Reduced order modeling for nonlinear structural analysis using Gaussian process regression

Mengwu Guo*, Jan S. Hesthaven

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

Abstract

A non-intrusive reduced basis (RB) method is proposed for parametrized nonlinear structural analysis undergoing large deformations and with elasto-plastic constitutive relations. In this method, a reduced basis is constructed from a set of full-order snapshots by the proper orthogonal decomposition (POD), and the Gaussian process regression (GPR) is used to approximate the projection coefficients. The GPR is carried out in the offline stage with active data selection, and the outputs for new parameter values can be obtained rapidly as probabilistic distributions during the online stage. Due to the complete decoupling of the offline and online stages, the proposed non-intrusive RB method provides a powerful tool to efficiently solve parametrized nonlinear problems with various engineering applications requiring multi-query or real-time evaluations. With both geometric and material nonlinearities taken into account, numerical results are presented for typical 1D and 3D examples, illustrating the accuracy and efficiency of the proposed method.

Keywords: Reduced basis method, nonlinear structural analysis, proper orthogonal decomposition, Gaussian process regression, machine learning

1. Introduction

Models expressed as *parametrized nonlinear partial differential equations* are widely used in structural engineering [17, 29]. In such models, parameters are defined to characterize material properties, loads, geometric features, boundary conditions and so on. Especially in the context of *multi-query* or *real-time* structural analysis, such as structural optimization [11], reliability analysis [25], real-time updating [21] and parameter estimation [8], it is required to solve the system for many parameter values.

The rapid development of computer-aided engineering (CAE) and simulation science during the past several decades has enabled high-fidelity simulations for complex engineering structures, for which finite element methods (FEMs) [37, 38] are the most popular tools and have been widely studied and used. In spite of the increasing computational power, high-fidelity simulations are still too expensive to allow multi-query or real-time problems, as a large amount of degrees of freedom (DOFs) are required to accurately solve a problem, implying great demands on both CPU time and memory. Due to some intrinsic similarities among the solutions at different parameter values, on the other side, repeatable high-fidelity calculations for varying parameters are potentially wasting substantial computational resource. To address this issue, reduced order modeling (ROM) has been extensively developed for decades, aiming at reducing the computational cost with a controlled loss of accuracy. The key idea of ROM is to replace the full-order system with a carefully constructed reduced-order model with much smaller dimension, to reduce memory needs and CPU time.

The *reduced basis* (RB) *method* [17, 28, 29, 31] is a powerful and widely used technique for ROM, carried out in an offline-online framework. In the *offline* stage, an RB space, with a significantly smaller dimension

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

than the full-order problem, is spanned by a set of RB functions carefully extracted from a set of high-fidelity snapshots obtained at specific parameter locations. The two major approaches for such extraction are the *Greedy algorithm* [10, 28] and the *proper orthogonal decomposition* (POD) [22, 29], of which the former selects a subset of snapshots as basis functions according to some optimality criterion, while the latter employs a singular value decomposition (SVD) to the collection of snapshots to recover the RB functions. Once the RB space is constructed, the approximated solution for a desired new parameter value is sought *online* as a linear combination of the RB functions. A Galerkin projection is often employed to determine the combination coefficients, and referred to as the Galerkin-projection-based approach for the online stage.

The success of the RB method in decreasing computational cost relies on the decoupling of the offline and online stages, ensuring that the online computation is independent of the dimension of the full-order model. For a general nonlinear structural problem with non-affine dependence on parameters, however, such a full decoupling is often not possible. The assembly of the reduced problem is directly embodied online, and both the configuration updating and the nonlinear iteration require the full-order model, which leads to a reduced efficiency. The empirical interpolation method (EIM) [1] and its discrete variants [9, 27] have been proposed to recover an affine expansion of the differential operator in a non-affine case. However, such schemes are problem-dependent and of an *intrusive* nature, and is often not practical for complex nonlinear problems.

In this paper, a non-intrusive RB method is proposed for nonlinear structural analysis. After extracting the RB functions from a set of snapshots by POD, a regression-based approach [18] is used to establish a mapping from parameter values to projection coefficients onto the RB space. A complete decoupling of offline and online stages is ensured by the regression-based approach, as the online solutions only require direct outputs from the reduce-order regression model that is trained offline. As an important part of *machine learning* [4, 26], regression methods have been intensively developed in *supervised learning*. Among the existing regression models, the proposed regression-based RB method employs a *Gaussian process regression* (GPR), which infers that the observed input-output pairs follow a priori of Gaussian process, and then makes predictions for new parameter values according to the posteriori. Equipped with an active data selection for the training samples, the efficiency of the GPR can be further enhanced. Numerical results also indicate that the GPR model shows good performance in both accuracy and efficiency of ROM's for nonlinear structural analysis.

Following the introduction, the basic equations of nonlinear structural analysis are briefly reviewed in Section 2. In Section 3, the regression-based RB method is presented and the procedure is specified. After an introduction to the key ideas of GPR, application of GPR to the ROM for structural problems is addressed in Section 4, with an active data selection algorithm proposed to enhance efficiency. In Section 5, the method is tested and validated by two examples of large deformation analysis, one in 1D and the other in 3D. Finally, some conclusions are drawn in Section 6.

For the clarity of the notation, italic bold symbols are adopted in this paper for coordinates, vector fields and tensor fields, such as coordinates $\boldsymbol{X}$, displacement vector field $\boldsymbol{u}$, strain tensor field $\boldsymbol{E}$ and stress tensor field $\boldsymbol{S}$; and upright bold symbols are used for vectors and matrices in linear algebra, such as regression input vector $\mathbf{x}$, collection of outputs $\mathbf{y}$, observed input matrix $\mathbf{X}$, discrete displacement solution $\mathbf{u}_h$ and the snapshot matrix $\mathbf{S}$.

2. Nonlinear structural analysis

2.1. Governing equations

In solid mechanics, the governing equations can often be split into three types: the equations of kinematics to describe the motion and deformation of the structure, the equations of equilibrium to describe the balance between stresses in the deformable body and external loads applied on it, and the equations of constitutive laws to specify the relation between stresses and strains for the material. In the following, a general deformable body is assumed to experience large deformation and a nonlinear constitutive response in a Cartesian coordinate system.

Kinematics

The motion of a deformable body is recorded as configurations. As shown in Figure 1, the original coordinates of the body are denoted as $\boldsymbol{X} \in \bar{\Omega}$, where $\bar{\Omega}$ is the undeformed configuration that the body occupies at time $t = 0$, and $\Omega \subset \mathbb{R}^3$ is the corresponding domain. $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X}, t) \in \bar{\omega}_t \subset \mathbb{R}^3$ reflects the coordinates at time $t > 0$ in the deformed configuration $\bar{\omega}_t$, which can also be considered as a vector field defined in $\bar{\Omega}$ with respect to the original coordinates $\boldsymbol{X}$. Note that $\omega_0 = \Omega$. The vector field of the displacement can be naturally defined as

$$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X}\,, \tag{1}$$

which will act as the unknowns of the nonlinear problems in this work. Moreover, it should be pointed out that only quasi-static problems are taken into account in this work, i.e. the problems are assumed to be time-independent.

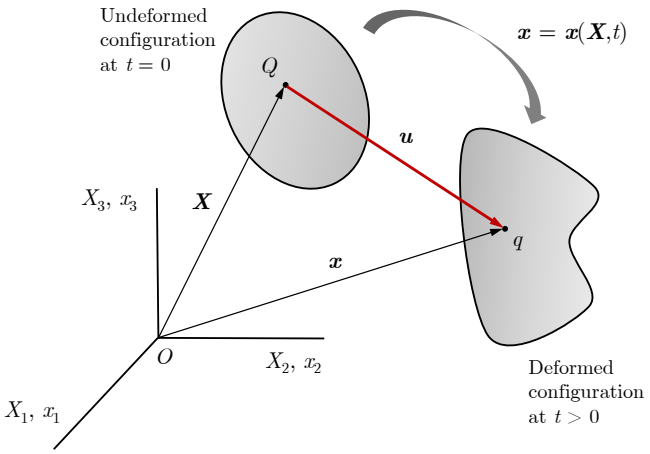


Figure 1: Motion of a deformable body

To evaluate the deformation of the body, a tensor field $\boldsymbol{F} : \bar{\Omega} \to \mathbb{M}^3$, referred to as the *deformation gradient* tensor, is introduced as

$$\boldsymbol{F} = \boldsymbol{\nabla}_{\boldsymbol{X}} \boldsymbol{x} = \boldsymbol{I} + \boldsymbol{\nabla}_{\boldsymbol{X}} \boldsymbol{u}\,, \tag{2}$$

where $\mathbb{M}^3$ stands for the set of all real square matrices of order 3, $\boldsymbol{I}$ is the unit tensor and $\boldsymbol{\nabla}_{\boldsymbol{X}}$ is the gradient operator with respect to the original coordinates $\boldsymbol{X}$. Then the *Green-Lagrange strain* tensor field $\boldsymbol{E} : \bar{\Omega} \to \mathbb{S}^3 := \left\{ \boldsymbol{B} \in \mathbb{M}^3 : \boldsymbol{B} = \boldsymbol{B}^{\mathrm{T}} \right\}$ is defined as

$$\boldsymbol{E}(\boldsymbol{u}) = \frac{1}{2}\left[\boldsymbol{F}(\boldsymbol{u})^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{u}) - \boldsymbol{I}\right] = \frac{1}{2}\left[\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u} + (\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u})^{\mathrm{T}} + (\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u})^{\mathrm{T}}\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\right]\,, \tag{3}$$

which can also be considered as a nonlinear operator acting on the displacement field $\boldsymbol{u}$. The displacement boundary condition on the Dirichlet boundary $\Gamma_D \subset \partial\Omega$ is prescribed as $\boldsymbol{u} = \boldsymbol{0}$.

Remark 1: In this paper, only problems with homogeneous Dirichlet boundary conditions are discussed, as problems with inhomogeneous Dirichlet boundary conditions $\boldsymbol{u} = \boldsymbol{u}_D$ on Γ_D can be transformed to the homogeneous case, i.e. one can define $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{p}$ with $\boldsymbol{p} \in [C^\infty(\bar{\Omega})]^3$ being a predefined function that satisfies the boundary conditions $\boldsymbol{p} = \boldsymbol{u}_D$ on Γ_D, and solve for $\boldsymbol{w}$ by replacing $\boldsymbol{u}$ with $\boldsymbol{w} + \boldsymbol{p}$ in the problem.

Equilibrium

The *second Piola-Kirchhoff stress* field $\boldsymbol{S}$ is usually used for large deformation analysis, and the corre-

sponding equation of equilibrium is given as

$$\boldsymbol{\nabla}_{\boldsymbol{X}} \cdot (\boldsymbol{F}(\boldsymbol{u})\boldsymbol{S}) + \boldsymbol{b} = \boldsymbol{0} \quad \text{in } \Omega\,, \tag{4}$$

where $\boldsymbol{b}$ is a prescribed body force applied on the structure with respect to the undeformed volume. Moreover, the traction boundary condition on Neumann boundary $\Gamma_N = \partial\Omega \setminus \Gamma_D$ is introduced as $(\boldsymbol{F}(\boldsymbol{u})\boldsymbol{S})^{\mathrm{T}}\hat{\boldsymbol{n}} = \boldsymbol{t}$, with $\boldsymbol{t}$ being the prescribed traction with respect to the undeformed surface area and $\hat{\boldsymbol{n}}$ is the unit outward normal vector along Γ_N.

