Data-driven reduced order modeling for time-dependent problems

Mengwu Guo*, Jan S. Hesthaven

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

Abstract

A data-driven reduced basis (RB) method is proposed for parametrized time-dependent problems. This method requires the offline preparation of a database comprising the full-order solutions of time history at parameter locations. Based on the full-order data, a reduced basis is constructed by the proper orthogonal decomposition (POD), and the maps between the time-parameter values and the projection coefficients onto the RB are approximated as a regression model. With a natural tensor grid between the inputs of time and parameters in the database, the singular-value decomposition (SVD) is used to extract the principal components in the data of projection coefficients, and the regression functions are represented as the linear combinations of several tensor products of two Gaussian processes, one of time and the other of parameters. During the online stage, the solutions at new time-parameter locations in the considered domain can be recovered rapidly via direct outputs from the regression models. Featuring a non-intrusive nature and the complete decoupling of the offline and online stages, the proposed approach provides a reliable and efficient tool for solving parametrized time-dependent problems, and its effectiveness is illustrated by non-trivial numerical examples.

Keywords: Data-driven, non-intrusive reduced order modeling, time-dependent problem, proper orthogonal decomposition, Gaussian process regression, tensor decomposition, machine learning

1. Introduction

In science and engineering, many *time-dependent* problems are described as *parametrized partial differential equations* [16, 28], in which the parameters characterize material properties, source terms, underlying geometry, boundary or initial conditions, and so on. In the context of *multi-entry* or *real-time* analysis, it is often required to solve the model for a large number of different parameter values. The great demands on both CPU time and memory make the high-fidelity simulations too expensive to allow repeated solutions of the time history for varying parameters. On the other hand, the repeated high-fidelity evaluations are often not really necessary as intrinsic similarities among the parameter-dependent solutions can be exploited to recover the solutions for new parameter values.

During the past decades, *reduced order modeling* (ROM) has been developed to address this issue. The key idea of the ROM is to replaced the original full-order model with a reduced-order model with significantly reduced dimensionality and controlled loss of accuracy, to ensure reliable evaluations of the reduced model at substantially reduced computational cost.

Featuring an *offline-online* framework, the *reduced basis* (RB) *method* [16, 26, 28, 30] is a powerful technique for the ROM of parametrized problems. With a significantly smaller dimension than the full-order model, a reduced space is spanned by a set of RB modes that are extracted offline from a collection of full-order snapshots at several time-parameter locations. Two main approaches for extracting the RB are

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

the *Greedy algorithm* [26, 30, 34] and the *proper orthogonal decomposition* (POD) [16, 19, 28]. The former selects a set of snapshots as the basis, following some error estimator/indicator and optimal criterion, while the latter utilizes the singular-value decomposition (SVD) to extract the RB. For time-dependent problems, required to capture the causality with the evolution over time, a combined version is proposed as the POD-greedy approach [14, 16]. Here the POD is used for the compressions over time, and the greedy procedure is employed in the parameter domain to enable an efficient treatment of parameter variations. Once the RB space is constructed, the approximated solution for a new desired parameter value is recovered online in the reduced space. Conventionally, a Galerkin projection is adopted to determine the (evolving) combination coefficients associated with the RB, yielding the (time-dependent) reduced-order solutions during the online stage.

For an RB scheme, the key to a successful reduction in computational cost is a full decoupling between the offline and online stages, i.e. the reduced space is parameter-independent and the online computations are independent of the size of the high-fidelity solution. However, such a decoupling is often difficult or even impossible for a general nonlinear problem with non-affine dependence on parameters. In such a case, assembly of the reduced model is embodied online, within the time steps, loading increments, nonlinear iterations and updates of the configuration. The empirical interpolation method (EIM) [4] and its discrete variants [10, 23], also referred to as the hyper-reductions [23], are developed to deal with this problem by recovering an affine expansion of the differential operator in a non-affine case. Such strategies are problem-dependent, or intrusive, and suffer from inflexibility in complex applications. For many time-dependent problems, some intrinsic structures inside the truth model are lost during order reduction, which can result in a qualitatively wrong or unstable reduced model. Stability of the reduced model for a time-dependent problem remains an important open question in many cases [6].

There are plenty of works on the ROM. For time-dependent problems, there are many successful cases of ROM, e.g. the RB method for linear evolution equations [14], the POD-based reduced-order fluid/structure modeling [20], the nonlinear ROM based on local reduced-order bases [3], some developments of the RB method for complex models [18, 27, 31], and some recently proposed structure-perserving RB approaches for Hamiltonian problems [1, 2]. One can refer to [6] for a comprehensive survey. To the best of our knowledge, however, no *non-intrusive* approaches have been proposed for the order reduction of parametrized time-dependent models.

In this work, we consider the parametrized time-dependent problems in the following form:

$$\mathcal{L}[\dot{u}(x,t;\boldsymbol{\mu});\boldsymbol{\mu}] + \mathcal{N}[u(x,t;\boldsymbol{\mu});\boldsymbol{\mu}] = f(x,t;\boldsymbol{\mu})\,, \quad (x,t,\boldsymbol{\mu}) \in \Omega \times \mathcal{T} \times \mathcal{P}\,, \tag{1}$$

with some properly defined initial and boundary conditions. Here $\Omega \subset \mathbb{R}^m$, $\mathcal{T} = [0,T]$ and $\mathcal{P} \subset \mathbb{R}^d$ represents the domains of space, time and parameters, respectively, with $m = 1, 2$ or 3 and d being the number of the parameters characterizing the model. $u : \Omega \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n$ denotes the parameterized time-dependent solution field, n is the dimension of the system, and $f : \Omega \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n$ is the source term. Moreover, $\mathcal{L}[\cdot;\boldsymbol{\mu}]$ is a linear operator and $\mathcal{N}[\cdot;\boldsymbol{\mu}]$ is a nonlinear differential operator, both associated with the space coordinates x and characterized by the parameters $\boldsymbol{\mu}$.

To enable good performance, flexibility, robustness and online efficiency, a regression-based approach is used in this work. A Gaussian-type regression was combined with the RB method in [24, 25] to predict scalar quantities of interest of the high-fidelity simulation. A regression-based approach has been developed to evaluate the reduced-order solution field for nonlinear time-independent problems, equipped with the artificial neural networks [15] in [17] and a Gaussian process regression (GPR) [29, 35] in [13]. The regression-based approach enjoys a complete decoupling of the offline and online stages. During the offline stage, the RB modes are extracted from a set of full-order snapshots by the POD, and the maps between the parameters and the projection coefficients onto the RB are approximated as regression models, which are trained from high-fidelity data by *supervised learning* [7, 21]. The evaluations for new parameter values only require direct outputs from the regression models and linear combinations of the RB, implying a very fast online computation. One can notice that the high-fidelity solver generates the full-order samples as a 'black-box', and these samples are used to construct the RB and regression models in a data-driven way. The whole offline-online procedure is carried out at the algebraic level, which guarantees a non-intrusive nature of the

regression-based approach. This allows dealing with different problems in the same way.

In this paper, the GPR-based scheme is utilized and further developed for time-dependent problems. The time coordinate t is considered as another parameter in the system. Over the time and parameter domains in consideration, a set of full-order solutions are prepared offline, which includes the snapshots for the POD and the samples for training the regression models. A tensor-decomposition-based approach is proposed for a more reliable and efficient regression. We use the SVD to extract the principal components of the data of each projection coefficient and represent the data as the combination of tensor products of discrete time- and parameter-models. The regression function for the projection coefficient is expressed in a similar formulation, with the continuous time- and parameter-modes approximated as GPR models. After obtaining the reduced model, an error surrogate can be recovered from the available data as another GPR model, which can provide indicators to certify the quality of the ROM.

Following this introduction, the data-driven RB method is presented in Section 2. After a brief review of the ideas of GPR, the tensor-decomposition-based regression and the error surrogate model are introduced in Section 3. In Section 4, the proposed method is validated by three numerical examples, the 1-D viscous Burgers' equation, incompressible fluid flow around a cylinder, and the large deformation analysis of a trussed frame. Conclusions are drawn in Section 5.

2. A data-driven reduced basis method

After discretizing (1) in space Ω by some discretization scheme, one has the following system of parameterized ODEs:

$$\mathbf{L}_h[\dot{\mathbf{u}}_h(t;\boldsymbol{\mu});\boldsymbol{\mu}] + \mathbf{N}_h[\mathbf{u}_h(t;\boldsymbol{\mu});(\boldsymbol{\mu})] = \mathbf{f}_h(t;\boldsymbol{\mu})\,, \quad (t,\boldsymbol{\mu}) \in \mathcal{T}\times\mathcal{P}\,, \tag{2}$$

where $\mathbf{u}_h : \mathcal{T}\times\mathcal{P} \to \mathbb{R}^{N_h}$ is the discrete solution, N_h is the number of degrees of freedom (DOFs), $\mathbf{L}_h$, $\mathbf{N}_h$ and $\mathbf{f}_h : \mathcal{T}\times\mathcal{P} \to \mathbb{R}^{N_h}$ are the discrete counterparts of the operators $\mathcal{L}$, $\mathcal{N}$ and the source term f, respectively.

Equipped with some approach for time integration, solving the parametrized time-dependent nonlinear problem (2) requires the assembly and solution of a number of linear systems. The dimension of such linear systems, i.e. N_h, is determined by both the underlying mesh and the polynomial order of the discretization scheme. The high-fidelity simulation for a complex real-world problem often requires a large number of DOFs and a lot of time steps and iterations, suggesting that such a full-order model is not affordable in many-query or real-time context of parametrized problems.

The RB method is a reliable and efficient tool for the model order reduction of parametrized problems. Spanned by a set of parameter-independent RB functions, a reduced space is constructed during the offline stage, and the approximate solution to the parameterized problem is sought online using this reduced space. From a collection of high-fidelity snapshots at different parameter values, the RB functions are carefully chosen either by the Greedy algorithm, or by the principal component analysis of the snapshots. The former requires an error estimator/indicator for the full-order solution, and picks the snapshot that maximizes the estimator/indicator until a criteria is satisfied. For time-dependent problems, a combination of the POD over time and the Greedy procedure in parameter domain can be used. In this work, however, suitable error estimators or indicators are not available for a general nonlinear time-dependent problem over the parameter domain, so the proper orthogonal decomposition (POD) is used to extract the RB, as detailed in the following.

To evaluate the reduced-order solution for any desired time-parameter location, a regression-based approach will be utilized in this work, rather than the conventional Galerkin-projection-based approach.

2.1. The database of the full-order solutions

For a parametrized time-dependent problem, the notion of a solution manifold $\mathcal{M}$ can be introduced, comprising all the solutions of (1) under variation of the time and parameters, i.e. $\mathcal{M} = \{u(t;\boldsymbol{\mu}) : (t,\boldsymbol{\mu}) \in \mathcal{T}\times\mathcal{P}\}$, and its discrete counterpart $\mathcal{M}_h = \{\mathbf{u}_h(t;\boldsymbol{\mu}) : (t,\boldsymbol{\mu}) \in \mathcal{T}\times\mathcal{P}\} \subset \mathbb{R}^{N_h}$.

Due to the data-driven nature of this work, a set of full-order solutions is prepared as a database prior to the construction of the reduced-order model. Here the database is represented as

$$\begin{aligned}\mathcal{D} &= \{\{(t,\boldsymbol{\mu}), \mathbf{u}_h(t;\boldsymbol{\mu})\} : (t,\boldsymbol{\mu}) \in \mathcal{T}_d \times \mathcal{P}_d\}\,,\\ \mathcal{T}_d &= \{t_1, t_2, \cdots, t_{N_t^d}\} \subset \mathcal{T}\,,\\ \mathcal{P}_d &= \{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \cdots, \boldsymbol{\mu}^{N_\mu^d}\} \subset \mathcal{P}\,,\end{aligned} \tag{3}$$

in which the time series are obtained by some selected time discretization scheme. Note that the discrete time-parameter inputs $(t,\boldsymbol{\mu})$, corresponding to the elements in this database, form a tensor grid between a point-set in the time domain and one in the parameter domain, i.e. $\mathcal{T}_d$ and $\mathcal{P}_d$. Usually, $\mathcal{T}_d$ contains a series of time steps in the full-order simulation, and $\mathcal{P}_d$ can be a uniform lattice or generated from the parameter domain. We assume that the database includes the essential information about the solution manifold $\mathcal{M}_h$.

