

# 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 for parametrized time-dependent problems is proposed. This method requires the offline preparation of a database comprising the time history of the full-order solutions 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 time and the parameters in the database, a singular-value decomposition (SVD) is used to extract the principal components in the data of projection coefficients. 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 domain of interest can be recovered rapidly as 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 approximating 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* [18, 36], in which the parameters characterize material properties, source terms, underlying geometry, boundary or initial conditions, etc. 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 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 replace the original full-order model with a reduced-order model with a significantly reduced dimensionality and a controlled loss of accuracy, to ensure the reliable evaluation of the reduced model at a substantially reduced computational cost.

Featuring an *offline-online* framework, the *reduced basis* (RB) *method* [18, 32, 36, 38] 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* [32, 38, 43] and the *proper orthogonal decomposition* (POD) [18, 23, 36]. The former selects a set of snapshots as the basis, by utilizing an error estimator/indicator and an 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 has been proposed as the POD-greedy approach [16, 18]. Here the POD is used for the compression in 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 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 a 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) [6] and its discrete variants [12, 28], also referred to as hyper-reductions [28], 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 and 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 [8].

There is substantial past work on ROM. For time-dependent problems, there are many successful cases of ROM, e.g. the RB method for linear evolution equations [16], the POD-based reduced-order fluid/structure modeling [24], the nonlinear ROM based on local reduced-order bases [3], the dynamic mode decomposition [21], several developments of the RB method for complex models [22, 34, 40], recently proposed structure-preserving RB approaches for Hamiltonian problems [1, 2], and non-intrusive frameworks based on radial basis function interpolation [4, 46]. One can refer to [8] for a comprehensive survey. Furthermore, some parametrized reduced order surrogates for dynamical systems have been developed in the domain of system identification, e.g. [5, 20, 25, 39, 45].

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

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

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.  $u : \Omega \times \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^n$  denotes the parameterized time-dependent solution field,  $n$  is the dimension of the system, and  $f : \Omega \times \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^n$  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 considered in this work. A Gaussian-type regression was combined with the RB method in [29, 30] 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, combined with artificial neural networks [17] in [19] and a Gaussian process regression (GPR) [37, 44] in [15]. 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* [9, 26]. The evaluations for new parameter values only require direct outputs from the regression models and linear combinations of the RB, yielding 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 further developed for time-dependent problems. The time coordinate  $t$  is considered as another parameter in the system. Over the time and parameter domain, a set of full-order solutions are prepared offline, including 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 provides 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 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 by some discretization scheme, one has the following system of parameterized ODEs:

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

where  $\mathbf{u}_h : \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^{N_h}$  is the discrete solution,  $N_h$  is the number of degrees of freedom (DOFs),  $\mathbf{L}_h$ ,  $\mathbf{N}_h$  and  $\mathbf{f}_h : \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^{N_h}$  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 substantial number of time steps and iterations, confirming that a full-order model is not affordable in a 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 by 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 selects the snapshot that maximizes the estimator/indicator until an error-criterion is satisfied. For time-dependent problems, a combination of the POD over time and the Greedy procedure in parameter domain can be used. However, suitable error estimators or indicators are not available for a general nonlinear time-dependent problem over the parameter domain, so in this work the proper orthogonal decomposition (POD) is used to extract the RB, as detailed in the following.

### 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

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

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 generated in the parameter domain. We assume that the database includes all 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

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

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

$$\mathcal{M}_\Theta = \text{span}\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{N_s}\} \subset \mathbb{R}^{N_h}.\tag{5}$$

If the point-set  $\Theta$  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 sought for  $\mathcal{M}_\Theta$ . Towards this end, the POD is employed to extract a set of orthogonal bases  $\mathbf{V} = [\mathbf{v}_1 | \mathbf{v}_2 | \dots | \mathbf{v}_L] \in \mathbb{R}^{N_h \times L}$  from the snapshots, so that the column space of  $\mathbf{V}$ , denoted by  $\text{Col}(\mathbf{V})$ , served as a reduced-order replacement for the full-order space  $\mathbb{R}^{N_h}$ . We assume that the reduced-order solution for time-parameter location  $(t, \boldsymbol{\mu}) \in \mathcal{T} \times \mathcal{P}$ , denoted as  $\mathbf{u}_{\text{rb}}(t; \boldsymbol{\mu})$ , is represented as a linear combination of the bases  $\mathbf{V}$ , i.e.  $\mathbf{u}_{\text{rb}}(t; \boldsymbol{\mu}) = \sum_{l=1}^L q_l(t; \boldsymbol{\mu}) \mathbf{v}_l = \mathbf{V} \mathbf{q}(t; \boldsymbol{\mu})$ , where  $\mathbf{q} = \{q_1, q_2, \dots, q_L\}^T \in \mathbb{R}^L$  collects the combination coefficients.