Correspondingly, the weak formulation for the equations of equilibrium, also referred to as the virtual work principle [2, 7], can be given as:

$$\int_\Omega \boldsymbol{S} : D\boldsymbol{E}[\boldsymbol{u}](\boldsymbol{v})\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{b}^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{t}^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Gamma \tag{5}$$

for all $\boldsymbol{v} : \bar{\Omega} \to \mathbb{R}^3$ that are smooth enough and vanish on Γ_D. Here $D\boldsymbol{E}[\boldsymbol{u}](\boldsymbol{v})$ stands for the Gâteaux derivative of $\boldsymbol{E}$ at $\boldsymbol{u}$ in the direction $\boldsymbol{v}$ and can be expressed explicitly as

$$D\boldsymbol{E}[\boldsymbol{u}](\boldsymbol{v}) = \frac{1}{2}[\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{v} + (\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{v})^{\mathrm{T}} + (\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{v})^{\mathrm{T}}\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u} + (\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u})^{\mathrm{T}}\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{v}]. \tag{6}$$

Constitutive relation

Generally, the constitutive relation of the material can be expressed as a nonlinear operator $\boldsymbol{\mathcal{C}}$ which maps a tensor field of the Green-Lagrange strain to the corresponding tensor field of second Piola-Kirchhoff stresses, i.e.

$$\boldsymbol{S} = \boldsymbol{\mathcal{C}}(\boldsymbol{E})\,. \tag{7}$$

In this work, two typical nonlinear constitutive relations in structural analysis, hyperelasticity and elasto-plasticity, are considered. We refer to [6, 35] for more details about them and a variety of other constitutive relations.

2.2. Nonlinear structural problems and their parametrization

Combining the governing equations in Subsection 2.1, one has the following weak formulation of a nonlinear structural problem: find $\boldsymbol{u} \in \mathcal{V}$ such that

$$\int_\Omega \boldsymbol{\mathcal{C}}\left(\boldsymbol{E}(\boldsymbol{u})\right) : D\boldsymbol{E}[\boldsymbol{u}](\boldsymbol{v})\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{b}^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{t}^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Gamma \quad \forall \boldsymbol{v} \in \mathcal{V}\,, \tag{8}$$

where $\mathcal{V} = \{\boldsymbol{v} : \bar{\Omega} \to \mathbb{R}^3$ smooth enough, $\boldsymbol{v} = \boldsymbol{0}$ on $\Gamma_D\}$, as mentioned after (5).

As is well known, the two major types of nonlinearities in structural analysis are material nonlinearities and geometric nonlinearities, which can be recognized from the variational form (8). The former is the nonlinearity in $\boldsymbol{\mathcal{C}}(\boldsymbol{E}(\boldsymbol{u}))$ with respect to $\boldsymbol{u}$, describing the nonlinear mechanical behaviour of the material, and the latter is caused by the fact that the Gâteaux derivative $D\boldsymbol{E}[\boldsymbol{u}]$ depends on $\boldsymbol{u}$, as is a natural result for large deformations.

Remark 2: To ensure regularity, the solution $\boldsymbol{u}$ to a nonlinear structural problem is considered to belong to Sobolev space $[W^{2,p}(\Omega)]^3$ with some $p > 3$, see [12] for example. In this paper, such considerations are irrelevant to the proposed reduced order modelling techniques, so the smoothness of solutions is merely described as 'smooth enough' for simplification.

Usually, finite elements are adopted to discretize the problem and an incremental formulation is employed for applying external loads onto the structure. Within a loading increment, iterative algorithms, such as the Newton-Raphson algorithm and the arc-length method [2, 7], can be used to solve the nonlinear algebraic/discrete equations.

The total Lagrangian formulation is used as an incremental formulation in this work. In the solution scheme, discrete equations are formulated and updated with respect to the undeformed configuration. The

$(k{+}1)$th Newton-Raphson iteration step in the loading increment j finds the displacement increment $\delta\boldsymbol{u}_h^{k+1} \in \mathcal{V}_h$ such that

$$\begin{aligned}\int_\Omega D\boldsymbol{\mathcal{C}}[\boldsymbol{E}({}^j\boldsymbol{u}_h^k)](D\boldsymbol{E}[{}^j\boldsymbol{u}_h^k](\delta\boldsymbol{u}_h^{k+1})) : D\boldsymbol{E}[{}^j\boldsymbol{u}_h^k](\boldsymbol{v})\,\mathrm{d}\Omega + \int_\Omega \boldsymbol{\mathcal{C}}(\boldsymbol{E}({}^j\boldsymbol{u}_h^k)) : \left[(\boldsymbol{\nabla}_{\boldsymbol{X}}\delta\boldsymbol{u}_h^{k+1})^{\mathrm{T}}\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{v}\right]\mathrm{d}\Omega \\ = \int_\Omega {}^j\boldsymbol{b}^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Omega + \int_{\Gamma_N} {}^j\boldsymbol{t}^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Gamma - \int_\Omega \boldsymbol{\mathcal{C}}(\boldsymbol{E}({}^j\boldsymbol{u}_h^k)) : D\boldsymbol{E}[{}^j\boldsymbol{u}_h^k](\boldsymbol{v})\,\mathrm{d}\Omega \quad \forall\boldsymbol{v}\in\mathcal{V}_h\end{aligned} \tag{9}$$

and updates the displacements as ${}^j\boldsymbol{u}_h^{k+1} = {}^j\boldsymbol{u}_h^k + \delta\boldsymbol{u}_h^{k+1} \in \mathcal{U}_h$, where ${}^j\boldsymbol{u}_h^k$ is the displacement field after k iterations in the loading increment j, ${}^j\boldsymbol{b}$ and ${}^j\boldsymbol{t}$ are the loads applied after j increments, and $\mathcal{V}_h \subset \mathcal{V}$ is the discrete counterpart of $\mathcal{V}$. The initial condition for loading increment j is given as ${}^j\boldsymbol{u}_h^0 = {}^{j-1}\boldsymbol{u}_h$ with ${}^{j-1}\boldsymbol{u}_h$ being the updated displacements after $j-1$ increments. Moreover, the final displacement field, obtained by finite element analysis after the loading increments, is denoted by $\boldsymbol{u}_h$ for simplification, i.e. ${}^j\boldsymbol{u}_h = \boldsymbol{u}_h$ when ${}^j\boldsymbol{b} = \boldsymbol{b}$ and ${}^j\boldsymbol{t} = \boldsymbol{t}$.

In this work, physical parametrization is taken into account. Several parameters are considered for some characteristics in the constitutive relation and the external loads. Then the parametrized nonlinear problem, corresponding to (8), is given as: for any given parameter $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^d$,

$$\int_\Omega \boldsymbol{\mathcal{C}}(\boldsymbol{E}(\boldsymbol{u}(\boldsymbol{\mu})); \boldsymbol{\mu}) : D\boldsymbol{E}[\boldsymbol{u}(\boldsymbol{\mu})](\boldsymbol{v})\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{b}(\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{t}(\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{v}\,\mathrm{d}\Gamma \quad \forall\boldsymbol{v}\in\mathcal{V}\,, \tag{10}$$

where $\mathcal{P}$ is the parameter domain and d is the total number of parameters. For cases with parameters in the geometry, [19, 24] can be referred to for details. After a transformation to the parameter-independent reference domain, the treatment will be similar to the strategy for physical parameters.

3. The reduced basis method for nonlinear structural analysis

Due to the use of an incremental formulation and iteration algorithms, solving the parametrized nonlinear problem (10) by a finite element discretization requires the assembly and solution of a number of linear systems. The dimension of such linear systems, denoted by N_h and referred to as the number of degrees of freedom (DOFs), is determined by both the underlying mesh and the polynomial order that the finite element analysis employs. The high-fidelity solution to a real-world structural problem often requires a large number of DOFs and many steps of increments and iterations, implying that the full-order model is expensive. Thus a direct numerical approximation of the full-order model is not affordable in many-query or real-time context of parametrized nonlinear structural analysis.

The reduced basis (RB) method is proposed as an efficient and convenient tool for model order reduction. It seeks the approximate solution to a parameterized problem in a reduced space spanned by a set of parameter-independent RB functions, constructed from a collection of high-fidelity snapshots at different parameter values. The RB functions are either carefully chosen from the snapshots by the Greedy algorithm [10, 28, 31], or by principal ingredient analysis of snapshots. The former requires an error estimator/indicator for the full-order solution, and picks the snapshot that maximizes the estimator/indicator as a new RB function until a criteria is satisfied. However, proper error estimators or indicators for a general nonlinear structural problem are unknown, so the latter approach is utilized with the aid of the proper orthogonal decomposition (POD) [17, 22, 29], as detailed in the following.

To evaluate the reduced-order solution for any desired value in the parameter domain, a regression-based approach will be introduced to parametrized nonlinear structural analysis, rather than the conventional Galerkin-projection-based approach.

3.1. Full-order solutions and snapshots

The notion of a solution manifold can be introduced, comprising all the solutions of the parametrized problem (10) under variation of the parameters, i.e.

$$\mathcal{M} = \{\boldsymbol{u}(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\} \subset \mathcal{V}\,. \tag{11}$$

Since the exact solutions are not available, a discrete counterpart of $\mathcal{M}$ can be considered as

$$\mathcal{M}_h = \{\boldsymbol{u}_h(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\} \subset \mathcal{V}_h \,, \tag{12}$$

where $\boldsymbol{u}_h(\boldsymbol{\mu})$ is the high-fidelity full-order solution obtained by finite element analysis, i.e.

$$\boldsymbol{u}_h(\boldsymbol{X};\boldsymbol{\mu}) = \sum_{i=1}^{N_h} (\mathbf{u}_h(\boldsymbol{\mu}))_i \boldsymbol{\phi}_i(\boldsymbol{X}) \,. \tag{13}$$

Here N_h is the number of DOFs, $\mathbf{u}_h(\boldsymbol{\mu})$ is an N_h-dimensional vector collecting all the values of the DOFs, and $\boldsymbol{\phi}_i$ is the ith basis/shape function. Note that the finite element space $\mathcal{V}_h$ is spanned by all the shape functions, i.e. $\mathcal{V}_h = \text{span}\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_{N_h}\}$. The discrete solution $\mathbf{u}_h(\boldsymbol{\mu})$ for any parameter $\boldsymbol{\mu}$ is calculated under a fixed finite element setting.

To generate an RB space for the nonlinear problem, one considers a collection of N_s *snapshots* $\{\boldsymbol{u}_h(\boldsymbol{\mu}^1), \boldsymbol{u}_h(\boldsymbol{\mu}^2), \cdots, \boldsymbol{u}_h(\boldsymbol{\mu}^{N_s})\}$ associated with a discrete point-set $\Theta = \{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \cdots, \boldsymbol{\mu}^{N_s}\} \subset \mathcal{P}$ in the parameter domain. Then a subspace of $\mathcal{V}_h$ can be spanned by the snapshots as

$$\mathcal{M}_\Theta = \text{span}\{\boldsymbol{u}_h(\boldsymbol{\mu}^1), \boldsymbol{u}_h(\boldsymbol{\mu}^2), \cdots, \boldsymbol{u}_h(\boldsymbol{\mu}^{N_s})\} \subset \mathcal{V}_h \,. \tag{14}$$

The discrete point-set Θ is either a uniform lattice or a collection of generated points over the parameter domain $\mathcal{P}$. If Θ is fine enough, $\mathcal{M}_\Theta$ can act as a good representation of $\mathcal{M}_h$.

To reduce the model, a low-rank approximation $\mathcal{V}_{\text{rb}}$ with rank $L \ll \min\{N_h, N_s\}$ should be found for $\mathcal{M}_\Theta$. Towards this end, the POD is employed in this work to extract RB functions $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_L\}$ from snapshots and then span the RB space $\mathcal{V}_{\text{rb}}$ as

$$\mathcal{V}_{\text{rb}} = \text{span}\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_L\} \,, \tag{15}$$

as detailed in next subsection.