2.2. The proper orthogonal decomposition and the reduced basis space

To generate the RB, we consider a collection of snapshots

$$\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{N^s}\} = \{\mathbf{u}_h(t;\boldsymbol{\mu}) : (t,\boldsymbol{\mu}) \in \Theta\} \tag{4}$$

associated with a discrete point-set $\Theta \subset \mathcal{T}_d \times \mathcal{P}_d$. A subspace of $\mathbb{R}^{N_h}$ can be spanned by the snapshots as

$$\mathcal{M}_\Theta = \mathrm{span}\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{N^s}\} \subset \mathbb{R}^{N_h}\,. \tag{5}$$

If the point-set Θ is fine enough, $\mathcal{M}_\Theta$ acts as a good representation of $\mathcal{M}_h$.

To reduce the model, a low-rank approximation with rank $L \ll \min\{N_h, N_s\}$ should be found for $\mathcal{M}_\Theta$. Towards this end, the POD is employed to extract a set of orthogonal bases $\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2|\cdots|\mathbf{v}_L] \in \mathbb{R}^{N_h \times L}$ from the snapshots, so that the column space of $\mathbf{V}$, denoted by $\mathrm{Col}(\mathbf{V})$, served as a reduced-order replacement for the full-order space $\mathbb{R}^{N_h}$. Then we assume that the reduced-order solution for time-parameter location $(t,\boldsymbol{\mu}) \in \mathcal{T} \times \mathcal{P}$, denoted as $\mathbf{u}_{\mathrm{rb}}(t;\boldsymbol{\mu})$, is represented as a linear combination of the bases $\mathbf{V}$, i.e. $\mathbf{u}_{\mathrm{rb}}(t;\boldsymbol{\mu}) = \sum_{l=1}^{L} q_l(t;\boldsymbol{\mu})\mathbf{v}_l = \mathbf{V}\mathbf{q}(t;\boldsymbol{\mu})$, where $\mathbf{q} = \{q_1, q_2, \cdots, q_L\}^{\mathrm{T}} \in \mathbb{R}^L$ collects the combination coefficients.

The *snapshot matrix* $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$, collecting all the snapshots, is defined as

$$\mathbf{S} = [\ \mathbf{s}_1 \mid \mathbf{s}_2 \mid \cdots \mid \mathbf{s}_{N_s}\ ]\,. \tag{6}$$

Note that $\mathrm{Col}(\mathbf{S}) = \mathcal{M}_\Theta$. The POD takes advantage of the *singular value decomposition* (SVD) of $\mathbf{S}$, given as

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^{\mathrm{T}} \tag{7}$$

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 $\boldsymbol{\Sigma} = \mathrm{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_{N_s}\}$ contains the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{N_s} \geq 0$.

At the algebraic level, one seeks to find the 'best' approximation of $\mathrm{Col}(\mathbf{S})$ in some optimal sense, among all L-dimensional subspaces with $L \leq \mathrm{rank}(\mathbf{S})$. Let $\mathbf{V} \in \mathbb{R}^{N_h \times L}$ be taken as 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 the snapshots onto the orthogonal bases $\mathbf{W} \in \mathbb{Y}_L$, measured in the Euclidean norm, can be expressed as $\sum_{i=1}^{N_s} \|\mathbf{s}_i - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{s}_i\|_{\mathbb{R}^{N_h}}^2$.

The Schmidt-Eckart-Young theorem [12, 28, 32] states that the basis consisting of the first L left singular vectors of $\mathbf{S}$ minimizes the projection error of the snapshots among all L-dimensional orthogonal bases in $\mathbb{R}^{N_h}$. The error can be evaluated by the $(L+1)$th to N_sth singular values, i.e.

$$\sum_{i=1}^{N_s} \left\|\mathbf{s}_i - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{s}_i\right\|_{\mathbb{R}^{N_h}}^2 = \min_{\mathbf{W}\in\mathbb{Y}_L} \sum_{i=1}^{N_s} \left\|\mathbf{s}_i - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{s}_i\right\|_{\mathbb{R}^{N_h}}^2 = \sum_{i=L+1}^{N_s} \sigma_i^2\,. \tag{8}$$

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

$$\frac{\sum_{i=1}^{N_s} \left\| \mathbf{s}_i - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{s}_i \right\|_{\mathbb{R}^{N_h}}^2}{\sum_{i=1}^{N_s} \|\mathbf{s}_i\|_{\mathbb{R}^{N_h}}^2} = \frac{\sum_{i=L+1}^{N_s} \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} \,. \tag{9}$$

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

2.3. Regression-based approach for reduced-order solutions

The computational efficiency of the RB method relies on the decoupling of the offline and online stages. As discussed, the RB modes are prepared offline from the high-fidelity snapshots and are parameter-independent. The reduced-order solution for a new parameter value is recovered in the online stage. In the conventional framework of the RB method, a standard Galerkin approach is used to determine the combination coefficients of the RB. Due to the non-affinity in parameter dependence, however, a Galerkin-projection-based scheme will not significantly save the computational cost for a general nonlinear problem. Moreover, there are online stability issues for time-dependent problem in many cases, as already discussed in the introduction.

To overcome these difficulties, a regression-based approach is used to calculate the reduced-order solutions for new time-parameter values. 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 the algebraic level,

$$\mathbf{u}_{\mathrm{rb}}(t;\boldsymbol{\mu}) = \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}_h(t;\boldsymbol{\mu}) = \arg\min_{\mathbf{w}_h \in \mathrm{Col}(\mathbf{V})} \|\mathbf{u}_h(t;\boldsymbol{\mu}) - \mathbf{w}_h\|_{\mathbb{R}^{N_h}} \,, \tag{10}$$

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

To obtain the projection coefficients $\mathbf{q}(t;\boldsymbol{\mu})$ for any desired time-parameter location $(t,\boldsymbol{\mu}) \in \mathcal{T}\times\mathcal{P}$, one can resort to a nonlinear regression $\hat{\mathbf{q}}$ between $d+1 = \dim(\mathcal{P})+1$ inputs and L outputs:

$$(t,\boldsymbol{\mu}) \mapsto \mathbf{q}(t;\boldsymbol{\mu}) = \mathbf{V}^{\mathrm{T}}\mathbf{u}_h(t;\boldsymbol{\mu}) \approx \hat{\mathbf{q}}(t;\boldsymbol{\mu}) \,. \tag{11}$$

During the offline stage, this regression model $\hat{\mathbf{q}}(\cdot;\cdot)$ should be constructed from a set of training data

$$\begin{aligned} \mathcal{D}_{tr} &= \left\{ \{(t,\boldsymbol{\mu}), \mathbf{V}^{\mathrm{T}}\mathbf{u}_h(t;\boldsymbol{\mu})\} : t \in \mathcal{T}_{tr},\ \boldsymbol{\mu} \in \mathcal{P}_{tr} \right\} , \\ \mathcal{T}_{tr} &= \{t_i :\ i = n_1, n_2, \cdots, n_{N_t^{tr}}\} \subset \mathcal{T}_d \,, \\ \mathcal{P}_{tr} &= \{\boldsymbol{\mu}^j :\ j = m_1, m_2, \cdots, m_{N_\mu^{tr}}\} \subset \mathcal{P}_d \,. \end{aligned} \tag{12}$$

with $1 \le n_1 \le n_2 \le \cdots \le n_{N_t^{tr}} \le N_t^d$ and $1 \le m_1 \le m_2 \le \cdots \le m_{N_\mu^{tr}} \le N_\mu^d$. The model is used during the online stage to recover the output $\hat{\mathbf{q}}(t^*;\boldsymbol{\mu}^*)$ for any new input $(t^*,\boldsymbol{\mu}^*) \in \mathcal{T}\times\mathcal{P}$. Correspondingly, the reduced-order solution $\mathbf{u}_{\mathrm{rb,reg}}(t^*,\boldsymbol{\mu}^*) \in \mathrm{Col}(\mathbf{V})$ is

$$\mathbf{u}_{\mathrm{rb,reg}}(t^*;\boldsymbol{\mu}^*) = \mathbf{V}\hat{\mathbf{q}}(t^*;\boldsymbol{\mu}^*) \,. \tag{13}$$

Once the regression model is recovered, the online stage only requires direct outputs from the regression models, i.e. the online solutions are obtained at very low cost.

Algorithm 1 Regression-based RB method for time-dependent problems

Offline stage:
Prepare the database $\mathcal{D}$ of full-order solutions;
Extract some snapshots from the database $\mathcal{D}$ and form the snapshot matrix $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$;
Perform POD for $\mathbf{S}$ and get the L orthogonal bases $\mathbf{V} \in \mathbb{R}^{N_h \times L}$;
Prepare the training set $\mathcal{D}_{tr}$ from the database $\mathcal{D}$;
Construct the regression model $\hat{\mathbf{q}}(\cdot;\cdot)$ from $\mathcal{D}_{tr}$.

Online stage:
Recover output $\hat{\mathbf{q}}(t^*;\boldsymbol{\mu}^*)$ for a new parameter value $(t^*,\boldsymbol{\mu}^*)$;
Evaluate the reduced-order solution $\mathbf{u}_{\mathrm{rb,reg}}(t^*;\boldsymbol{\mu}^*) = \mathbf{V}\hat{\mathbf{q}}(t^*;\boldsymbol{\mu}^*)$.

To meet the high demand on the quantity of data, all samples in the database are usually used when training the regression models. We note the complete decoupling of the offline and online stages, and the non-intrusive nature of the regression-based RB method. In this work, Gaussian process models are utilized to construct the regression $\hat{\mathbf{q}}(\cdot)$, as discussed in the following section.

3. Gaussian process regression models

Regression is concerned with prediction of continuous quantities of interest by the construction of a model from a set of observation data. Let $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \cdots, M\}$ 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 [29, 35], the observed input-output pairs are assumed to follow some regression function $f : \mathcal{P} \to \mathbb{R}$, the prior of which is defined as a Gaussian process (GP). Given the training data, the model uses the posterior GP to make predictions for new inputs.

3.1. 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 corrupted by an independent Gaussian noise term, i.e. for $(\mathbf{x}, \mathbf{x}') \in \mathcal{P} \times \mathcal{P}$,

$$f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))\,, \quad y = f(\mathbf{x}) + \epsilon\,, \quad \epsilon \sim \mathcal{N}(0, \sigma_y^2)\,. \tag{14}$$

Here $m(\mathbf{x}) := \boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}(\mathbf{x})$, $\mathbf{H}(\mathbf{x}) = \{H_1(\mathbf{x}), H_2(\mathbf{x}), \cdots, H_N(\mathbf{x})\}^{\mathrm{T}}$ contains N basis functions defined in $\mathcal{P}$, and $\boldsymbol{\beta} = \{\beta_1, \beta_2, \cdots, \beta_N\}^{\mathrm{T}}$ are the corresponding combination coefficients. There are many different options for the covariance function $\kappa : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$. A frequently used one is the *automatic relevance determination* (ARD) *squared exponential* (SE) *kernel*:

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{m=1}^{d}\frac{(x_m - x'_m)^2}{\ell_m^2}\right)\,, \tag{15}$$

which considers an individual correlated lengthscale ℓ_m 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, referred to as the parameter locations of training data, a prior joint Gaussian is defined for the regression outputs:

$$\mathbf{y}|\mathbf{X} \;\sim\; \mathcal{N}(m(\mathbf{X}), \mathbf{K}_y)\,, \quad \mathbf{K}_y = \mathrm{cov}[\mathbf{y}|\mathbf{X}] = \kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2\mathbf{I}_M\,, \tag{16}$$

where $\mathbf{y} = \{y_1, y_2, \cdots, y_M\}^{\mathrm{T}}$, $\mathbf{X} = [\ \mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_M\ ]$ and $\mathbf{I}_M$ is the M-dimensional unit matrix.

