_3$ and one only needs to rewrite the definition of the gradient in terms of

ρ_2 by using the chain rule:

$$\frac{\partial K}{\partial \rho_2} = \frac{\partial K}{\partial W_3}\frac{\partial W_3}{\partial \rho_2} = \left[\frac{\partial K}{\partial W_3}\right]_1 + \left[\frac{\partial K}{\partial W_3}\right]_2 .$$