3.2. The proper orthogonal decomposition and the reduced basis space

Consider a *snapshot matrix* $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$ collecting the DOFs of all snapshots, i.e

$$\mathbf{S} = \left[\, \mathbf{u}_h(\boldsymbol{\mu}^1) \mid \mathbf{u}_h(\boldsymbol{\mu}^2) \mid \cdots \mid \mathbf{u}_h(\boldsymbol{\mu}^{N_s}) \, \right] . \tag{16}$$

In the context of nonlinear structural analysis, it is assumed that the number of snapshots is less than that of DOFs, i.e. $N_s \ll N_h$, to avoid a high cost of preparing full-order snapshots.

The POD takes advantage of the *singular value decomposition* (SVD) of matrix $\mathbf{S}$, given as

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^{\mathrm{T}} \tag{17}$$

with $\mathbf{U} \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{Z} \in \mathbb{R}^{N_s \times N_s}$ being orthogonal matrices, i.e. $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}_{N_h}$ and $\mathbf{Z}^{\mathrm{T}}\mathbf{Z} = \mathbf{I}_{N_s}$, and $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_{N_s}\}$ containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{N_s} \geq 0$.

Defined as a subspace of $\mathbb{R}^{N_h}$ spanned by all the N_s columns of $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$, the column space of $\mathbf{S}$ is denoted by $\text{Col}(\mathbf{S})$. At the algebraic level, one seeks to find the 'best' approximation of $\text{Col}(\mathbf{S})$, in some optimal sense, among all L-dimensional subspaces with $L \leq \text{rank}(\mathbf{S})$. Let $\mathbf{V} \in \mathbb{R}^{N_h \times L}$ denote the first L columns of $\mathbf{U}$, and let $\mathbb{Y}_L = \{\mathbf{W} \in \mathbb{R}^{N_h \times L} : \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}_L\}$ represent the set of all L-dimensional orthogonal bases. The projection error of snapshots onto orthogonal bases $\mathbf{W} \in \mathbb{Y}_L$ in the Euclidean norm can be expressed as $\sum_{i=1}^{N_s} \|\mathbf{u}_h(\boldsymbol{\mu}^i) - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}^i)\|^2_{\mathbb{R}^{N_h}}$.

The Schmidt-Eckart-Young theorem [14, 29, 32] states that the basis consisting of the first L left singular vectors of $\mathbf{S}$ minimizes the projection error of snapshots among all the L-dimensional orthogonal bases in

$\mathbb{R}^{N_h}$, and the error can be evaluated by the $(L+1)$th to N_sth singular values, i.e.

$$\sum_{i=1}^{N_s} \left\| \mathbf{u}_h(\boldsymbol{\mu}^i) - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}^i) \right\|_{\mathbb{R}^{N_h}}^2 = \min_{\mathbf{W}\in\mathbb{Y}_L} \sum_{i=1}^{N_s} \left\| \mathbf{u}_h(\boldsymbol{\mu}^i) - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}^i) \right\|_{\mathbb{R}^{N_h}}^2 = \sum_{i=L+1}^{N_s} \sigma_i^2 \,. \tag{18}$$

Hence the relative error, corresponding to the minimized projection error, is defined as

$$\frac{\sum_{i=1}^{N_s} \left\| \mathbf{u}_h(\boldsymbol{\mu}^i) - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}^i) \right\|_{\mathbb{R}^{N_h}}^2}{\sum_{i=1}^{N_s} \|\mathbf{u}_h(\boldsymbol{\mu}^i)\|_{\mathbb{R}^{N_h}}^2} = \frac{\sum_{i=L+1}^{N_s} \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} \,. \tag{19}$$

Thus one obtains that $\mathrm{Col}(\mathbf{S})$ can be well approximated by $\mathrm{Col}(\mathbf{V})$ with a small L if the singular values decay rapidly.

The procedure of the POD is then given as the following algorithm:

Algorithm 1 POD

Input: Snapshot matrix $\mathbf{S}$, projection error tolerance ϵ_{POD}
Output: Reduced rank L, matrix $\mathbf{V}$ collecting the RB

1: Form the correlation matrix $\mathbf{M} = \mathbf{S}^{\mathrm{T}}\mathbf{S} \in \mathbb{R}^{N_s \times N_s}$;
2: Solve the eigenvalue problem for $\mathbf{M}$, i.e. $\mathbf{M}\mathbf{x}_i = \sigma_i^2 \mathbf{x}_i$, $i = 1, 2, \cdots, N_s$;
3: Set $\mathbf{v}_i = \frac{1}{\sigma_i}\mathbf{S}\mathbf{x}_i$, $i = 1, 2, \cdots, \mathrm{rank}(\mathbf{S})$;
4: Define $L \leq \mathrm{rank}(\mathbf{S})$ as the minimum integer s.t. $\frac{\sum_{i=1}^{L} \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} > 1 - \epsilon_{\mathrm{POD}}^2$;
5: Define $\mathbf{V} = [\ \mathbf{v}_1 \mid \mathbf{v}_2 \mid \cdots \mid \mathbf{v}_L\ ]$.

The desired rank L can also be defined directly, rather than determined by the tolerance ϵ_{POD}.

Corresponding to the approximation of $\mathrm{Col}(\mathbf{S})$ by $\mathrm{Col}(\mathbf{V})$ on the algebraic level, function space $\mathcal{M}_\Theta$ is hence approximated by $\mathcal{V}_{\mathrm{rb}} = \mathrm{span}\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_L\}$ with the RB functions $\boldsymbol{\psi}_l$ defined as

$$\boldsymbol{\psi}_l = \sum_{k=1}^{N_h} V_{kl}\boldsymbol{\phi}_k \,, \quad l = 1, 2, \cdots, L \,. \tag{20}$$

It is noticed that there exists a biunique correspondence between the elements in $\mathcal{V}_{\mathrm{rb}}$ and those in $\mathrm{Col}(\mathbf{V})$, i.e. for any $\mathbf{w}_L \in \mathbb{R}^L$

$$\begin{aligned} \mathbf{w}_{\mathrm{rb}} := \mathbf{V}\mathbf{w}_L \in \mathrm{Col}(\mathbf{V}) \quad \Leftrightarrow \quad \boldsymbol{w}_{\mathrm{rb}} &= \sum_{k=1}^{N_h} (\mathbf{w}_{\mathrm{rb}})_k \boldsymbol{\phi}_k = \sum_{k=1}^{N_h} \left(\sum_{l=1}^{L} V_{kl} (\mathbf{w}_L)_l \right) \boldsymbol{\phi}_k \\ &= \sum_{l=1}^{L} (\mathbf{w}_L)_l \left(\sum_{k=1}^{N_h} V_{kl}\boldsymbol{\phi}_k \right) = \sum_{l=1}^{L} (\mathbf{w}_L)_l \boldsymbol{\psi}_l \in \mathcal{V}_{\mathrm{rb}} \,. \end{aligned} \tag{21}$$

3.3. Regression-based approach for reduced-order solutions

The numerical procedure of the RB method is efficiently carried out in an *offline-online* framework. As discussed, the RB functions are prepared from the high-fidelity snapshots in the parameter-independent offline stage. The reduced-order solution for a new parameter is then sought in the online stage. The Galerkin-projection-based approach is the most often used for this, i.e. the problem for a new parameter value is solved in the RB space $\mathcal{V}_{\mathrm{rb}}$ by a standard Galerkin approach.

However, the Galerkin-projection-based scheme will not significantly save computational cost for a general nonlinear structural problem. In addition to compromising the efficiency due to the non-affinity in parameter dependence, the structural configuration and matrix assembly have to be updated during all the loading increments and iterations when solving nonlinear algebraic equations. Moreover, there may exist convergence

or updating issues in some complex cases due to the possibility that some configurations in the incremental procedure are not represented well in $\mathcal{V}_{\rm rb}$.

Therefore, a regression-based approach is proposed to calculate reduced-order solutions for new parameters. In this scenario, the projection of a full-order discrete solution $\mathbf{u}_h(\boldsymbol{\mu})$ onto $\mathrm{Col}(\mathbf{V})$ acts as the corresponding reduced-order solution at algebraic level,

$$\mathbf{u}_{\rm rb}(\boldsymbol{\mu}) = \mathbf{V}\mathbf{V}^{\rm T}\mathbf{u}_h(\boldsymbol{\mu}) = \arg\min_{\mathbf{w}_h \in \mathrm{Col}(\mathbf{V})} \|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{w}_h\|_{\mathbb{R}^{N_h}} \,, \tag{22}$$

in which $\mathbf{V}^{\rm T}\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{u}_L(\boldsymbol{\mu})$ collects the coefficients associated with column bases of $\mathbf{V}$.

To obtain the projection coefficients $\mathbf{u}_L(\boldsymbol{\mu})$ for any desired parameter $\boldsymbol{\mu} \in \mathcal{P}$, one can resort to a nonlinear regression $\hat{\boldsymbol{\pi}}$ between $d = \dim(\mathcal{P})$ inputs and L outputs:

$$\boldsymbol{\mu} \mapsto \mathbf{u}_L(\boldsymbol{\mu}) = \mathbf{V}^{\rm T}\mathbf{u}_h(\boldsymbol{\mu}) \approx \hat{\boldsymbol{\pi}}(\boldsymbol{\mu})\,. \tag{23}$$

This regression model $\hat{\boldsymbol{\pi}}(\cdot)$ should be constructed from a set of training data $\mathcal{D} = \{(\boldsymbol{\mu}_i, \mathbf{V}^{\rm T}\mathbf{u}_h(\boldsymbol{\mu}_i)) : i = 1, 2, \cdots, M\}$ during the offline stage, where $\mathbf{u}_h(\boldsymbol{\mu}_i)$ is the full-order solution for each sample. The model is used during the online stage to recover the output $\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*)$ for any new input $\boldsymbol{\mu}^* \in \mathcal{P}$. Correspondingly, the reduced-order solution $\boldsymbol{u}_{\rm rb,reg}(\boldsymbol{\mu}^*) \in \mathcal{V}_{\rm rb}$ is given as

$$\boldsymbol{u}_{\rm rb,reg}(\boldsymbol{\mu}^*) = \sum_{l=1}^{L} (\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*))_l \boldsymbol{\psi}_l = \sum_{k=1}^{N_h} \left(\mathbf{V}\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*)\right)_k \boldsymbol{\phi}_k = \sum_{k=1}^{N_h} \left(\mathbf{u}_{\rm rb,reg}(\boldsymbol{\mu}^*)\right)_k \boldsymbol{\phi}_k \,, \tag{24}$$

where $\mathbf{V}\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*) = \mathbf{u}_{\rm rb,reg}(\boldsymbol{\mu}^*)$ recovers the nodal values of the solution. Once the regression model is obtained, the online stage only requires direct outputs from this model, ensuring that the online computation is carried out at low cost.

Algorithm 2 Regression-based RB method for nonlinear structural analysis (algebraic level)

1: **Offline stage:**
2: Compute N_s full-order snapshots $\{\mathbf{u}_h(\boldsymbol{\mu}^1), \mathbf{u}_h(\boldsymbol{\mu}^2), \cdots, \mathbf{u}_h(\boldsymbol{\mu}^{N_s})\}$ and form the snapshot matrix $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$;
3: Perform POD for $\mathbf{S}$ and get the L orthogonal bases $\mathbf{V} \in \mathbb{R}^{N_h \times L}$;
4: Prepare the training set $\mathcal{D} = \{(\boldsymbol{\mu}_i, \mathbf{V}^{\rm T}\mathbf{u}_h(\boldsymbol{\mu}_i)) : i = 1, 2, \cdots, M\}$;
5: Construct the regression model $\hat{\boldsymbol{\pi}}(\cdot)$ from $\mathcal{D}$.
6: **Online stage:**
7: Recover output $\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*)$ for a new parameter value $\boldsymbol{\mu}^*$;
8: Evaluate the reduced-order solution $\boldsymbol{u}_{\rm rb,reg}(\boldsymbol{\mu}^*) = \sum_{k=1}^{N_h} \left(\mathbf{V}\hat{\boldsymbol{\pi}}(\boldsymbol{\mu}^*)\right)_k \boldsymbol{\phi}_k$.