From a regression model, the goal is prediction of the noise-free output $f^*(\mathbf{s})$ for a new test input $\mathbf{s} \in \mathcal{P}$. Then one can combine the information from training set with the predictions for test samples in

a joint density of the observed outputs $\mathbf{y}$ and the noise-free test output $f^*(\mathbf{s})$. By the standard rules for conditioning Gaussians, the posterior predictive distribution can be obtained as a new GP:

$$\begin{aligned} &f^*(\mathbf{s})|\ \mathbf{s},\mathbf{X},\mathbf{y}\ \sim\ \mathrm{GP}(m^*(\mathbf{s}),c^*(\mathbf{s},\mathbf{s}'))\,,\\ &m^*(\mathbf{s})=m(\mathbf{s})+\kappa(\mathbf{s},\mathbf{X})\mathbf{K}_y^{-1}(\mathbf{y}-m(\mathbf{X}))\,,\quad c^*(\mathbf{s},\mathbf{s}')=\kappa(\mathbf{s},\mathbf{s}')-\kappa(\mathbf{s},\mathbf{X})\mathbf{K}_y^{-1}\kappa(\mathbf{X},\mathbf{s}')\,, \end{aligned} \tag{17}$$

The values of the hyperparameters $\boldsymbol{\theta}$ make significant difference on the predictive performance, with $\boldsymbol{\theta}=\{\beta_1,\cdots,\beta_N,\ell_1,\cdots,\ell_d,\sigma_f,\sigma_y\}$ for the case of ARD SE kernel. In this paper, 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, the optimal hyperparameters $\boldsymbol{\theta}_{\mathrm{opt}}$ can be estimated via the maximization problem:

$$\begin{aligned} \boldsymbol{\theta}_{\mathrm{opt}}&=\arg\max_{\boldsymbol{\theta}}\ \log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})\\ &=\arg\max_{\boldsymbol{\theta}}\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}(\mathbf{X}))^{\mathrm{T}}\mathbf{K}_y^{-1}(\boldsymbol{\theta})(\mathbf{y}-\boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}(\mathbf{X}))-\frac{1}{2}\log|\mathbf{K}_y(\boldsymbol{\theta})|-\frac{M}{2}\log(2\pi)\right\}\,, \end{aligned} \tag{18}$$

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

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

We only use one constant basis function in this paper, i.e. $\mathbf{H}(\mathbf{x})=H(\mathbf{x})=1$, so there exists only $N=1$ combination coefficient β. In the numerical implementations for time-dependent problems, moreover, the predictive mean function $m^*(\cdot)$ is of the most interest.

3.2. Tensor-decomposition-based regression

In this work, the regression is equipped with tensor decomposition to ensure reliable and efficient fitting results. For the lth entry $q_l=\mathbf{v}_l^{\mathrm{T}}\mathbf{u}_h$ of the projection coefficients $\mathbf{q}=\{q_1,q_2,\cdots,q_L\}^{\mathrm{T}}$, $l=1,2,\cdots,L$, the training data can be written in a matrix as

$$\mathbf{P}_l=[q_l(t_{n_i};\boldsymbol{\mu}^{m_j})]_{ij}\,,\quad 1\le i\le N_t^{tr}\,,\ 1\le j\le N_\mu^{tr}\,, \tag{19}$$

as a natural result of the tensor grid between the time and parameter locations in the training data.

To decompose the data of a projection coefficient into several time- and parameter-modes, the SVD is employed again as

$$\mathbf{P}_l\approx\tilde{\mathbf{P}}_l=\sum_{k=1}^{Q_l}\lambda_k^l\boldsymbol{\psi}_k^l(\boldsymbol{\phi}_k^l)^{\mathrm{T}}\,. \tag{20}$$

Here $\boldsymbol{\psi}_k^l$ and $\boldsymbol{\phi}_k^l$ are the kth discrete modes of time and parameters for the lth projection coefficient, respectively, λ_k^l is the kth singular value for the same coefficient, and Q_l is the corresponding rank of truncation.

From the data of discrete modes, Gaussian processes are trained to approximate the corresponding continuous modes, as follows

$$\begin{aligned} &t\mapsto\hat{\psi}_k^l(t)\quad\text{trained from }\{(t_{n_i},(\boldsymbol{\psi}_k^l)_i):\ i=1,2,\cdots,N_t^{tr}\}\,,\\ &\boldsymbol{\mu}\mapsto\hat{\phi}_k^l(\boldsymbol{\mu})\quad\text{trained from }\{(\boldsymbol{\mu}^{m_j},(\boldsymbol{\phi}_k^l)_j):\ j=1,2,\cdots,N_\mu^{tr}\}\,, \end{aligned} \tag{21}$$

where $\hat{\psi}_k^l(t)$ and $\hat{\phi}_k^l(\boldsymbol{\mu})$ are the kth continuous time- and parameter-modes for the lth projection coefficient, respectively. Hence we have

$$(\mathbf{P}_l)_{ij}=q_l(t_{n_i};\boldsymbol{\mu}^{m_j})\approx\sum_{k=1}^{Q_l}\lambda_k^l\hat{\psi}_k^l(t_{n_i})\hat{\phi}_k^l(\boldsymbol{\mu}^{m_j})\,,\quad 1\le i\le N_t^{tr}\,,\ 1\le j\le N_\mu^{tr}\,.$$

A continuous regression function $\hat{q}_l$ for the lth projection coefficient q_l with respect to time-parameter values can be recovered as

$$q_l(t;\boldsymbol{\mu}) \approx \hat{q}_l(t;\boldsymbol{\mu}) = \sum_{k=1}^{Q_l} \lambda_k^l \hat{\psi}_k^l(t)\hat{\phi}_k^l(\boldsymbol{\mu}), \quad (t,\boldsymbol{\mu}) \in \mathcal{T}\times\mathcal{P}, \tag{22}$$

which shares a similar formulation with (20).

We would like to comment on the advantages of the tensor-decomposition-based regression approach in the context of time-dependent problems. Usually, the projection coefficients are evolving substantially with respect to time, leading to difficult (global) GPRs. After the tensor decomposition, the 1D regressions with respect to time, combined with the multi-dimensional regressions with respect to the parameters, are both easier. For the lth projection coefficient, there are $N+d+3$ hyperparameters in the global regression model, while $Q_l(2N+d+5)$ in the tensor-decomposition-based one. Much more hyperparameters in the tensor-decomposition-based approach imply a more flexible scheme for function fitting.

As can be noticed, the major computational effort for obtaining the posterior GP is to calculate $\mathbf{K}_y^{-1}$, the complexity of which is $\mathcal{O}((N_t^{tr}N_\mu^{tr})^3)$ for the global regression and $Q_l[\mathcal{O}((N_t^{tr})^3)+\mathcal{O}((N_\mu^{tr})^3)]$ for the alternative. Thus the tensor decomposition significantly reduces the computational cost of the regression of each projection coefficient, even though extra effort of the SVD with complexity $\mathcal{O}(N_t^{tr}N_\mu^{tr}\max\{N_t^{tr},N_\mu^{tr}\})$ is required. However, this is usually not expensive.

3.3. Error control for the truncations

In this subsection, we suppose the database $\mathcal{D}$ is completely used, i.e. all samples have been used both as snapshots for constructing the RB and for training samples of regressions, i.e.

$$\{\mathbf{u}_h(t;\boldsymbol{\mu}) : (t,\boldsymbol{\mu}) \in \mathcal{T}_d\times\mathcal{P}_d\} = \{\mathbf{u}_h(t;\boldsymbol{\mu}) : (t,\boldsymbol{\mu}) \in \mathcal{S}\} = \{\mathbf{u}_h(t;\boldsymbol{\mu}) : (t,\boldsymbol{\mu}) \in \mathcal{T}_{tr}\times\mathcal{P}_{tr}\} := \mathcal{F},$$

where $\mathcal{F}$ collects all the full-order solution vectors in the database.

Following the truncations in the SVDs for the projection coefficients, a relative error for the recovery can be given as:

$$\begin{aligned}
\bar{e}^2_{\mathrm{rcv}}(\mathcal{F}) &= \frac{\sum_{i=1}^{N_t^d}\sum_{j=1}^{N_\mu^d}\|\mathbf{u}_h(t_i;\boldsymbol{\mu}^j) - \sum_{l=l}^{L}(\tilde{\mathbf{P}}_l)_{ij}\mathbf{v}_l\|^2_{\mathbb{R}^{N_h}}}{\sum_{\mathbf{u}\in\mathcal{F}}\|\mathbf{u}\|_2^2} \\
&= \frac{1}{\sum_{\mathbf{u}\in\mathcal{F}}\|\mathbf{u}\|^2_{\mathbb{R}^{N_h}}}\left[\sum_{\mathbf{u}\in\mathcal{F}}\|\mathbf{u}-\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}\|^2_{\mathbb{R}^{N_h}} + \sum_{i=1}^{N_t^d}\sum_{j=1}^{N_\mu^d}\sum_{l=1}^{L}\left((\mathbf{P}_l)_{ij}-(\tilde{\mathbf{P}}_l)_{ij}\right)^2\right] \\
&= \varepsilon_L^2 + \frac{\sum_{l=1}^{L}\|\mathbf{P}_l-\tilde{\mathbf{P}}_l\|_F^2}{\sum_{\mathbf{u}\in\mathcal{F}}\|\mathbf{u}\|^2_{\mathbb{R}^{N_h}}} = \varepsilon_L^2 + \frac{\sum_{l=1}^{L}\|\mathbf{P}_l\|_F^2\ \delta_l^2}{\sum_{\mathbf{u}\in\mathcal{F}}\|\mathbf{u}\|^2_{\mathbb{R}^{N_h}}} \\
&= \varepsilon_L^2 + \sum_{l=1}^{L}\frac{\|\mathbf{P}_l\|_F^2}{\sum_{\mathbf{u}\in\mathcal{F}}\|\mathbf{u}\|^2_{\mathbb{R}^{N_h}}}\delta_l^2 = \varepsilon_L^2 + \sum_{l=1}^{L}\eta_l^2\delta_l^2,
\end{aligned} \tag{23}$$

where $\hat{\mathbf{P}}_l$ represents the coefficients after the SVD truncation, as defined in (20). Here δ_l^2 stands for the truncation error of the SVD for the lth projection coefficient, i.e. $\delta_l^2 = \|\mathbf{P}_l-\tilde{\mathbf{P}}_l\|_F^2/\|\mathbf{P}_l\|_F^2$, η_l^2 is defined as $\sigma_l^2/\sum_{l=1}^{N_s}\sigma_l^2$, σ_l is the lth singular value in the POD, used to construct the RB, and $\varepsilon_L^2 = 1-\sum_{l=1}^{L}\eta_l^2$ gives the truncation error by the POD. In the data-driven context, (23) can be used to evaluate the truncation errors due to both the POD for constructing the RB and the SVDs for the training data of projection coefficients. This provides an estimator for the corresponding error control.

3.4. An error surrogate model

Regression have also been used to approximate the errors introduced by reduced-order models. As presented in [11], the errors in quantities of interest are modelled statistically, and the GPR is employed

to map from a small number of inexpensive error indicators to a distribution over the true error. Recently in [33], some regression functions are directly constructed between the input parameters and the errors in both the state and quantities of interest. Based on a similar idea, we give a regression model as an error surrogate to assess or certify the quality of the proposed reduced model.

For each time-parameter location $(t_{n_i}, \boldsymbol{\mu}^{m_j})$ in the training set, $i = 1, 2, \cdots, N_t^{tr}$, $j = 1, 2, \cdots, N_\mu^{tr}$, the relative error of the reduced-order solution is given as

$$e_r(t_{n_i}, \boldsymbol{\mu}^{m_j}) = \frac{1}{\|\mathbf{u}_h(t_{n_i}; \boldsymbol{\mu}^{m_j}))\|_{\mathbb{R}^{N_h}}} \|\mathbf{u}_h(t_{n_i}; \boldsymbol{\mu}^{m_j})) - \mathbf{V}\mathbb{E}[\hat{\mathbf{q}}(t_{n_i}; \boldsymbol{\mu}^{m_j})]\|_{\mathbb{R}^{N_h}} \ , \tag{24}$$

which can be evaluated once all the GPR models are constructed. Another regression model can be trained as an error surrogate, denoted by $\hat{e}_r(\cdot, \cdot) : \mathcal{T} \times \mathcal{P} \to \mathbb{R}$, i.e.

$$(t, \boldsymbol{\mu}) \mapsto \hat{e}_r(t, \boldsymbol{\mu}) \quad \text{trained from } \{(t_{n_i}, \boldsymbol{\mu}^{m_j}, e_r(t_{n_i}, \boldsymbol{\mu}^{m_j})) : i = 1, 2, \cdots, N_t^{tr},\ j = 1, 2, \cdots, N_\mu^{tr}\} . \tag{25}$$

In this work, the error surrogate is constructed as a Gaussian process, and the outputs are given as distributions, such that the predictive error indicator for any time-parameter location is represented by a mean value bounded by confidence bounds. Usually, the relative error of a reduced-order solution with respect to time and parameters is a much more complex function than the projection coefficients onto the RB. As noticed, the training for error surrogate shares the same sampling locations as those for the projection coefficients, thus one should not expect too much about the accuracy of its predictive mean function. Even so, one can resort to the confidence level for non-rigorous bounds of the relative error.