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

$$\mathbf{S} = [\mathbf{s}_1 | \mathbf{s}_2 | \dots | \mathbf{s}_{N_s}].\tag{6}$$

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

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

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.  $\mathbf{U}^T \mathbf{U} = \mathbf{I}_{N_h}$  and  $\mathbf{Z}^T \mathbf{Z} = \mathbf{I}_{N_s}$ , and  $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_{N_s}\}$  contains the singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{N_s} \geq 0$ .

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  $\mathbf{V} \in \mathbb{R}^{N_h \times L}$  be taken as the first  $L$  columns of  $\mathbf{U}$ , and let  $\mathbb{Y}_L = \{\mathbf{W} \in \mathbb{R}^{N_h \times L} : \mathbf{W}^T \mathbf{W} = \mathbf{I}_L\}$  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  $\sum_{i=1}^{N_s} \|\mathbf{s}_i - \mathbf{W} \mathbf{W}^T \mathbf{s}_i\|_{\mathbb{R}^{N_h}}^2$ .

The Schmidt-Eckart-Young theorem [14, 36, 41] 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  $N_s$ th singular values, i.e.

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

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

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

Thus  $\text{Col}(\mathbf{S})$  can be well approximated by  $\text{Col}(\mathbf{V})$  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 reduce the computational cost for a general nonlinear problem.

To overcome these difficulties, a regression-based approach is proposed 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  $\text{Col}(\mathbf{V})$  acts as the corresponding reduced-order solution at the algebraic level,

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

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

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

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

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

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

with  $1 \leq n_1 \leq n_2 \leq \dots \leq n_{N_t^{tr}} \leq N_t^d$  and  $1 \leq m_1 \leq m_2 \leq \dots \leq m_{N_\mu^{tr}} \leq N_\mu^d$ . 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  $\mathbf{u}_{\text{rb,reg}}(t^*, \boldsymbol{\mu}^*) \in \text{Col}(\mathbf{V})$  is

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

Once the regression model is trained, 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

---

1: **Offline stage:**

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

7: **Online stage:**

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

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.

**Remark 1:** The dynamic mode decomposition (DMD) [21] has been proposed for the model reduction of time-dependent problems, and is also a data-driven framework. Based on the database of snapshots over a time domain, the DMD is defined as an approximate eigendecomposition of the operator that relates one time step to the next, and is the best-fit of the operator in some sense. In this approach, the reduced basis, i.e. the set of truncated eigenvectors, is extracted based on the correlation between each pair of adjacent snapshots over the database, and is essentially a re-combination of the POD basis. With a nature of a low-rank approximation, the DMD explicitly gives the evolving solution in a reduced space spanned by the eigenvectors. The DMD enjoys several advantages, i.e. it is equation-free and gives explicit solutions, without solving the reduced model in the online stage, and a future-state prediction is possible provided the DMD approximation remains valid at the future time. However, it cannot be directly applied to parametrized time-dependent problems. As in such a situation, the operator relating adjacent time steps, or the dynamic modes, will be parameter-dependent.

**Remark 2:** In the context of uncertainty quantification, probability distributions are defined for the characterizing parameters. Sampling techniques with respect to the parameters often require a large amount of online evaluations. Instead of solving the high-fidelity models, the proposed regression-based reduced model can be used online as a cheap surrogate. Using different numbers of bases, a natural hierarchy of fidelities is generated, and multi-fidelity schemes can be used for the evaluations in the sampling procedure. One can refer to [33] for some multi-fidelity techniques and corrections for obtaining unbiased solutions.

### 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 observational data. Let  $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \dots, M\}$  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 [37, 44], the observed input-output pairs are assumed to follow some regression function  $f : \mathcal{P} \rightarrow \mathbb{R}$ , 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  $(\mathbf{x}, \mathbf{x}') \in \mathcal{P} \times \mathcal{P}$ ,

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

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

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

which considers an individual correlated lengthscale  $\ell_m$  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:

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

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

From a regression model, the goal is to predict the noise-free output  $f^*(\mathbf{s})$  for a new test input  $\mathbf{s} \in \mathcal{P}$ . Then one can combine the information from the 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:

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

The values of the hyperparameters  $\boldsymbol{\theta}$  make significant difference on the predictive performance, with  $\boldsymbol{\theta} = \{\beta_1, \dots, \beta_N, \ell_1, \dots, \ell_d, \sigma_f, \sigma_y\}$  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  $\boldsymbol{\theta}_{\text{opt}}$  can be estimated via the maximization problem:

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

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

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

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  $\beta$ . In the numerical implementations for time-dependent problems, the predictive mean function  $m^*(\cdot)$  is of primary interest.