Remark 3: In some cases, the snapshots for the construction of the RB space can be included into the training set and reused as training samples.

We note the complete decoupling of online and offline stages, and the non-intrusive nature of the regression-based RB method. A Gaussian process model is utilized to construct the regression $\hat{\boldsymbol{\pi}}(\cdot)$, as discussed in the following section.

4. Gaussian process regression model

In supervised learning, regression is concerned with prediction of continuous quantities of interest by constructing a model from a set of observation data. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \cdots, M\}$ denote the training set of M observations, where each input $\mathbf{x}_i \in \mathcal{P} \subset \mathbb{R}^d$ consists of d entries and lies in the input domain $\mathcal{P}$, and y_i is the output corresponding to $\mathbf{x}_i$. In a Gaussian process regression (GPR) model [30, 36],

the observed input-output pairs are assumed to follow some unknown regression function $f : \mathcal{P} \to \mathbb{R}$ as $y_i = f(\mathbf{x}_i)$, possibly corrupted by noise. The model then infers a probabilistic distribution over functions given the data, and uses this distribution to make predictions when given new inputs.

4.1. Gaussian processes for regression

A Gaussian process (GP) is a collection of random variables, any finite number of which obeys a joint Gaussian distribution. In the case of GPR, let the prior on the regression function be a GP: for $(\mathbf{x}, \mathbf{x}') \in \mathcal{P} \times \mathcal{P}$,

$$f(\mathbf{x}) \sim \mathrm{GP}(0, \kappa(\mathbf{x}, \mathbf{x}'))\,, \tag{25}$$

whose mean is zero and covariance function is $\kappa : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, i.e.

$$\mathbb{E}[f(\mathbf{x})] = 0\,, \quad \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \kappa(\mathbf{x}, \mathbf{x}')\,. \tag{26}$$

There are many different choices for the covariance function. A frequently used one is the *squared exponential* (SE) *kernel*:

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|_{\mathbb{R}^d}^2\right)\,, \tag{27}$$

containing two hyperparameters: the standard deviation parameter σ_f and the correlated lengthscale ℓ. Another covariance function, that we will use in this work, is the *automatic relevance determination* (ARD) SE kernel:

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{m=1}^{d}\frac{(x_m - x'_m)^2}{\ell_m^2}\right)\,, \tag{28}$$

which considers an individual correlated lengthscale for each input dimension, and allows for differentiated relevances of input features to the regression.

Given a finite number of points in the input domain, a prior joint Gaussian is thus defined for the regression outputs:

$$\mathbf{f}|\mathbf{X} \;\sim\; \mathcal{N}(\mathbf{0}, \mathbf{K})\,. \tag{29}$$

where $\mathbf{f} = \{f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_M)\}^{\mathrm{T}}$, $\mathbf{X} = [\ \mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_M\ ]$, $\mathcal{N}(\mathbf{0}, \mathbf{K})$ represents a joint Gaussian distribution with mean values $\mathbf{0}$ and covariance matrix $\mathbf{K}$, and the covariance matrix $\mathbf{K}$ is calculated as

$$\mathbf{K} = \mathrm{cov}[\mathbf{f}|\mathbf{X}] = \mathbb{E}[\mathbf{f}\,\mathbf{f}^{\mathrm{T}}|\mathbf{X}] = \kappa(\mathbf{X}, \mathbf{X})\,. \tag{30}$$

It is realistic that the observations are corrupted by noise. In such cases, a noisy version of the regression function can be considered as

$$y = f(\mathbf{x}) + \epsilon\,, \tag{31}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$ is an independent Gaussian noise term. In this case, the interpolation of the training data is not required, but a 'close variation' from the observed outputs is controlled by the standard deviation σ_y for noise. The covariance of observed outputs and the prior joint Gaussian should be modified as

$$\mathbf{K}_y = \mathrm{cov}[\mathbf{y}|\mathbf{X}] = \mathbf{K} + \sigma_y^2\mathbf{I}_M\,, \quad \mathbf{y}|\mathbf{X} \;\sim\; \mathcal{N}(\mathbf{0}, \mathbf{K}_y)\,, \tag{32}$$

where $\mathbf{y} = \{y_1, y_2, \cdots, y_M\}^{\mathrm{T}}$ and $\mathbf{I}_M$ is the M-dimensional unit matrix.

Given a set of M^* new test inputs denoted by $\mathbf{X}^* \in \mathbb{R}^{d \times M^*}$, predictions of the corresponding noise-free outputs $\mathbf{f}^* \in \mathbb{R}^{M^*}$ are desired. To combine the information from training set with the predictions for test samples, the joint density of the observed outputs $\mathbf{y}$ and the noise-free test outputs $\mathbf{f}^*$ is expressed as

$$\begin{pmatrix}\mathbf{y}\\ \mathbf{f}^*\end{pmatrix} \;\sim\; \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}_y & \mathbf{K}^* \\ \mathbf{K}^{*T} & \mathbf{K}^{**}\end{bmatrix}\right)\,, \tag{33}$$

where $\mathbf{K}^{**} = \kappa(\mathbf{X}^*, \mathbf{X}^*)$ and $\mathbf{K}^* = \kappa(\mathbf{X}^*, \mathbf{X})$. Hence the posteriori predictive distribution can be obtained

by the standard rules for conditioning Gaussians, i.e.

$$\begin{aligned} &\mathbf{f}^*|\mathbf{X}^*,\mathbf{X},\mathbf{y} \ \sim \ \mathcal{N}(\mathbf{m}^*,\mathbf{C}^*)\,, \\ &\mathbf{m}^* = \mathbf{K}^{*\mathrm{T}}\mathbf{K}_y^{-1}\mathbf{y}\,, \quad \mathbf{C}^* = \mathbf{K}^{**} - \mathbf{K}^{*\mathrm{T}}\mathbf{K}_y^{-1}\mathbf{K}^*\,. \end{aligned} \tag{34}$$

The values of the hyperparameters $\boldsymbol{\theta}$ make significant difference on the predictive performance, with $\boldsymbol{\theta} = \{\sigma_f, \ell, \sigma_y\}$ for the case of SE kernel and $\boldsymbol{\theta} = \{\sigma_f, \ell_1, \cdots, \ell_d, \sigma_y\}$ for the case of ARD SE kernel. In this work, an empirical Bayesian approach of maximizing likelihood is adopted to determine a set of optimal values of the parameters. Using a standard gradient-based optimizer, one can estimate the optimal hyperparameters $\boldsymbol{\theta}_{\mathrm{opt}}$ via the maximization problem:

$$\boldsymbol{\theta}_{\mathrm{opt}} = \arg\max_{\boldsymbol{\theta}} \ \log p(\mathbf{y}|\mathbf{X}) = \arg\max_{\boldsymbol{\theta}} \left\{ -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}_y^{-1}(\boldsymbol{\theta})\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y(\boldsymbol{\theta})| - \frac{M}{2}\log(2\pi) \right\}, \tag{35}$$

where $p(\mathbf{y}|\mathbf{X})$ is the conditional density function of $\mathbf{y}$ given $\mathbf{X}$, also considered as the marginal likelihood defined by

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X})p(\mathbf{f}|\mathbf{X})\,\mathrm{d}\mathbf{f}\ . \tag{36}$$

The procedure of a GPR is given as the following algorithm.

Algorithm 3 GPR

Input: A training set of M observations $\mathcal{D} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \cdots, M\}$, a chosen kernel function $\kappa(\cdot,\cdot)$, test inputs $\mathbf{X}^* \in \mathbb{R}^{d\times M^*}$

Output: Test outputs $\mathbf{f}^*|\mathbf{X}^*,\mathbf{X},\mathbf{y}$

1: Estimate the optimal hyperparameters $\boldsymbol{\theta}_{\mathrm{opt}}$ by maximizing the likelihood, in each iterative step of which one needs to
2: Form a covariance matrix $\mathbf{K}_y = \kappa(\mathbf{X},\mathbf{X}) + \sigma_y^2\mathbf{I}_M$;
3: Calculate a vector $\mathbf{a} \in \mathbb{R}^M$ such that $\mathbf{K}_y\mathbf{a} = \mathbf{y}$;
4: Calculate the likelihood $\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{a} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{M}{2}\log(2\pi)$;
5: Calculate the gradient of the likelihood with respect to the hyperparameters;
6: Set $\mathbf{K}_y = \mathbf{K}_y(\boldsymbol{\theta}_{\mathrm{opt}})$ and $\mathbf{a} = \mathbf{a}(\boldsymbol{\theta}_{\mathrm{opt}})$ for the optimal hyperparameters;
7: Form correlation matrices $\mathbf{K}^{**} = \kappa(\mathbf{X}^*,\mathbf{X}^*)$ and $\mathbf{K}^* = \kappa(\mathbf{X}^*,\mathbf{X})$ for the optimal hyperparameters;
8: Calculate a matrix $\mathbf{A}^* \in \mathbb{R}^{M\times M^*}$ such that $\mathbf{K}_y\mathbf{A}^* = \mathbf{K}^*$;
9: Form the conditioning mean value vector $\mathbf{m}^* = \mathbf{K}^{*\mathrm{T}}\mathbf{a}$ and the corresponding covariance matrix $\mathbf{C}^* = \mathbf{K}^{**} - \mathbf{K}^{*\mathrm{T}}\mathbf{A}^*$;
10: Define $\mathbf{f}^*|\mathbf{X}^*,\mathbf{X},\mathbf{y} \ \sim \ \mathcal{N}(\mathbf{m}^*,\mathbf{C}^*)$.

4.2. Gaussian process regression for the reduced basis method of nonlinear structural analysis

As already mentioned in Section 3, the GPR is utilized in the RB method for nonlinear structural analysis. A GP regression model $\hat{\boldsymbol{\pi}}_{\mathrm{GP}} : \mathcal{P} \to \mathbb{R}^L$ is constructed for the mapping $\boldsymbol{\mu} \mapsto \mathbf{V}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu})$.