4. Numerical examples

In this section, numerical results of three time-dependent or pseudo-time-dependent examples are presented. The first one is a simple problem of 1-D Burgers' equation with parametrized viscosity. Both the constructed RB and the reduced-order solutions are illustrated, and the effectiveness of the proposed framework is validated. The second example is a benchmark problem of incompressible fluid flow around a cylinder. Regression results in the offline stage are plotted, and some online tests are performed to compare the accuracy of reduced-order solutions with that of full-order ones. Error surrogates are constructed as GPR models to provide error indicators for the online evaluations. Moreover, by setting a series of carefully defined criteria for the SVDs of the training data, offline regression models are constructed with further reduced computational cost and guaranteed online accuracy. In the last example, a structural problem with large deformation is treated as pseudo-time-dependent to build a reduced model which can recover both the solution fields and a class of equilibrium paths with respect to several parameters. The GPR models in these examples are constructed by the MATLAB function `fitrgp`.

4.1. Example 1: 1-D viscous Burgers' equation

As a simple test, a one-dimensional viscous Burgers' equation with parameterized diffusion coefficient is taken into consideration, given as

$$\begin{aligned} &u_t + uu_x - \frac{\mu}{50\pi} u_{xx} = 0 \,, \quad (x, t, \mu) \in (-1, 1) \times (0, 1] \times [1, 7.5] \,, \\ &u(-1, t; \mu) = u(1, t; \mu) = 0 \,, \quad u(x, 0; \mu) = -\sin(\pi x) \,. \end{aligned}$$

In the offline stage, full-order solutions are calculated by the finite difference method with $N_h = 201$ nodes at $N_\mu^d = 27$ uniformly distributed parameter locations of μ, i.e. $\mathcal{P}_d = \{1.00, 1.25, 1.50, \cdots, 7.25, 7.5\}$. Among the 1000 time steps, $N_t^d = 250$ uniformly distributed ones are included in the database, i.e. $\mathcal{T}_d = \{0.004, 0.008, 0.012, \cdots, 0.996, 1.000\}$. Then all the full-order solution vectors are used both as snapshots for constructing the RB and training data for the GPR models. By the POD, a set of $L = 7$ orthogonal modes are extracted as the RB with a tolerance $\varepsilon_{L,\mathtt{tol}} = 0.1\%$, as shown in Figure 1. As proposed previously,

regression models are then constructed as the combinations of time- and parameter-modes, which are trained as GPs after the SVD truncations with tolerances $\delta_{l,\texttt{tol}} = 0.1\%$, $1 \leq l \leq L = 7$.

To validate the accuracy of the reduced model, the reduced-order solutions are recovered at 5 time points with respect to 4 different values of μ, and are compared with the corresponding full-order solutions in Figure 2. It is seen that the full-order and reduced-order solutions are matching well with each other, even when the viscous parameter μ is quite small, confirming the effectiveness of the proposed method.

4.2. Example 2: incompressible fluid flow around a cylinder

In this example, we consider a classic benchmark in CFD – the fluid flow around a cylinder. As shown in Figure 3, the problem is defined in a domain $\Omega = [0, 2.2] \times [0, 0.41] \setminus B_r(0.2, 0.2)$ with $r = 0.05$. The governing equations are the non-stationary Navier-Stokes equations for an incompressible fluid:

$$\rho \boldsymbol{u}_t - \nu \Delta \boldsymbol{u} + \rho \boldsymbol{u} \nabla \boldsymbol{u} + \nabla p = \boldsymbol{0}, \quad \nabla \cdot \boldsymbol{u} = 0\,.$$

Here we take the fluid density as $\rho = 1.0$ and the viscosity as $\nu = 0.001$. No-slip conditions are set along the lower and upper walls and $\partial B_r(0.2, 0.2)$, and Neumann conditions are set along the right (outflow) edge. Moreover, the following parabolic velocity profile is prescribed on the left (inflow) edge:

$$\boldsymbol{u}(0, y) = \left[\frac{4Uy(0.41 - y)}{0.41^2}, 0\right]^{\mathrm{T}},$$

in which the value of parameter U is subject to change in an interval $[1.125, 1.5]$. Correspondingly, the Reynolds number Re lies in an interval $\mathcal{P} = [75, 100]$, under the definition of $Re = U_m L/\nu$, where the mean velocity is given as $U_m = 2U/3$ and the characteristic length of the flow configuration is $L = 0.1$. Reduced models will be constructed for the solution history in time domain $\mathcal{T} = [2.5, 5.0]$.

More details about this benchmark problem can be found at `http://www.featflow.de/`, including some visualizations of both the velocity solution $\boldsymbol{u}$ and the pressure solution p. In this example, we use the finite element [5] solver in the MATLAB library `redbKIT v2.2` [22, 28] to calculate the full-order solutions. At 51 different values $\mathcal{P}_d = \{75.0, 75.5, 76.0, \cdots, 99.5, 100.0\}$ of the parameter Re, we use a small time step to get the high-fidelity time series of the solutions, for the velocity and the pressure, and add the full-order solutions at 126 time points $t \in \mathcal{T}_d = \{2.50, 2.52, 2.54, \cdots, 4.98, 5.00\}$ into the database $\mathcal{D}$. All these samples will be used both as snapshots and training data.

After the POD with a truncation tolerance $\varepsilon_{L,\texttt{tol}} = 0.5\%$, $L = 36$ reduced bases are extracted for the velocity solution and $L = 33$ for the pressure solution. For comparison we have the full orders $N_h = 29318$ for velocity and $N_h = 3899$ for pressure.

Case 1:

In this case, the truncation criteria of the SVDs for all projection coefficients are set as $Q_1 = 10$ and $\delta_{l,\texttt{tol}} = \varepsilon_{L,\texttt{tol}} = 0.5\%$ for $l = 2, 3, \cdots, L$. A more strict criterion is defined for the 1st coefficient due to its dominant position in the accuracy of reduced-order solutions.

After the tensor decompositions for the data of all projection coefficients, the time- and parameter-modes are obtained as GPR models, some of which are shown in Figures 4 and 5. The mean regression functions of all the coefficients are recovered in a similar decomposition form, and some of these regression results are plotted in Figures 6 and 7. It is worth pointing out that all the regressions for the problem can be finished within minutes.

After the offline training, online tests are performed on the solutions for 4 new Re values that are not included in the offline database, chosen as $Re \in \{80.25,\ 85.25,\ 90.25,\ 95.25\}$. The reduced-order time-dependent for these 4 parameter values are calculated as combinations of the RB modes, with their coefficients obtained as direct outputs from the regression models. Compared with the corresponding full-order solutions, the relative errors, measured in 2-norm, are calculated and plotted in Figure 8. As can be noticed, the relative error are quite close to their lower bounds – the projection errors onto the RB spaces constructed by the POD – implying a good quality of the regressions. Moreover, the coefficients for lift and drag, denoted by C_L and C_D, are also extracted from the reduced-order solutions for the 4 test cases. As

shown in Figure 9, they are matching well with the high-fidelity results from the corresponding full-order solutions.

As discussed in subsection 3.4, error surrogates are constructed as GPR models for the solutions of both the velocity field and the pressure field. For the online test samples, the predictive errors with their $\pm\sigma$ bands are shown in Figure 10. It can be seen that the true errors between the reduced-order and full-order solutions are essentially bounded by the $\pm\sigma$ bands, and they can be strictly bounded by the $\pm 2\sigma$ bands or 95% confidence levels. As surrogates for relative errors, the GPs provide non-rigorous error indicators for the reduced-order solutions.

Case 2:

According to (23), the recovery error caused by all the offline truncations, both the POD for constructing the RB and the SVDs for the data of projection coefficient, can be controlled using

$$\bar{e}^2_{\mathrm{rcv},\mathtt{tol}}(\mathcal{F}) = \varepsilon^2_{L,\mathtt{tol}} + \sum_{l=1}^{L} \eta_l^2 \delta^2_{l,\mathtt{tol}} \,. \tag{26}$$

Note that the weight η_l^2 is decaying as l increases, meaning that the lower-order projection coefficients contribute more to the recovery accuracy. Moreover, the regressions usually get more difficult when l gets larger. A natural thought is to set a more strict SVD-truncation criterion, i.e. a smaller δ_l, for a lower-order projection coefficient. In this way, the regressions are easier and more efficient, provided the accuracy is under control.

In this case, we set $\delta_{l,\mathtt{tol}} = \alpha\varepsilon_{L,\mathtt{tol}}/(\eta_l\sqrt{L})$, so that $\eta_l^2\delta^2_{l,\mathtt{tol}} = (\alpha\varepsilon_{L,\mathtt{tol}})^2$ for each $1 \le l \le L$, i.e., each term of the summation in (26) contributes equally. To ensure similar values of $\bar{e}^2_{\mathrm{rcv},\mathtt{tol}}$ as those in Case 1, we take $\alpha = 0.2$ for the velocity and $\alpha = 0.27$ for the pressure, in which case one has $\bar{e}^2_{\mathrm{rcv},\mathtt{tol}} = 1.040\varepsilon^2_{L,\mathtt{tol}}$ for the velocity and $\bar{e}^2_{\mathrm{rcv},\mathtt{tol}} = 1.073\varepsilon^2_{L,\mathtt{tol}}$ for the pressure, accordingly.

Following the same procedure of offline training and online tests as in Case 1, the reduced-order solutions for the 4 test values of Re are obtained, and their relative error are shown in Figure 11, compared with those of Case 1. It can be seen that the solutions in both cases have a similar accuracy, as a natural results of sharing similar $\bar{e}^2_{\mathrm{rcv},\mathtt{tol}}$'s. On the other side, the rank of SVD-truncations, Q_l, for both velocity and pressure in the two cases are shown in Figure 12. It is evident that the total number of regressions for time- or parameter-modes, $\sum_{l=1}^{L} Q_l$, is much reduced in Case 2, which significantly enhances the offline efficiency.

One advantage of the proposed regression-based framework is that the regression models for the projection coefficients are trained separately, and one can set an individual criterion for the construction of each model. In Case 2, we take advantage of this to further reduce the efficiency. The way of setting criteria is problem-independent and can be adopted in other numerical applications.

4.3. Example 3: equilibrium paths of a trussed frame

A frame made of a beam and a column is considered in this example. As shown in Figure 13, the frame is trussed by 596 one-dimensional hyperelastic elements and loaded by a concentrated load on the beam. The number of DOFs of the full-order model is $N_h = 476$, and the `FLagSHyP` MATLAB program [8, 9] is used as a high-fidelity solver for this problem. The quantities in Figure 13 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, the equilibrium paths, referred to as the load-displacement curve, is shown in Figure 14 together with some configurations of the frame at different loading stages.

There are two parameters μ_1 and μ_2 in this example: $\mu_1 \in [0.5, 1.5]$ 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 $\mu_2 \in [0.8, 1.2]$ is that of the lower half of column. We seek to build a reduced model for a class of equilibrium paths with respect to the two varying material parameters.

The loading increments for an equilibrium path are obtained by the arc-length method. It is natural to consider the incremental procedure as a time series, and consider the deformation of this structure along an equilibrium path as a pseudo-time-dependent problem. For the full-order database, we calculate the 100 loading increments with fixed arc-length at randomly generated $N_\mu^d = 100$ parameter locations in

$\mathcal{P} = [0.5, 1.5] \times [0.8, 1.2]$. We define $\mathcal{T} = [0, 1]$, and the $N_t^d = 101$ pseudo-time steps, representing the loading increments, are $t_i = (i-1)/(N_t^d - 1)$, $i = 1, 2, \cdots, N_t^d$. All full-order samples will be used both as snapshots and training data.

The proposed approach is employed to construct the RB and the regression models. As in Case 2 of Example 2, we adopt different criteria for the SVDs of projection coefficient data, with $\varepsilon_{L,\texttt{tol}} = 0.1\%$ and $\alpha = 1.0$. To plot the paths, one extracts the load F from the training samples and build a GPR model $\hat{F}(t, \mu_1, \mu_2)$ for this load.