### 3.2. Tensor-decomposition-based regression

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

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

as a result of the tensor grid between the time and the 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

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

Here  $\boldsymbol{\psi}_k^l$  and  $\boldsymbol{\phi}_k^l$  are the  $k$ th discrete time- and parameter-modes for the  $l$ th projection coefficient, respectively,  $\lambda_k^l$  is the  $k$ th singular value for the same coefficient, and  $Q_l$  is the corresponding truncation rank.

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

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

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

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

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

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

which shares a similar formulation with (20). The regression procedure is shown in Figure 1.

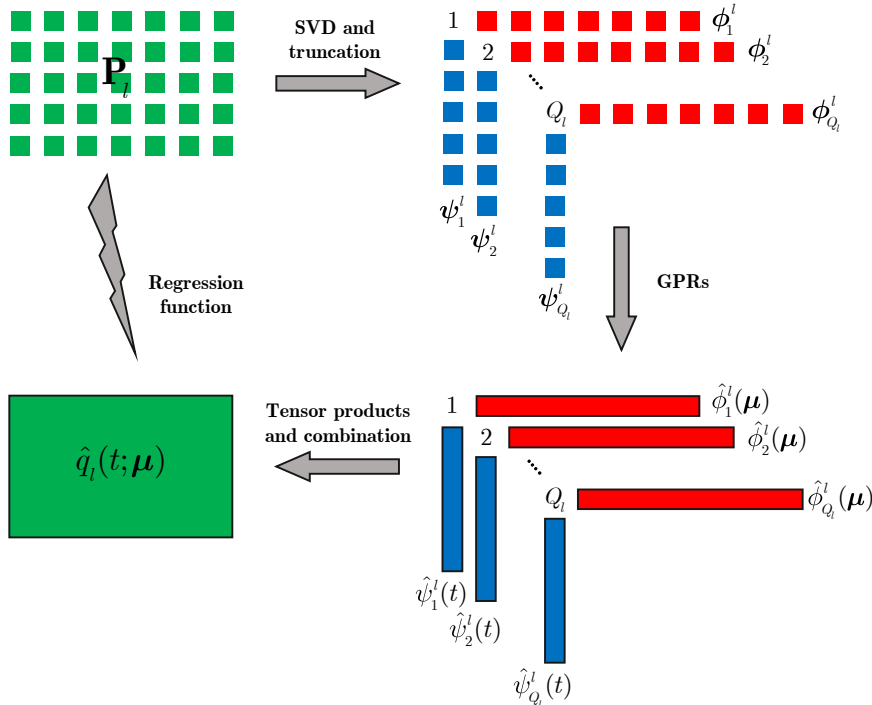


Figure 1: Tensor-decomposition-based regression

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, often resulting in a difficult (global) GPR. After the tensor decomposition, the 1D regressions with respect to time, combined with the multi-dimensional regressions with respect to the parameters, are both simpler. For the  $l$ th 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. More hyperparameters in the tensor-decomposition-based approach imply a more flexible scheme.

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.

**Remark 3:** In some cases of strong nonlinearities, the corresponding response may lie in some low-probability areas, and it may no longer be as smooth as in the regular cases. In such circumstance, there can be non-smoothness, large gradients, or even discontinuities in the functions to approximate, and the conventional GPR may be unable to yield a good representation. In such cases, we can resort to the non-stationary Gaussian process regression [31, 35]. As an inherent drawback of the numerical schemes based on supervised learning, the proposed reduced model cannot ensure good predictive results at time/parameter locations outside the coverage of the database. If it is required to predict future states after the training time domain  $\mathcal{T} = [0, T]$ , one can use some regression models for time series, such as autoregression [5], to deal with the data of time-modes and recover solutions. At future times, however, the predictive accuracy must be carefully evaluated.

### 3.3. Error control for the truncations

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

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

where  $\tilde{\mathbf{P}}_l$  represents the coefficients following the SVD truncation, as defined in (20). Here  $\delta_l^2$  represents the truncation error of the SVD for the  $l$ th projection coefficient, i.e.  $\delta_l^2 = \|\mathbf{P}_l - \tilde{\mathbf{P}}_l\|_F^2 / \|\mathbf{P}_l\|_F^2$ ,  $\eta_l^2$  is defined as  $\sigma_l^2 / \sum_{i=1}^{N_s} \sigma_i^2$ ,  $\sigma_l$  is the  $l$ th singular value in the POD, used to construct the RB, and  $\varepsilon_L^2 = 1 - \sum_{l=1}^L \eta_l^2$  yields the truncation error of the POD. In the data-driven context, (23) can be used to evaluate the truncation errors both due to the POD, used 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 [13], 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. In [42], 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 use 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,  $i = 1, 2, \dots, N_t^{tr}$ ,  $j = 1, 2, \dots, N_\mu^{tr}$ , the relative error of the reduced-order solution is given as

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

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

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

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 a 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 for 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 the 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 defining 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 field 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 a parameterized diffusion coefficient is considered,