For the lth of L entries of $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\cdot)$, $1 \le l \le L$, the training data is set as $\mathbf{x}_i^l = \boldsymbol{\mu}_i$, $y_i^l = \mathbf{v}_l^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}_i)$ and $\mathcal{D}^l = \{(\mathbf{x}_i^l, y_i^l) : i = 1, 2, \cdots, M\}$ for a GPR model, with $\mathbf{v}_l$ being the jth column of $\mathbf{V}$. For a new parameter $\mathbf{x}^* = \boldsymbol{\mu}^* \in \mathcal{P}$ as test input, the corresponding output $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*)$ consists of L independent Gaussian distributions, i.e.

$$(\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*))_l|\mathbf{X}^l,\mathbf{y}^l \ \sim \ \mathcal{N}(\mathbf{m}^{l*},\mathbf{C}^{l*})\,, \quad l = 1, 2, \cdots, L\,, \tag{37}$$

where $\mathbf{X}^l = [\,\mathbf{x}_1^l \mid \mathbf{x}_2^l \mid \cdots \mid \mathbf{x}_M^l\,]$, $\mathbf{y}^j = \{y_1^l, y_2^l, \cdots, y_M^l\}^{\mathrm{T}}$, $\mathbf{m}^{l*}$ and $\mathbf{C}^{l*}$ for the lth entry of $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\cdot)$ are defined in the same way as $\mathbf{m}^*$ and $\mathbf{C}^*$ in (33). Correspondingly, the reduced-order discrete solution $\tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\boldsymbol{\mu}^*)$ collects N_h Gaussian distributions, i.e.

$$\tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\boldsymbol{\mu}^*) = \mathbf{V}\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*)\,. \tag{38}$$

The reduced order solution, expressed as a random field over Ω, is given as

$$\tilde{\boldsymbol{u}}_{\mathrm{rb,GPR}}(\boldsymbol{\mu}^*) = \sum_{l=1}^{L} (\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*))_l \boldsymbol{\psi}_l = \sum_{k=1}^{N_h} (\mathbf{V}\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*))_k \, \boldsymbol{\phi}_k = \sum_{k=1}^{N_h} (\tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\boldsymbol{\mu}^*))_k \, \boldsymbol{\phi}_k \,. \tag{39}$$

For a set of r test samples $\mathcal{T} = \{(\boldsymbol{\mu}_i^*, \mathbf{u}_h(\boldsymbol{\mu}_i^*)) : i = 1, 2, \cdots, r\}$, $\boldsymbol{\mu}_i^*$ being the ith test input and $\mathbf{u}_h(\boldsymbol{\mu}_i^*)$ being the corresponding full-order discrete solution, an average relative error for GPR predictions can be defined as

$$\bar{\epsilon}_t(\mathcal{T}) = \frac{1}{r} \sum_{i=1}^{r} \frac{\|\mathbf{u}_h(\boldsymbol{\mu}_i^*) - \mathbb{E}[\tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\boldsymbol{\mu}_i^*)]\|_{\mathbb{R}^{N_h}}}{\|\mathbf{u}_h(\boldsymbol{\mu}_i^*)\|_{\mathbb{R}^{N_h}}} = \frac{1}{r} \sum_{i=1}^{r} \frac{\|\mathbf{u}_h(\boldsymbol{\mu}_i^*) - \mathbf{V}\mathbb{E}[\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}_i^*)]\|_{\mathbb{R}^{N_h}}}{\|\mathbf{u}_h(\boldsymbol{\mu}_i^*)\|_{\mathbb{R}^{N_h}}} \,, \tag{40}$$

note that the mean values of GPR test outputs are considered as predictions.

Remark 4: In the context of structural optimization, reliability analysis, etc., gradient-based algorithms are used for solving optimization problems, see [11, 15, 25]. These algorithms usually require the derivatives of the structural responses with respect to the parameters. In (33), the mean value of GPR output $m^*(\mathbf{x}^*) = \mathbb{E}[f(\mathbf{x}^*)|\mathbf{X}, \mathbf{y}]$ for a new test parameter $\mathbf{x}^*$ is obtained as

$$m^*(\mathbf{x}^*) = \kappa(\mathbf{x}^*, \mathbf{X})^{\mathrm{T}} \mathbf{K}_y^{-1} \mathbf{y} \,. \tag{41}$$

Correspondingly, the derivative of m^* with respect to $\mathbf{x}^*$ is thus derived as

$$\frac{\partial m^*(\mathbf{x}^*)}{\partial \mathbf{x}^*} = \left[\frac{\partial \kappa(\mathbf{x}^*, \mathbf{X})}{\partial \mathbf{x}^*}\right]^{\mathrm{T}} \mathbf{K}_y^{-1} \mathbf{y} \,, \tag{42}$$

which only depends on the parameter location $\mathbf{x}^*$ and is not correlated to any other test points. For a nonlinear structural problem, the derivative of test output $\mathbb{E}[\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*)]$ can be calculated for test input $\boldsymbol{\mu}^*$, entry by entry, and the response sensitivity derivative expressed as

$$\frac{\partial}{\partial \boldsymbol{\mu}^*} \tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\boldsymbol{\mu}^*) = \mathbf{V} \frac{\partial}{\partial \boldsymbol{\mu}^*} \mathbb{E}[\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}^*)] \,. \tag{43}$$

4.3. Active data selection for training samples

In the training set $\mathcal{D}$, each input-output pair $(\boldsymbol{\mu}_i, \mathbf{V}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}_i))$ requires the calculation of full-order solution $\mathbf{u}_h(\boldsymbol{\mu}_i)$ at parameter $\boldsymbol{\mu}_i$, $i = 1, 2, \cdots, M$. A large number of training samples M implies substantial computation to prepare the training data. For efficiency, one should choose a set of 'optimal' training samples from a pool of parameters, so the full-order solutions are only calculated at a smaller number of optimized parameter values without substantial loss of accuracy. Referred to as *active data selection*, this type of selecting technique has been studied and developed in the field of *active learning* [13, 20, 23, 33, 34].

In the context of GPR model for nonlinear structural analysis, an active data selection algorithm is given as follows, analogous to a scheme of active learning [3, 34].

Algorithm 4 Selection algorithm for active training data

Input: A parameter pool $\mathcal{P}_s \subset \mathcal{P}$ with a large number of elements – uniform lattice or generated in $\mathcal{P}$, $\boldsymbol{\mu}_1 \in \mathcal{P}_s$, tolerance `tol`, $M = 1$, training set $\mathcal{D} = \emptyset$ and active parameter set $\mathcal{P}_{tr} = \{\boldsymbol{\mu}_1\}$, test samples $\mathcal{T} = \{(\boldsymbol{\mu}_i^*, \mathbf{u}_h(\boldsymbol{\mu}_i^*)) : i = 1, 2, \cdots, r\}$ with full-order solutions

Output: GPR model $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\cdot)$

Calculate full-order solution $\mathbf{u}_h(\boldsymbol{\mu}_M)$;
Set $\mathcal{D} = \mathcal{D} \cup \{(\boldsymbol{\mu}_M, \mathbf{V}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}_M))\}$;
Train a GPR model $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\cdot)$ based on $\mathcal{D}$;
Calculate average relative error for test samples $\bar{\epsilon}_t(\mathcal{T})$;
if $\bar{\epsilon}_t(\mathcal{T}) \leq$ `tol` **then**
  Terminate;
else
  Update the parameter pool $\mathcal{P}_s = \mathcal{P}_s \setminus \mathcal{P}_{tr}$;
  for each $\boldsymbol{\mu} \in \mathcal{P}_s$ **do**
    Compute the output $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu})$;
    Evaluate the error indicator $\eta(\boldsymbol{\mu})$;
  end for
  Choose $\boldsymbol{\mu}_{M+1} = \arg\max_{\boldsymbol{\mu} \in \mathcal{P}_s} \eta(\boldsymbol{\mu})$;
  Set $M = M + 1$, $\mathcal{P}_{tr} = \mathcal{P}_{tr} \cup \{\boldsymbol{\mu}_{M+1}\}$ and **go to** 1.
end if

In the active data selection algorithm, a natural and simple consideration is to use *standard deviations* to define the error indicator $\eta(\cdot)$ for evaluating the regression model $\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\cdot)$. Here, one choice of $\eta(\cdot)$ is

$$\eta(\boldsymbol{\mu}) = \sqrt{\sum_{k=1}^{N_h} \sum_{l=1}^{L} V_{kl}^2 \ \mathrm{sd}[(\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\boldsymbol{\mu}))_l]^2}\,,$$

analogous to the error in $\mathbb{R}^{N_h}$-norm, and $\mathrm{sd}[\cdot]$ denotes the standard deviation of a random variable. Each new active training sample is selected from the pool as the maximizer of the standard-deviation-based error indicator η, and the selection procedure is terminated once a satisfactory prediction quality is achieved. Alternatively, a desired training sample number M can be defined in advance.

Since structural responses with respect to typical structural parameters are usually continuous, even smooth in most cases, the GPR is accurate even though it is not as powerful as some more advanced methods in regression, such as *artificial neural networks* (ANNs) [16]. The GPR can provide a natural and simple standard-deviation-based error indicator in active data selection, and the conciseness of the GPR model guarantees an efficient procedure of data selection and regression during the offline stage. In the context of uncertainty quantification, one can take both the uncertainty in the GPR model and the uncertainty in parameters into account using the Bayesian theory.

5. Numerical examples

In this section, numerical results for two examples, one in 1D and one in 3D, will be presented to validate the effectiveness and accuracy of the proposed approach.

The `FLagSHyP` MATLAB program is used as high-fidelity solver for the numerical examples. The MATLAB version of `FLagSHyP` [5] is a program for the finite element analysis of static nonlinear problems in solid mechanics. Its numerical scheme is introduced in [6, 7]. The two example problems, large deformation analysis of a trussed frame and that of a twisting column, can also be found in [6] as computational implementations of the nonlinear finite element method.

5.1. *One-dimensional example: a trussed frame*

The first example is a frame made of a beam and a column. As shown in Figure 2, the frame is trussed by 596 one-dimensional elements and loaded by a concentrated load on the beam. The number of DOFs of the full-order model is $N_h = 476$. Two types of constitutive relations are considered in this problem: one-dimensional strech-based hyperelasticity and hyperelasto-plasticity. We refer to [6] for more details of these constitutive laws. The quantities in Figure 2 are given as: Young's modulus $E = 210$ GPa, unit force $F_0 = 1$ N and unit displacement $\Delta_0 = 1$ mm. With a uniform Young's modulus E in the whole structure, equilibrium paths, referred to as load-displacement curves, for the two constitutive relations are shown in Figure 3(a). For the case of hyperelasticity, we note that the *snap-back* phenomena occurs, and the configurations at different loading stages are plotted in Figure 3(b).

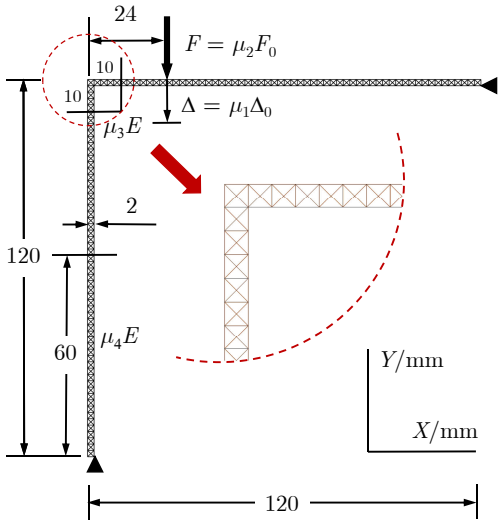
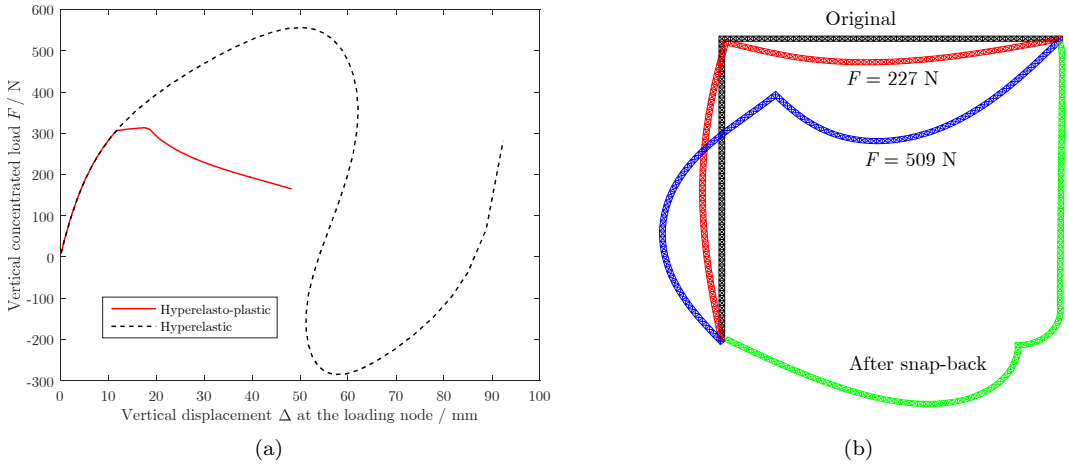


Figure 2: Geometry and input parameters for a trussed frame



(a) (b)

Figure 3: (a) Load-displacement curves in both hyperelastic and hyperelasto-plastic cases; (b) Configurations at different loading stages in the hyperelastic case.