Online tests are performed at 4 new parameter locations, not included in the database. One can extract the vertical displacement at the loading node, denoted as $\hat{\Delta}(t, \mu_1, \mu_2)$, from the combination of RB modes with their approximate coefficients, and plot the paths as curves of $\hat{F}(t, \mu_1, \mu_2)$ versus $\hat{\Delta}(t, \mu_1, \mu_2)$ with t eliminated. The curves for the 4 test locations of (μ_1, μ_2) are shown in Figure 15, matching very well with those obtained by full-order simulations.

In the large deformation analysis in structural mechanics, it is important, but usually challenging, to obtain the equilibrium path of a structure under some loading pattern. As shown in this example, such a problem can be dealt with by constructing a reduced order model for fast recovery of both the displacement solution fields and the equilibrium paths. It is worth noting that any state in the loading procedure and any path within the parameter domain can be recovered by the regression models.

5. Conclusions

A regression-based reduced basis method is proposed for time-dependent problems. A database of full-order solution samples is prepared offline, and provide data for the constructions of both the RB and the regression models for projection coefficients. Equipped with the SVDs to extract the principal components of the data of projection coefficients, each regression function is expressed in the form of a tensor decomposition via a series of time- and parameter-modes of GPs. The control of truncation errors is discussed, and a GPR model of an error surrogate is introduced to certify the quality of the reduced-order models. An example of 1-D viscous Burgers' equation and one of incompressible fluid flow around a cylinder are tested to validate the effectiveness of the proposed method. Moreover, a large deformation analysis of a trussed frame is considered as a pseudo-time-dependent problem with loading increment defined as the pseudo-time, and a reduced model is derived for a class of equilibrium paths with respect to parameter variation.

Due to its non-intrusive nature, the proposed method trains reduced-order models at the algebraic level, and provides fast and reliable online calculations for parametrized time-dependent problems. This technique provides a promising tool for the solution of multi-entry or real-time large-scale complex problems in science and engineering.

References

[1] B. M. Afkham, A. Bhatt, B. Haasdonk, and J. S. Hesthaven. Symplectic model-reduction with a weighted inner product. *arXiv*, No. 1803.07799, 2018.

[2] B. M. Afkham and J. S. Hesthaven. Structure preserving model reduction of parametric hamiltonian systems. *SIAM Journal on Scientific Computing*, 39(6):A2616–A2644, 2017.

[3] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

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

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

[6] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

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

[8] J. Bonet, A. J. Gil, and R. D. Wood. FLagSHyP software. `http://www.flagshyp.com/`, 2016.

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

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

[11] M. Drohmann and K. Carlberg. The ROMES method for statistical modeling of reduced-order-model error. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):116–145, 2015.
[12] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
[13] M. Guo and J. S. Hesthaven. Reduced order modeling for nonlinear structural analysis using gaussian process regression. Technical report, EPFL, 2017.
[14] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.
[15] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
[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] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. A reduced computational and geometrical framework for inverse problems in hemodynamics. *International journal for numerical methods in biomedical engineering*, 29(7):741–776, 2013.
[19] 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.
[20] T. Lieu, C. Farhat, and M. Lesoinne. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer methods in applied mechanics and engineering*, 195(41-43):5730–5742, 2006.
[21] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.
[22] F. Negri. redbKIT Version 2.2. `http://redbkit.github.io/redbKIT/`, 2016.
[23] 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.
[24] N. C. Nguyen and J. Peraire. Gaussian functional regression for linear partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 287:69–89, 2015.
[25] N. C. Nguyen and J. Peraire. Gaussian functional regression for output prediction: Model assimilation and experimental design. *Journal of Computational Physics*, 309:52–68, 2016.
[26] 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.
[27] G. Pitton and G. Rozza. On the application of reduced basis methods to bifurcation problems in incompressible fluid dynamics. *Journal of Scientific Computing*, 73(1):157–177, 2017.
[28] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.
[29] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT press Cambridge, 2006.
[30] 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.
[31] A. Sartori, D. Baroli, A. Cammi, L. Luzzi, and G. Rozza. A reduced order model for multi-group time-dependent parametrized reactor spatial kinetics. In *2014 22nd International Conference on Nuclear Engineering*, pages V005T17A048–V005T17A048. American Society of Mechanical Engineers, 2014.
[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. Trehan, K. Carlberg, and L. J. Durlofsky. Error modeling for surrogates of dynamical systems using machine learning. *International Journal for Numerical Methods in Engineering*, 2017.
[34] C. Wang, J. S. Hesthaven, J. Bai, Y. Zhang, and T. Yang. A greedy non-intrusive reduced order model for fluid dynamics. Technical report, submitted to American Institute of Aeronautics and Astronautics, 2017.
[35] C. K. Williams and C. E. Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.

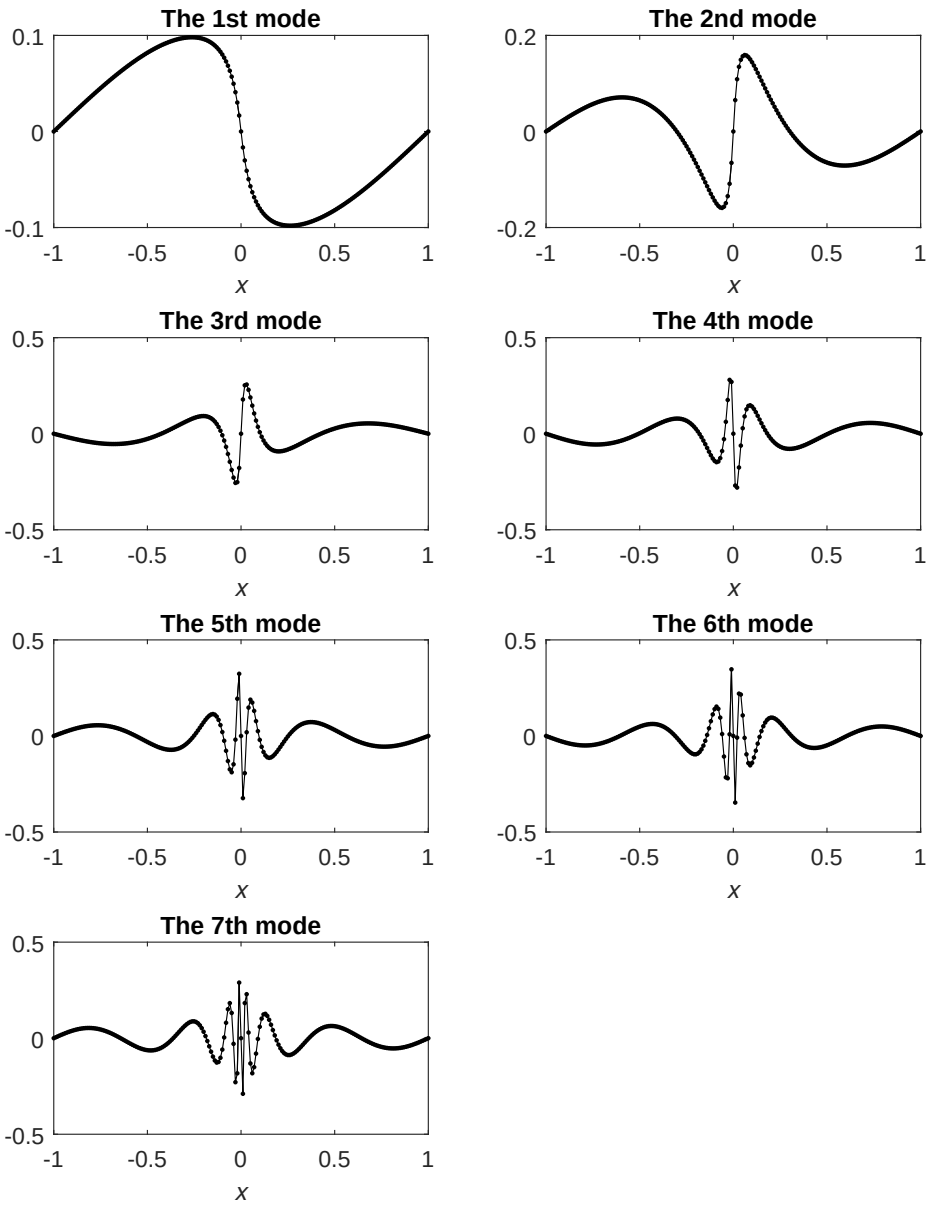


Figure 1: The RB modes for the parametrized viscous Burgers' equation

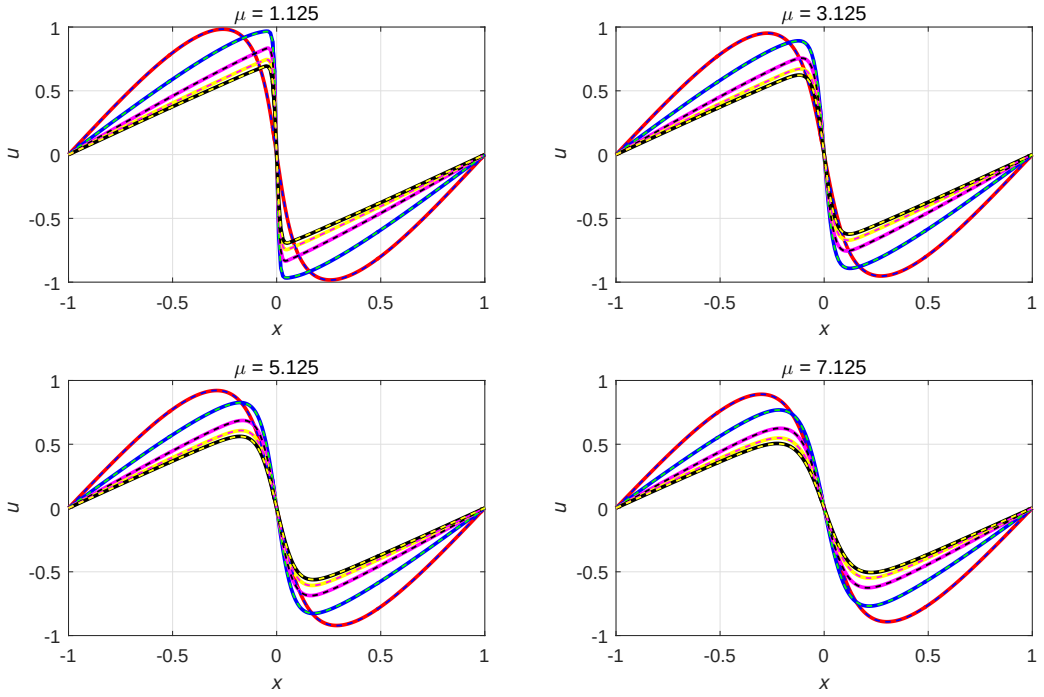


Figure 2: Comparison between the full-order solutions and the corresponding reduced-order solutions: solutions at different $t \in \{0.25, 0.50, 0.75, 0.90, 1.00\}$ with different values of μ: full-order solutions – dashed, reduced-order solution – solid.

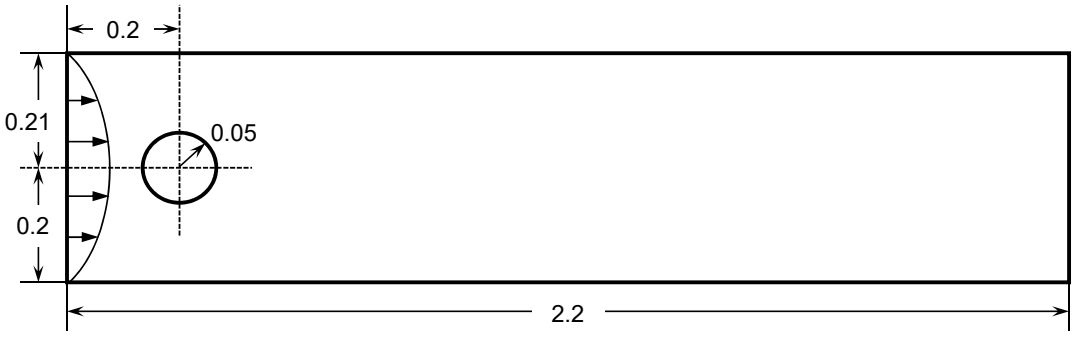


Figure 3: Underlying geometry for the problem of fluid flow around a cylinder

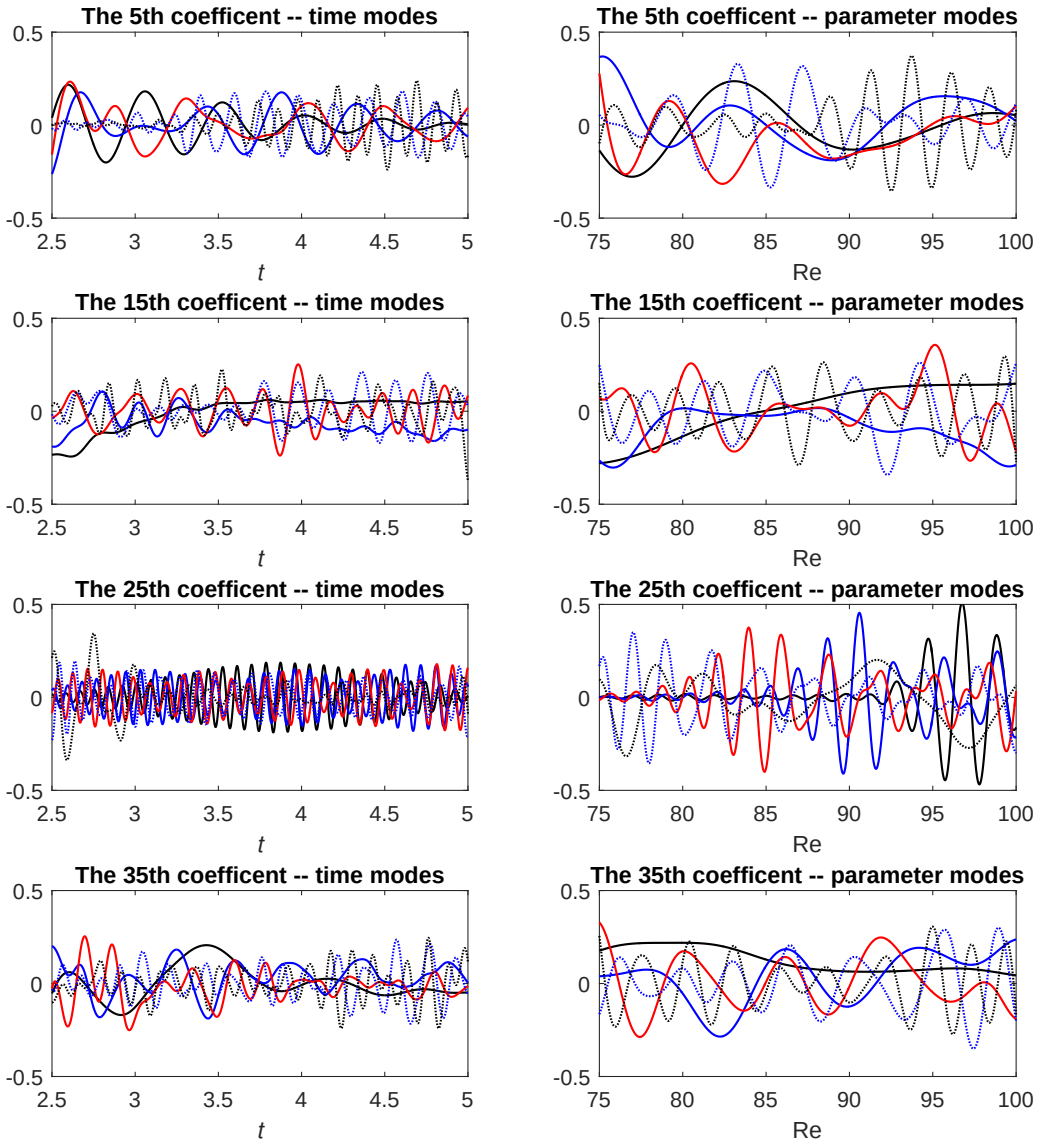


Figure 4: Time- and parameter-modes for the 5th, 15th, 25th and 35th projection coefficients for the velocity field: the 1st modes – black, the 5th modes – blue, the 10th modes – red, the 15th modes – blue dashed, the 20th modes – black dashed

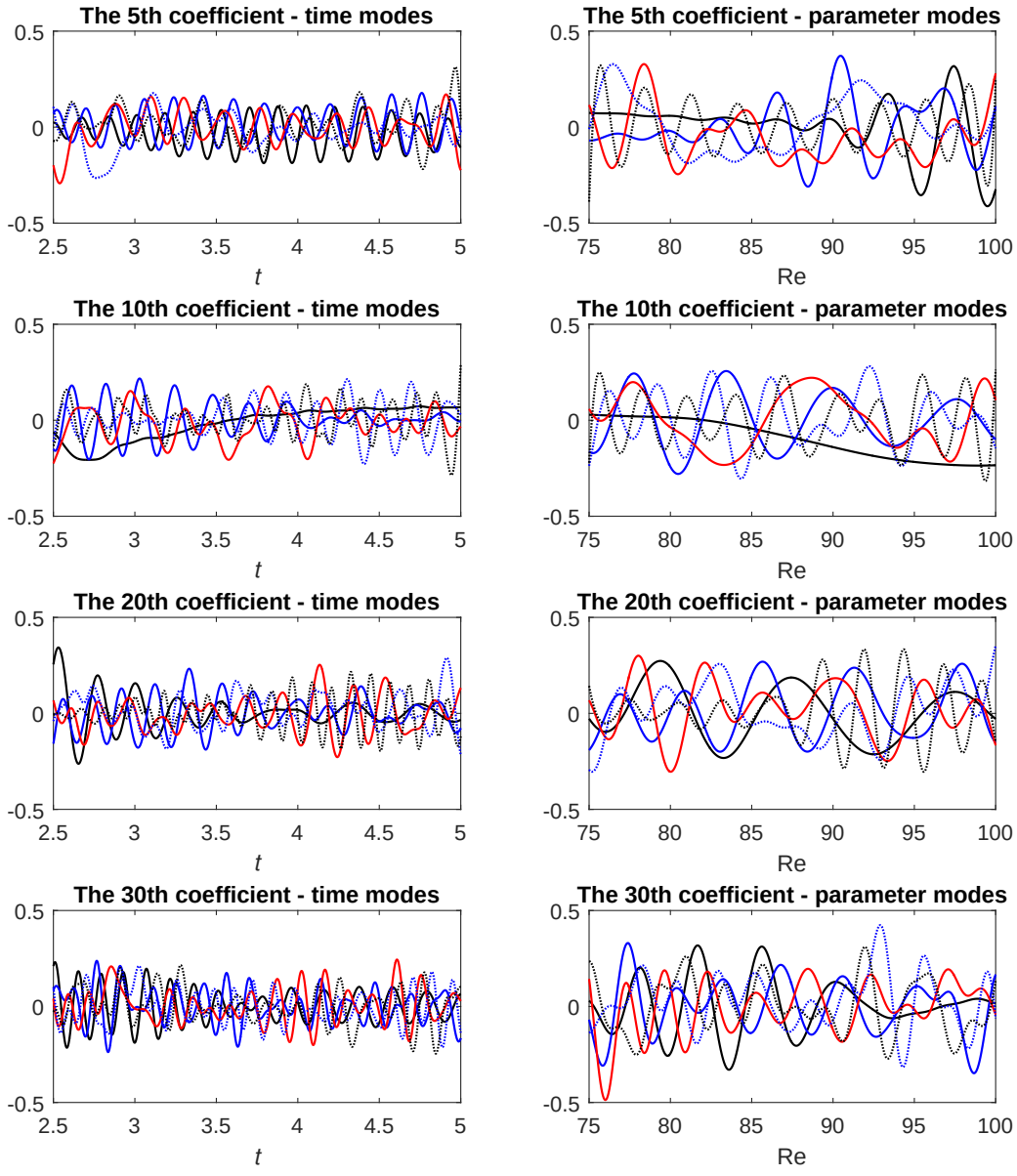


Figure 5: Time- and parameter-modes for the 5th, 10th, 20th and 30th projection coefficients for the pressure field: the 1st modes – black, the 5th modes – blue, the 10th modes – red, the 15th modes – blue dashed, the 20th modes – black dashed

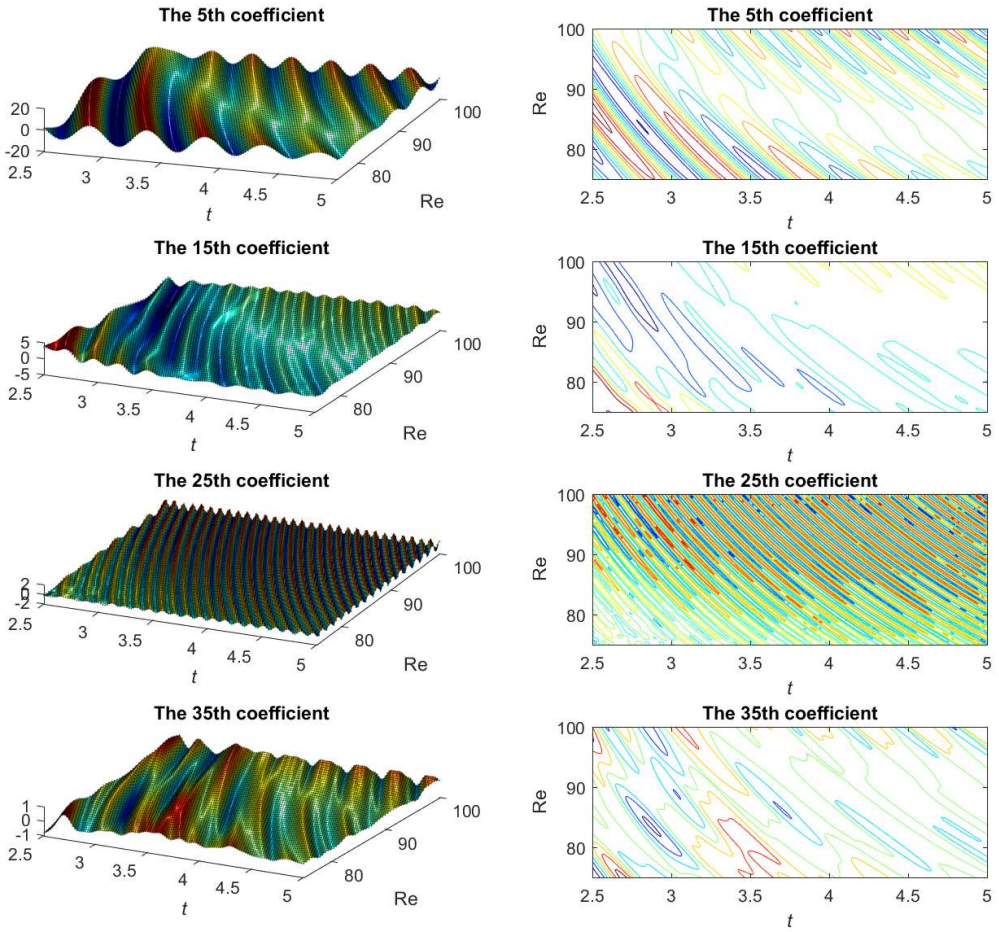


Figure 6: Regression results for the 5th, 15th, 25th and 35th projection coefficients for the velocity field