This problem is parametrized by four parameters: μ_1 is the quantitative value of the concentrated load measured in N, μ_2 is the quantitative value of the downward displacement at the loading node, measured

in mm, μ_3 is the scaling factor of Young's modulus of a 10 mm $\times$ 10 mm inverted L-shaped zone at the beam-column joint, and μ_4 is the scaling factor of Young's modulus of the lower half of column. The loading procedure can be either force-controlled or displacement-controlled, i.e., μ_1 and μ_2 are the two different controlling parameters of loading, respectively. For the complementary parts of the zones with parametrized Young's modulus, the modulus is fixed as E. Three parametrized cases will be analyzed as follows.

Case 1: μ_1, hyperelasto-plasticity

The value of controlling downward deflection Δ for the loading procedure is considered as the only parameter in this case. The concentrated load is applied to the frame by 40 loading increments with the arc-length method used, and the size of each increment is determined automatically by the method.

After this full-order incremental procedure, 40 high-fidelity solutions with different values of μ_1 are naturally collected, but those at other values of μ_1 are not available. It is impractical to get the high-fidelity solution at any parameter value. Thus the 40 full-order solutions are used as snapshots for constructing the RB space and as training data for the GPR model, i.e. $\Theta = \mathcal{P}_{tr}$. As the 40 training samples are prepared in advance, active data selection does not make much difference on the computational efforts for training the GPR model, so the data selection algorithm will not be adopted in this case.

The POD for the 40 snapshots gives $L = 5$ bases, and a GPR model with an SE kernel is constructed for the 5 projection coefficients onto the bases, as shown in Figure 4. In this figure, the prediction curves show mean values of the GPR outputs, lying in the interval of 95% confidence level. In Figure 4(f), the vertical displacement Δ at the loading node is extracted from the reduced-order prediction $\tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\mu_1) = \mathbf{V}\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\mu_1)$ for each parameter μ_1 in the training samples, matching well with the 'perfect' identical fitting.

The average relative error of the projection for $N_s = 40$ snapshots is 0.0033, calculated as

$$\bar{\epsilon}_{\mathbf{V}}(\Theta) = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{\|\mathbf{u}_h(\boldsymbol{\mu}^i) - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}_h(\boldsymbol{\mu}^i)\|_{\mathbb{R}^{N_s}}}{\|\mathbf{u}_h(\boldsymbol{\mu}^i)\|_{\mathbb{R}^{N_s}}}\,, \tag{44}$$

while the GPR predictions for the training samples $\Theta = \mathcal{P}_{tr}$ have an average relative error $\bar{\epsilon}_t(\Theta) = 0.0154$ compared with the corresponding full-order solutions.

Case 2: μ_2, hyperelasto-plasticity

The load F is considered as parameter μ_2 in this case. As shown in Figure 3(a), the equilibrium path for hyperelasto-plastic constitutive relation is not monotone, meaning that the displacement field is multi-valued with respect to the external load $F = \mu_2$. Thus the regressions for increasing and decreasing stages in the equilibrium path are carried out separately in this case. Based on 170 training samples, among which 97 are in the increasing stage and 73 are in the decreasing stage, a regression model is obtained by an ANN, based on *multi-layer perceptrons* (MLPs) [4]. As is well known, the ANN is a powerful tool for nonlinear regression, so the regression results by the ANN based on the refined set of training data are considered as a reference. Using the proposed algorithm of active data selection, 85 samples are picked from the 170-sample set to derive a GPR model. Predictive results by the GPR model are shown in Figure 5. In this figure, predictions of the projection coefficients, obtained by both the ANN and the GPR, are plotted versus their 'exact' values directly calculated from the full-order solutions. After extracting the vertical displacement values from the reduced-order solutions by both regression approaches, the corresponding equilibrium paths are compared in Figure 5(f). It can be seen that the results by the GPR match well with those obtained by the ANN, even though they are not exactly coincident at some parameter locations in the predictions for the 5th coefficient. Confirmed by the fact that GPR achieves the similar accuracy with ANN by using half of the training samples, it supports that GPR is a good choice for the regression method in this context.

Case 3: $(\mu_3, \mu_4) \in [0.5, 1.5] \times [0.8, 1.2]$, hyperelasticity

The parameters μ_3 and μ_4 reflect local material properties and vary in a closed set $\mathcal{P} = [0.5, 1.5]\times[0.8, 1.2]$. Under the hyperelastic constitutive relation, the configuration under a fixed load $F = 500$ N in the first increasing stage of equilibrium path (see Figure 3(a)) is taken into account. A Newton-Raphson algorithm is employed in loading increments until $F = 500$ N is reached. From a set snapshots at $N_s = 25$ randomly generated points in $\mathcal{P}$, an RB space of rank $L = 5$ is constructed. Then the active data selection algorithm

is adopted to select M training sample from a pool of 400 randomly generated parameter locations. For $M = 50$, the GPR results for the 1st, 3rd and 5th projection coefficients, i.e. the corresponding entries of $\mathbf{V}^{\mathrm{T}}\mathbf{u}_h$, are plotted in Figure 6. During the selection procedure, as M increases, the first 90 parameter positions are shown in Figure 7(a). In this case, $r = 30$ test samples $\mathcal{T}$ with an average relative projection error $\bar{\epsilon}_{\mathbf{V}}(\mathcal{T}) = 1.41 \times 10^{-4}$ are randomly generated to evaluate the prediction quality of the GPR model. As can be seen in Figure 7(b), the average relative error of GPR test predictions is decaying rapidly as M increases from 10 to 50. When the number of selected training samples increases to 40, the order of magnitude of the average relative error decreases to 10^{-4}, showing the accuracy of GPR model and the efficiency of active data selection.

5.2. Three-dimensional example: a twisting column

The second example considers a three-dimensional column under the torsion of a pair of uniformly distributed pressure loads $p = \mu_1 p_0$ that are opposite to each other, as illustrated in Figure 8. The stress-strain behavior follows a relation of compressible neo-Hookean elasticity [6], with a fixed bulk modulus $K = 5E_0/3$ and a parametrized shear modulus $G = \mu_2 E_0$. The units of load and modulus are given as $p_0 = 1$ and $E_0 = 100$. Thus the problem is parametrized by μ_1 and μ_2. The full-order solution of this system is obtained via finite element analysis, in which 576 hexahedral elements are employed, Newton-Raphson algorithm is used for the iteration and the number of DOFs is $N_h = 2700$. Taking $\mu_2 = 1.0$, twisting configurations of the column at different loading stages are shown in Figure 8.

Case 1: $\mu_1 \in \ ]0, 130[$

The magnitude of pressure load is considered as the only parameter μ_1 in this case. High-fidelity solutions at 50 loading increments are prepared in advance, with μ_1 approaching 130. $N_s = 25$ of them with even sequence numbers are taken as snapshots, from which an RB space of $L = 10$ dimensions is constructed. If projecting all the 50 full-order solutions onto the RB space and using them as training data, a GPR model can be trained with an average relative error 1.70×10^{-3} of GPR predictive approximations for the 50 samples. When active data selection is adopted with a predictive error tolerance $\texttt{tol} = 4 \times 10^{-3}$, as introduced in Algorithm 4, 28 training samples are selected. The GPR model obtained is shown in Figure 10(a). In this figure, displacement u_X in the X-direction at node B, labelled in Figure 8, is extracted from the reduced-order solution $\tilde{\mathbf{u}}_{\mathrm{rb,GPR}}(\cdot) = \mathbf{V}\hat{\boldsymbol{\pi}}_{\mathrm{GP}}(\cdot)$, and plotted versus μ_1. The derivative of this displacement with respect to μ_1, i.e. $\mathrm{d}u_X/\mathrm{d}\mu_1$, is then calculated from the same GPR model, as discussed in Remark 4, and shown in Figure 10(b).

Case 2: $(\mu_1, \mu_2) \in \ ]0, 40] \ \times [0.8, 1.2]$

In this case, the parameter pair (μ_1, μ_2) varies in $]0, 40] \times [0.8, 1.2]$. Snapshots are calculated at $N_s = 25$ randomly picked parameter points to construct an RB space of rank $L = 6$. From a pool of 10×11 parameter locations, 40 are selected as training samples, based on which a GPR model with the ARD SE kernel is derived with an average relative error of 2.1×10^{-3} for test samples. As in Figure 11, the 40 selected parameter locations are labelled, and the predictive results for u_X at node B are plotted.

Furthermore, three computational times are compared in Figure 12; the time for calculating one full-order solution at $(\mu_1, \mu_2) = (20, 1.0)$, average time for 40 loops of GPRs in the active data selection, and average time for the calculation of 861 predictive outputs for test samples. One can see that the regressions are recovered very efficiently, and direct outputs from the GPR model in the online stage are obtained at low cost, providing an efficient tool for solving parametrized nonlinear problems.

6. Conclusions

A non-intrusive RB method is proposed for the ROM of parametrized nonlinear structural problems. In the framework of this method, an RB space is constructed offline by POD as the low-rank approximation to the space spanned by a collection of full-order snapshots. Rather than the conventionally used Galerkin projection scheme, a regression-based approach is adopted to determine the reduced-order solution for any desired new parameter value. Based on the offline establishment of a GPR model between parameter values and projection coefficients, only direct outputs from the model are required during the online stage

to obtain the reduced-order solutions at new parameter locations. Hence, the regression-based approach ensures a full decoupling between offline and online stages, and is non-intrusive. With both the accuracy and the efficiency validated by numerical examples, the proposed RB method is shown to be a powerful tool for solving parametrized nonlinear structural problems.

In multi-query and real-time contexts of structural analysis, the proposed scheme is able to reduce the model order effectively with a controlled loss of accuracy, and can achieve fast and reliable online calculations for desired parameter values, saving the high computational cost of full-order solutions. This provides a promising technique for the CAE softwares of large-scale structural systems.

References

[1] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An empirical interpolationmethod: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[2] K.-J. Bathe. *Finite Element Procedures*. Klaus-Jurgen Bathe, 2006.

[3] J. Beck and S. Guillas. Sequential design with mutual information for computer experiments (mice): Emulation of a tsunami model. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):739–766, 2016.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] J. Bonet, A. J. Gil, and R. D. Wood. *FLagSHyP software*. http://www.flagshyp.com, attached to *Nonlinear Solid Mechanics for Finite Element Analysis: Statics,* Cambridge University Press, 2016.

[6] J. Bonet, A. J. Gil, and R. D. Wood. *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. Cambridge University Press, 2016.

[7] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge university press, 1997.

[8] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

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

[10] W. Chen, J. S. Hesthaven, B. Junqiang, Z. Yang, and Y. Tihao. A greedy non-intrusive reduced order model for fluid dynamics. Technical report, submitted to American Institute of Aeronautics and Astronautics, 2017.

[11] K. K. Choi and N.-H. Kim. *Structural Sensitivity Analysis and Optimization 1: Linear Systems*. Springer Science & Business Media, 2006.

[12] P. G. Ciarlet. *Mathematical Elasticity, Vol. I: Three-dimentional Elasticity*. North-Holland Publishing Co., Amsterdam, 1988.

[13] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

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

[15] M. Guo and H. Zhong. Strict upper and lower bounds for quantities of interest in static response sensitivity analysis. *Applied Mathematical Modelling*, 49:17–34, 2017.

[16] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[17] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.

[18] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. Technical report, submitted to Journal of Computational Physics, 2017.

[19] C. Jäggli, L. Iapichino, and G. Rozza. An improvement on geometrical parameterizations by transfinite maps. *Comptes Rendus Mathematique*, 352(3):263–268, 2014.

[20] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[21] J. Ko and Y. Ni. Technology developments in structural health monitoring of large-scale bridges. *Engineering structures*, 27(12):1715–1725, 2005.