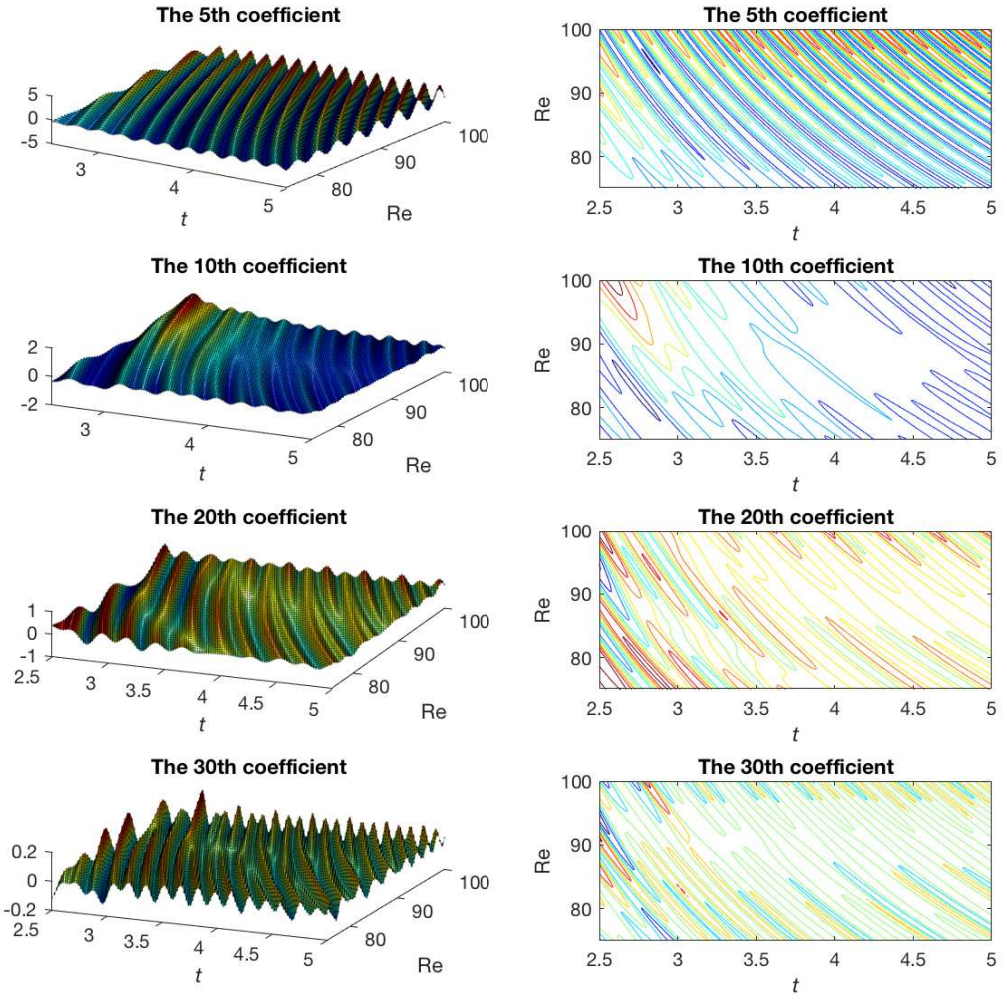
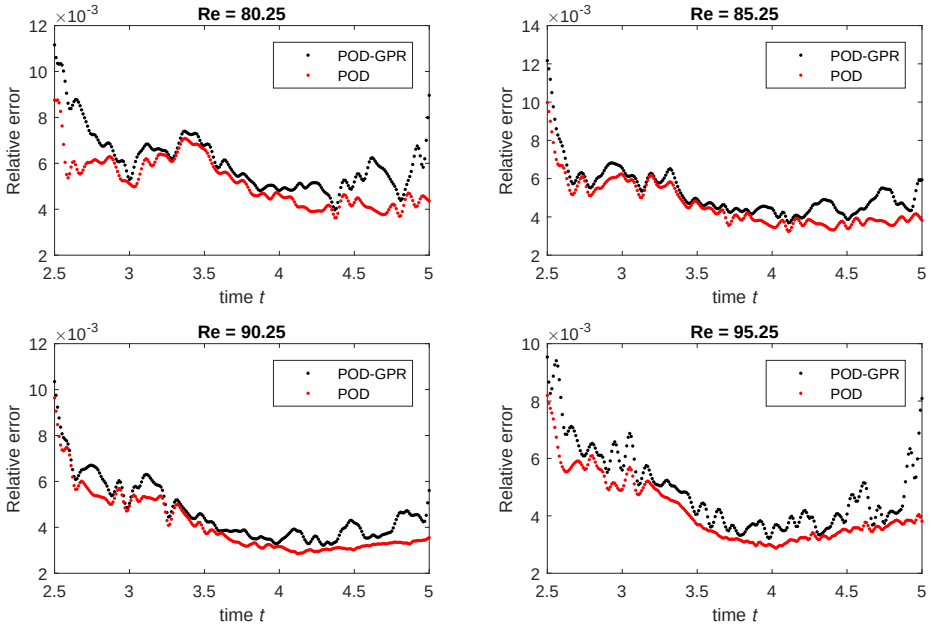
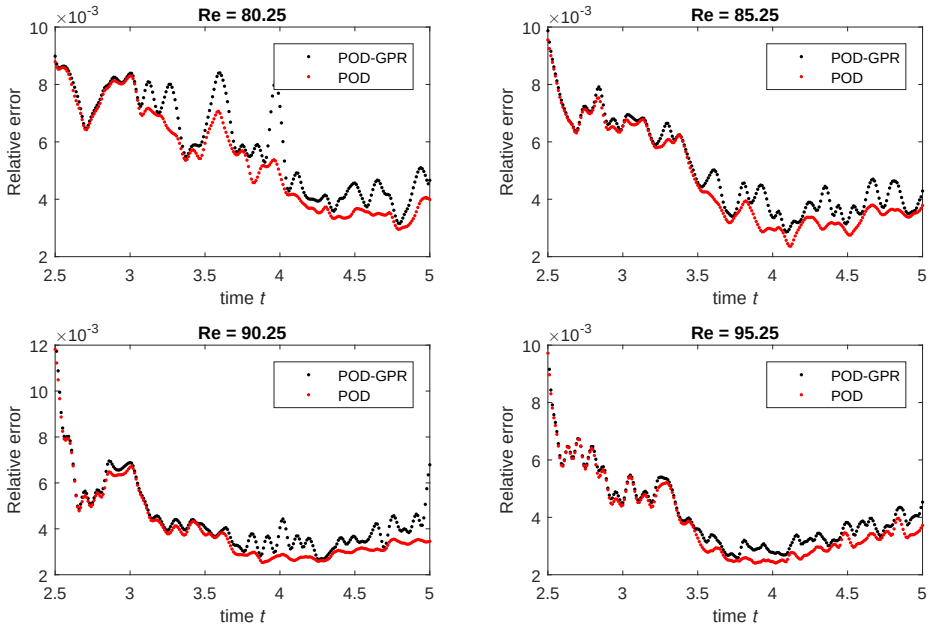


Figure 7: Regression results for the 5th, 10th, 20th and 30th projection coefficients for the pressure field



(a)



(b)

Figure 8: Relative errors of the solutions at the time steps with different Re values: (a) the velocity field; (b) the pressure field.

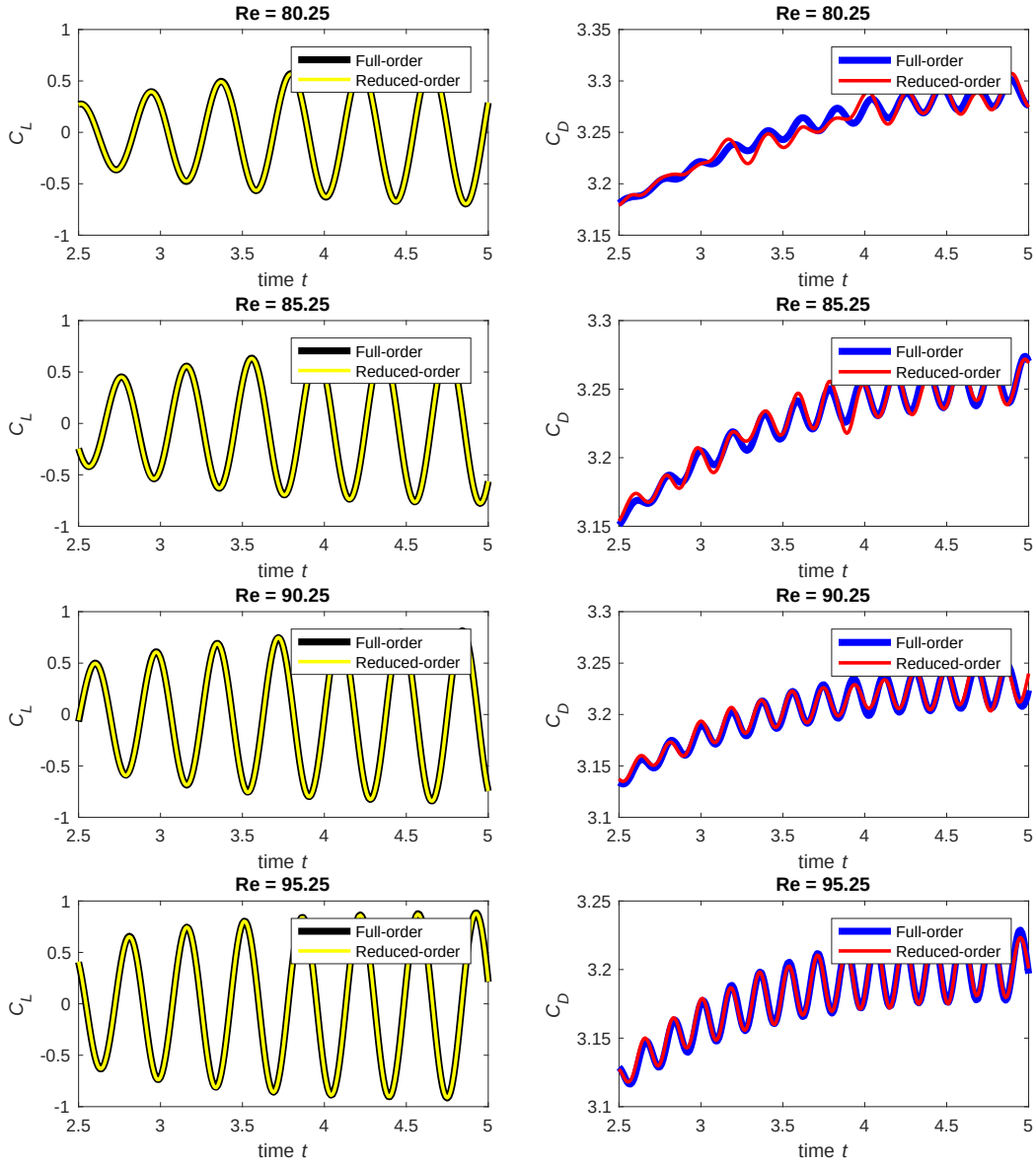
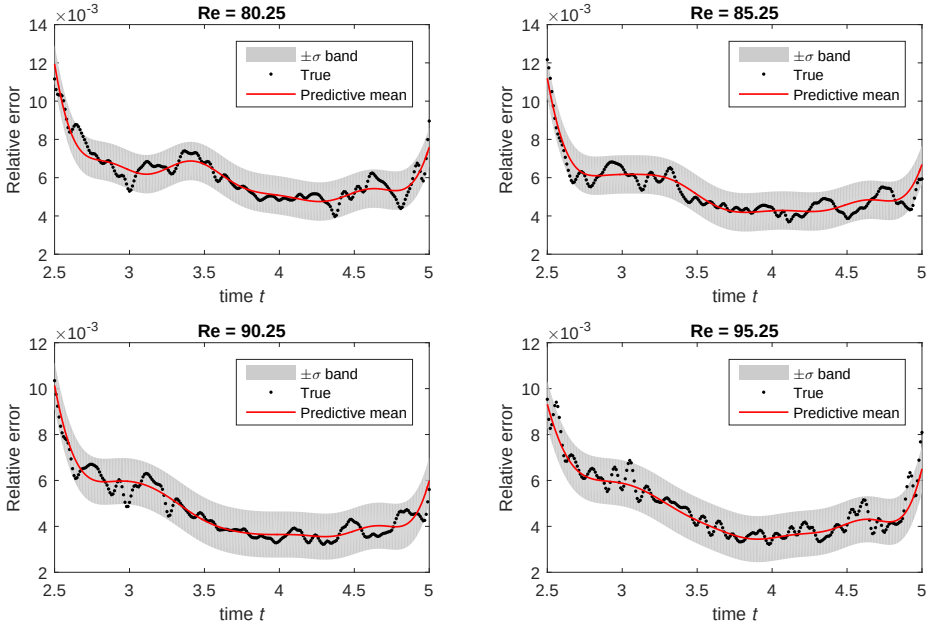
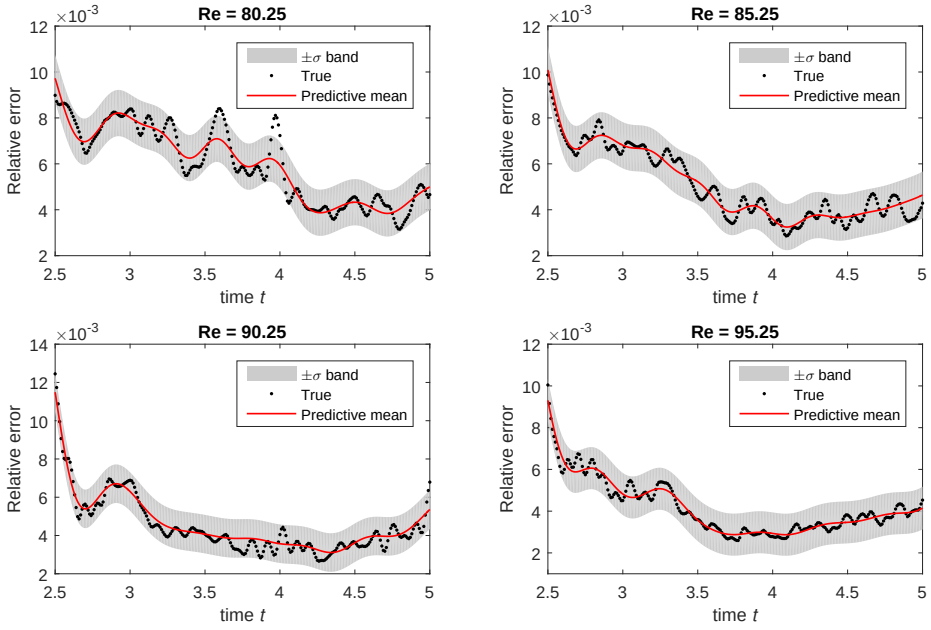


Figure 9: Comparisons of the drag and lift coefficients extracted from the full-order and reduced-order solutions under different Re values

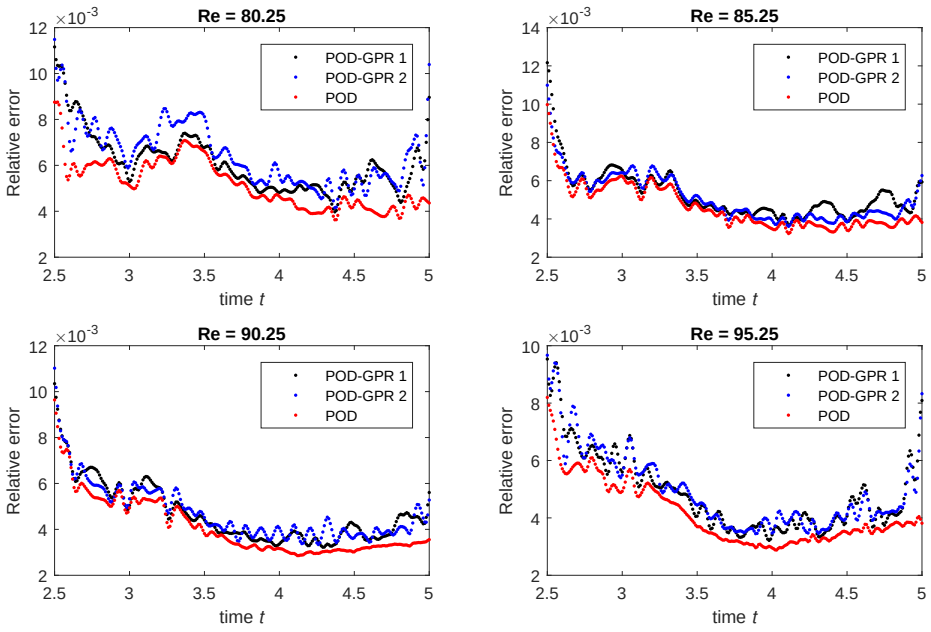


(a)

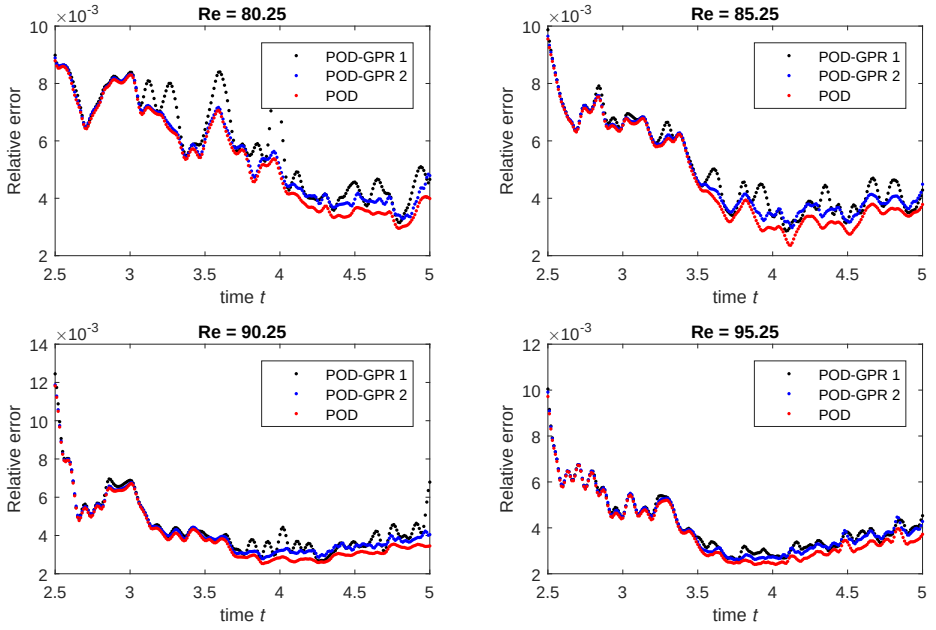


(b)

Figure 10: GP Surrogate of the relative errors of the solutions with different Re values: (a) the velocity field; (b) the pressure field.



(a)



(b)

Figure 11: Relative errors of the solutions at the time steps with different Re values: (a) the velocity field; (b) the pressure field.

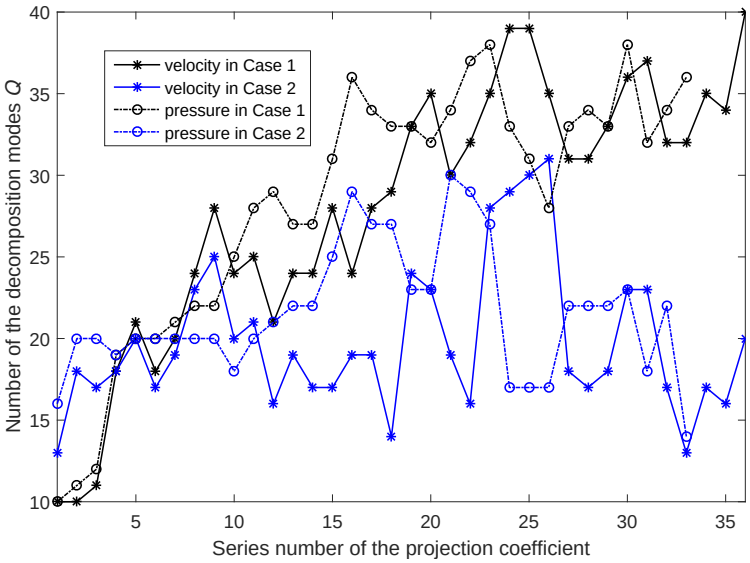


Figure 12: Comparison of the number of the decomposition modes Q for the projection coefficients

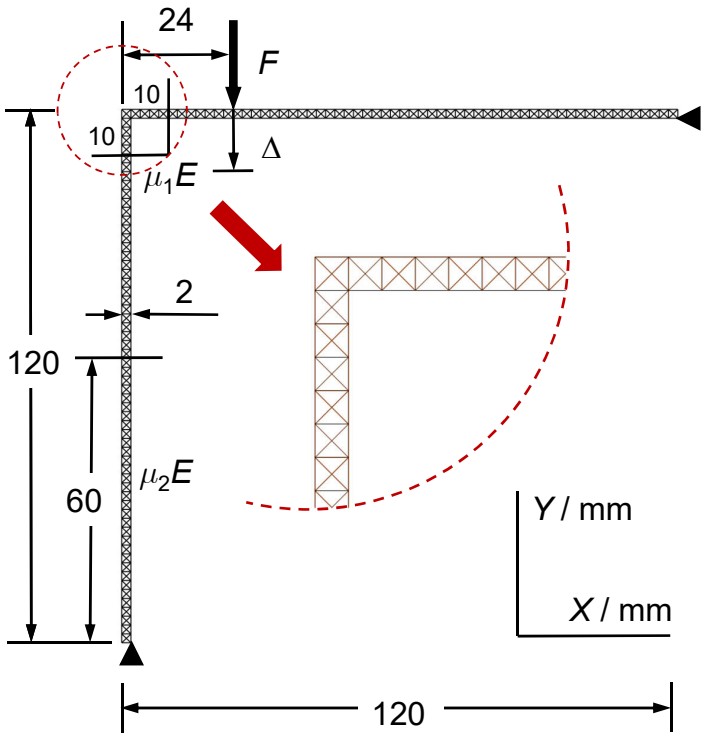


Figure 13: A trussed frame

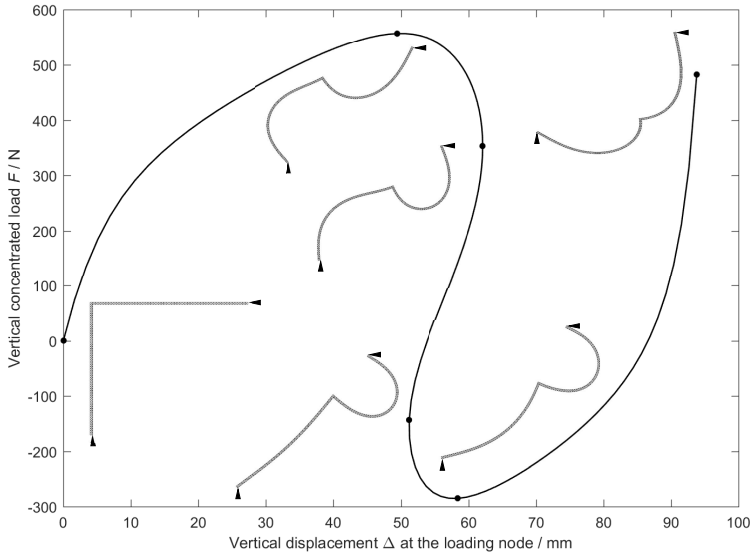


Figure 14: Configurations at different loading stages

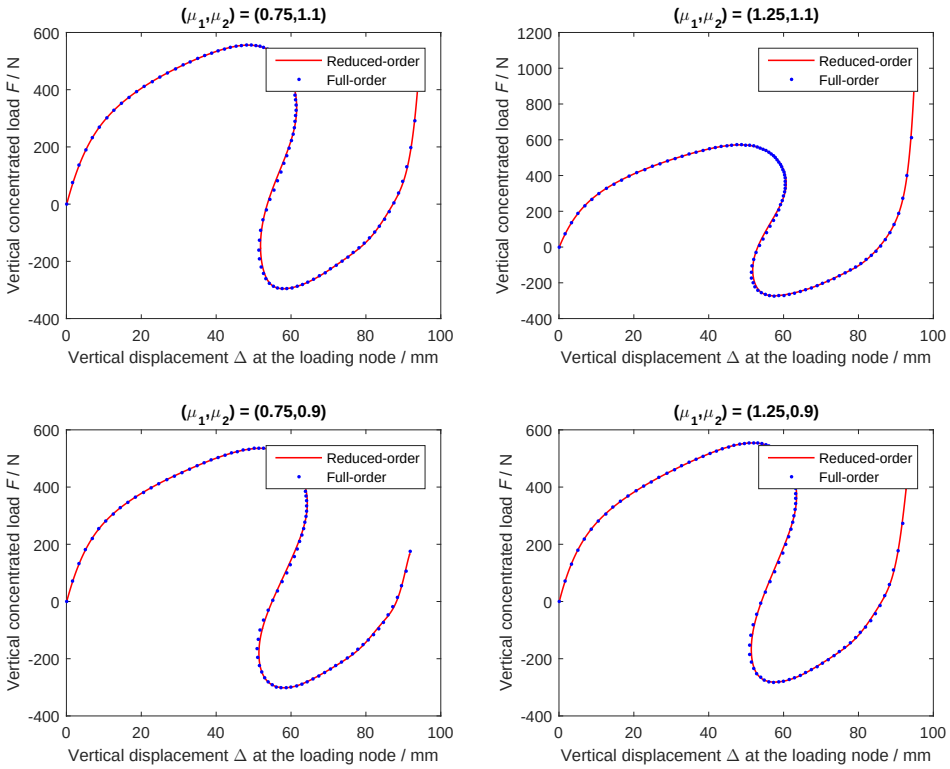


Figure 15: Comparison between the equilibrium paths extracted from the reduced-order and the full-order solutions