[22] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. Proper orthogonal decomposition and its applicationspart i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.

[23] D. J. MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[24] A. Manzoni and F. Negri. Automatic reduction of pdes defined on domains with variable shape. MATHICSE Technical Report, EPFL, 2016.

[25] R. E. Melchers and A. T. Beck. *Structural Reliability Analysis and Prediction*. John Wiley & Sons, 2017.

[26] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[27] 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.

[28] A. T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. Copyright MIT 2007, MIT Pappalardo Graduate Monographs in Mechanical Engineering, http://www.augustine.mit.edu, 2007.

[29] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

[30] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT press Cambridge, 2006.
[31] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1, 2007.
[32] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. i. teil: Entwicklung willkrlicher funktionen nach systemen vorgeschriebener. *Mathematische Annalen*, 63:433–476, 1907.
[33] S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: Active data selection and test point rejection. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 241–246. IEEE, 2000.
[34] B. Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, 2010.
[35] J. C. Simo and T. J. Hughes. *Computational Inelasticity*. Springer Science & Business Media, 2006.
[36] C. K. Williams and C. E. Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.
[37] O. C. Zienkiewicz, R. L. Taylor, and D. D. Fox. *The Finite Element Method for Solid and Structural Mechanics, 7th edition*. Elsevier, 2014.
[38] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, 7th edition*. Elsevier, 2013.

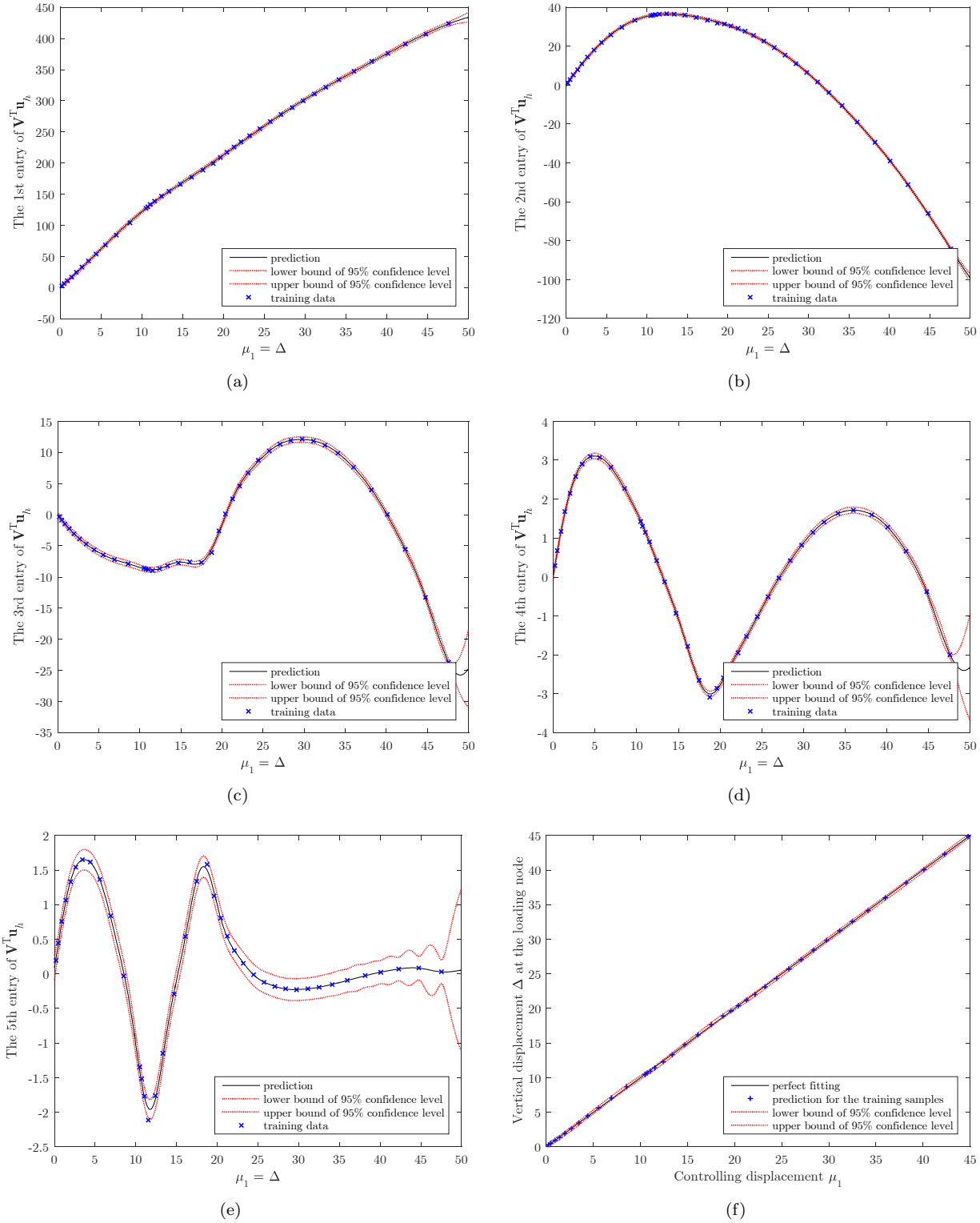


Figure 4: Predictive results by the GPR: (a) – (e) Regression results by the GPR for the 5 entries of $\mathbf{V}^{\mathrm{T}}\mathbf{u}_h$; (f) Extracting vertical displacement Δ at the loading node from the GPR results.

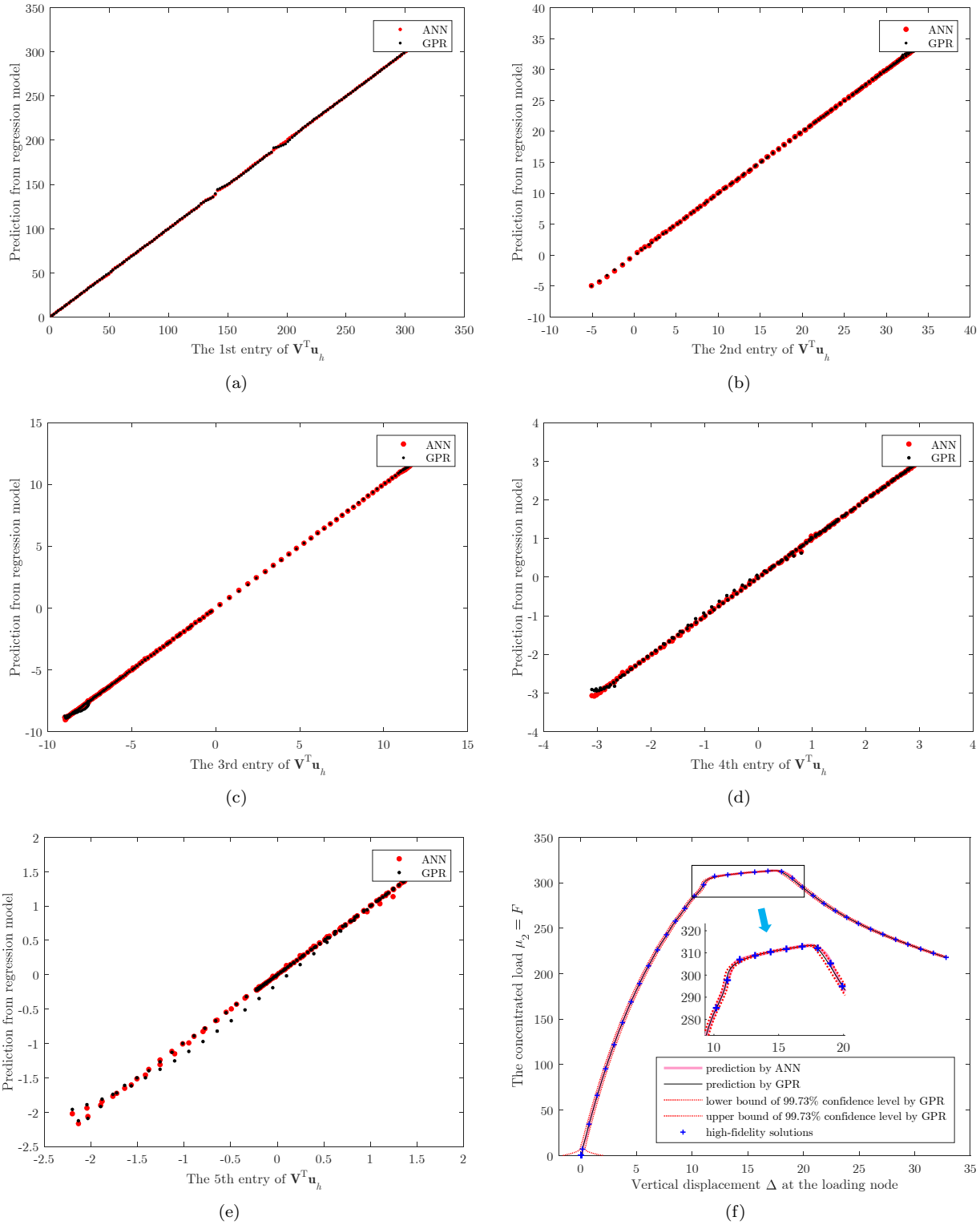


Figure 5: Predictive results by the GPR and the ANN: (a) – (e) Comparisons of predictive results by the GPR and the ANN for the 5 entries of $\mathbf{V}^{\mathrm{T}}\mathbf{u}_h$; (f) Regression results for the load-displacement curve by both the GPR and the ANN.

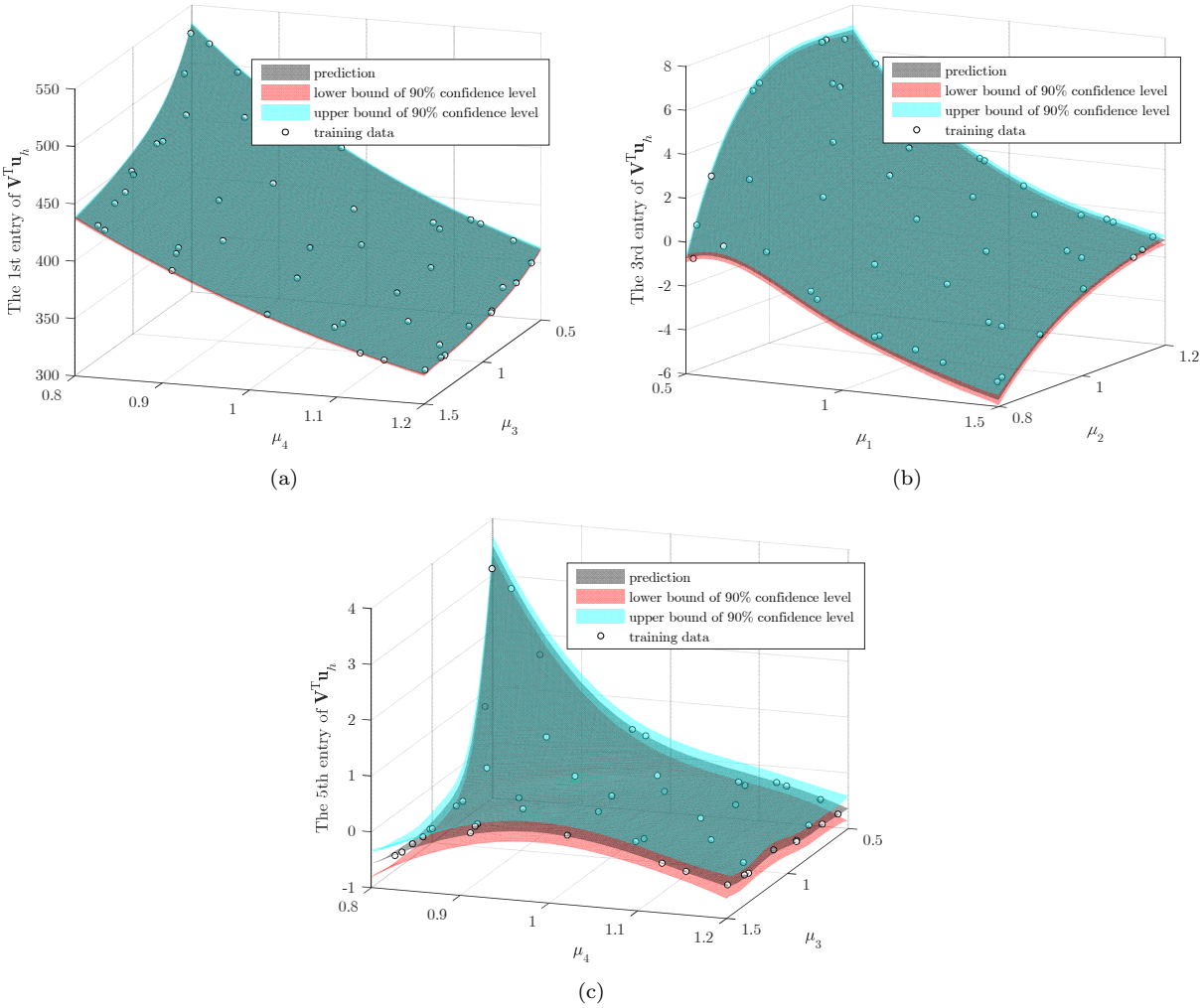
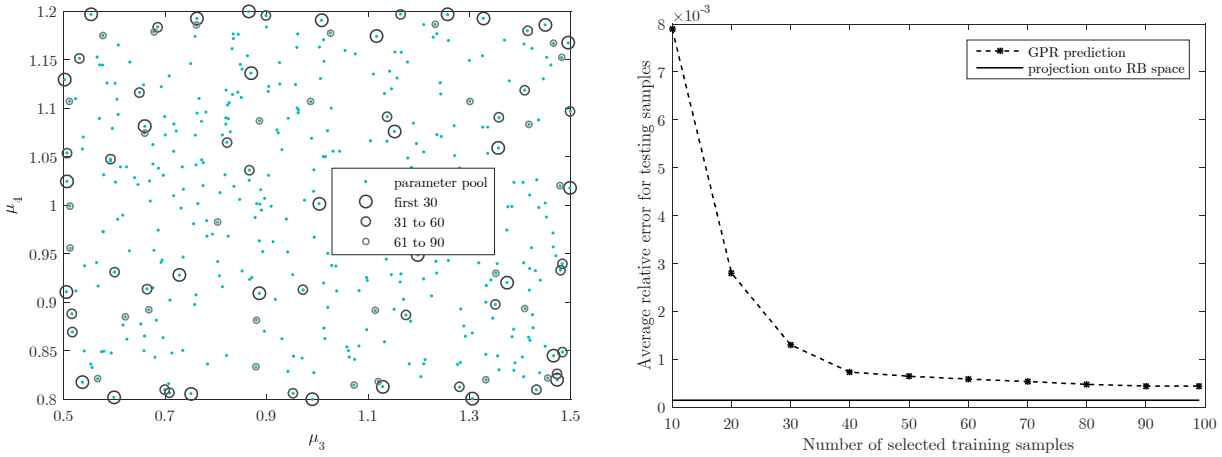


Figure 6: Regression results for the 1st, 3rd and 5th entries of $\mathbf{V}^{\mathrm{T}}\mathbf{u}_h$ from a GPR model trained by 50 samples selected from a pool of 400 parameter values.



(a) (b)

Figure 7: (a) Parameters corresponding to the first 90 samples selected from the pool; (b) Average error decay for 30 testing samples.

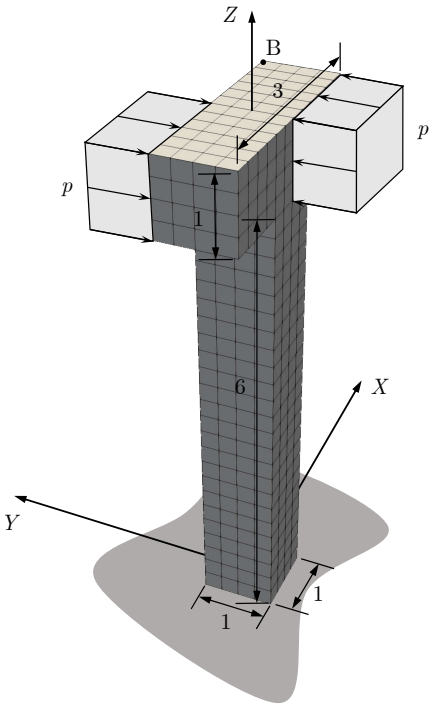


Figure 8: Geometry and pressure loads for the twisting column

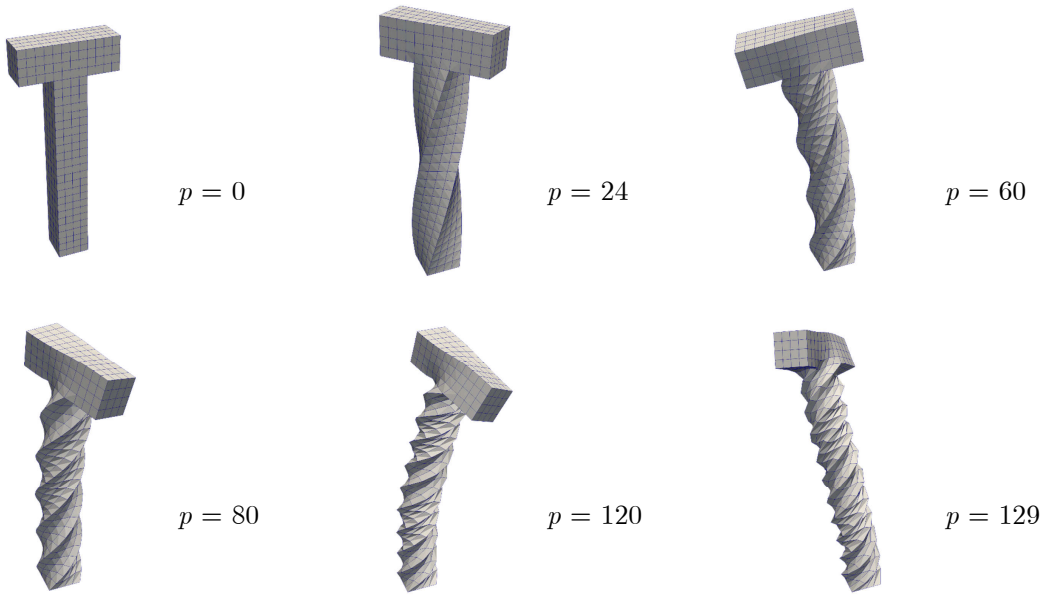
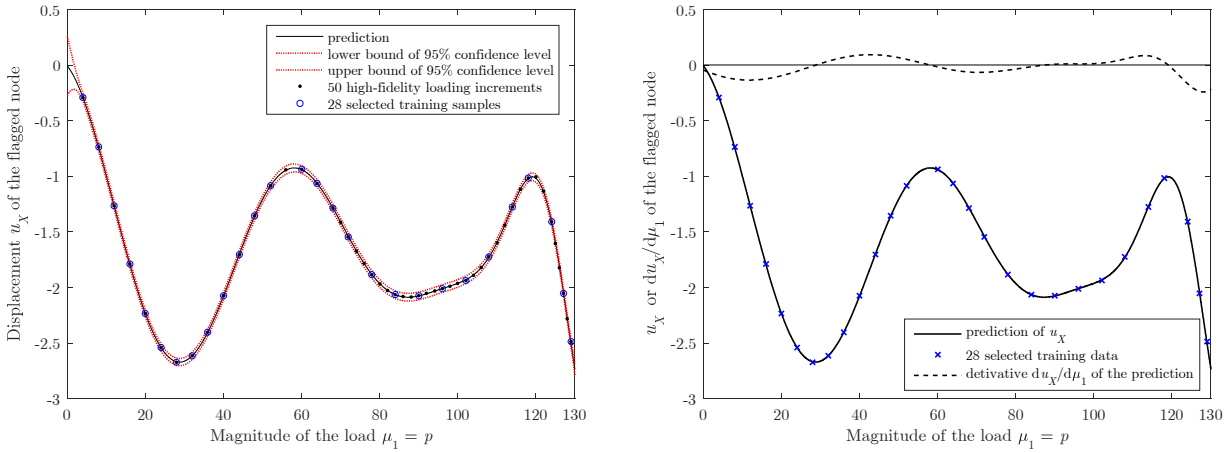


Figure 9: Configurations of the twisting column at different loading stages



(a) (b)

Figure 10: (a) Regression results for the curve of displacement u_X of the labelled node B versus pressure load $\mu_1 = p$; (b) Prediction of u_X at node B and its derivative with respect to μ_1, both calculated from the GPR model trained by 28 selected samples.

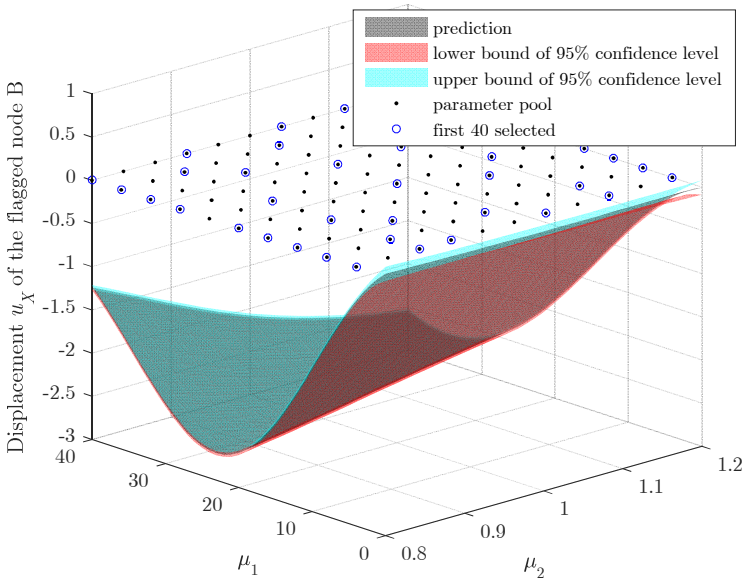
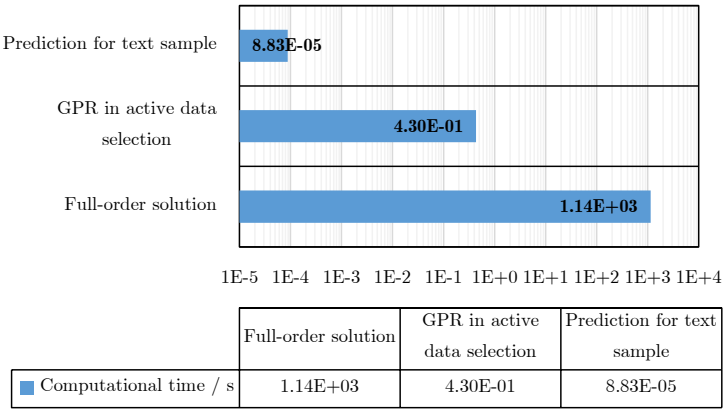


Figure 11: Regression results for the surface of displacement u_X of the labelled node B versus μ_1 and μ_2



	Full-order solution	GPR in active data selection	Prediction for text sample
Computational time / s	1.14E+03	4.30E-01	8.83E-05

Figure 12: Comparison of times: computational time of a full-order solution at $(\mu_1, \mu_2) = (20, 1.0)$, average time for 40 loops of GPRs in active date selection, and average time for 861 GPR predictive outputs for test samples