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

In the offline stage, full-order solutions are calculated by a finite difference method with  $N_h = 201$  nodes at  $N_\mu^d = 27$  uniformly distributed parameter locations of  $\mu$ , i.e.  $\mathcal{P}_d = \{1.00, 1.25, 1.50, \dots, 7.25, 7.5\}$ . Among the 1000 time steps,  $N_t^d = 250$  uniformly distributed ones are included in the database, i.e.  $\mathcal{T}_d =$

$\{0.004, 0.008, 0.012, \dots, 0.996, 1.000\}$ . All the full-order solution-vectors are used both as snapshots for constructing the RB and training data for the GPR models. Through the POD, a set of  $L = 7$  orthogonal modes are extracted as the RB with a tolerance  $\varepsilon_{L,\text{tol}} = 0.1\%$ , shown in Figure 2. As proposed previously, regression models are then constructed as the combination of time- and parameter-modes, which are trained as GPs after the SVD truncations with tolerances  $\delta_{l,\text{tol}} = 0.1\%$ ,  $1 \leq l \leq L = 7$ .

To validate the accuracy of the reduced model, the reduced-order solutions are recovered at 5 time points for 4 different values of  $\mu$ , and are compared with the corresponding full-order solutions in Figure 3. It is clear that the full-order and reduced-order solutions are matching well with each other, even when the viscous parameter  $\mu$  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 4, 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:

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

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 at  $\partial B_r(0.2, 0.2)$ , and Neumann conditions are set along the right (outflow) edge. Moreover, the following parabolic velocity profile is prescribed on the left (inflow) edge:

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

in which the value of parameter  $U$  is subject to change in an interval  $[1.125, 1.5]$ . Correspondingly, the Reynolds number  $Re$ , defined as  $Re = U_m L / \nu$ , lies in an interval  $\mathcal{P} = [75, 100]$ . Here 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  $\mathbf{u}$  and the pressure solution  $p$ . In this example, we use the finite element [7] solver in the MATLAB library `redbKIT v2.2` [27, 36] to compute the full-order solutions. At 51 different values  $\mathcal{P}_d = \{75.0, 75.5, 76.0, \dots, 99.5, 100.0\}$  of the parameter  $Re$ , we use a small time step to obtain a high-fidelity time series of the solutions for the velocity and the pressure, and add the full-order solutions at 126 time points  $t \in \mathcal{T}_d = \{2.50, 2.52, 2.54, \dots, 4.98, 5.00\}$  into the database  $\mathcal{D}$ . These samples will be used as snapshots and training data.

After the POD with a truncation tolerance  $\varepsilon_{L,\text{tol}} = 0.5\%$ ,  $L = 36$  reduced bases are extracted for the velocity and  $L = 33$  for the pressure. 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  $\delta_{l,\text{tol}} = \varepsilon_{L,\text{tol}} = 0.5\%$  for  $l = 2, 3, \dots, L$ . A stricter criterion is defined for the 1st coefficient due to its dominant role in the accuracy of reduced-order solutions.

After the tensor decompositions of the data of all projection coefficients, the time- and parameter-modes are obtained as GPR models, some of which are shown in Figures 5 and 6. The mean regression functions of all the coefficients are recovered in a similar decomposition form and some of these regression results are shown in Figures 7 and 8. It is worth pointing out that the regression can be finished within minutes, while one run of the full-order time series takes around 15 minutes on the same computational platform for the time domain  $t \in [0.0, 5.0]$ .

After the offline training, online tests are performed on the solutions for four new  $Re$  values, not included in the offline database and chosen as  $Re \in \{80.25, 85.25, 90.25, 95.25\}$ . The reduced-order time-dependent solutions for these four 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 the 2-norm, are calculated and plotted in Figure 9. As can be observed, the relative error are 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 four test cases. As shown in Figure 10, they are matching 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 11. It can be seen that the true errors between the reduced-order and the 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 but practical error indicators for the reduced-order solutions.

**Case 2:**

Following (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

$$\bar{e}_{\text{rcv},\text{to1}}^2(\mathcal{F}) = \varepsilon_{L,\text{to1}}^2 + \sum_{l=1}^L \eta_l^2 \delta_{l,\text{to1}}^2. \quad (26)$$

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

We set  $\delta_{l,\text{to1}} = \alpha \varepsilon_{L,\text{to1}} / (\eta_l \sqrt{L})$ , so that  $\eta_l^2 \delta_{l,\text{to1}}^2 = (\alpha \varepsilon_{L,\text{to1}})^2$  for each  $1 \leq l \leq L$ , i.e., each term of the summation in (26) contributes equally. To ensure similar values of  $\bar{e}_{\text{rcv},\text{to1}}^2$  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  $\bar{e}_{\text{rcv},\text{to1}}^2 = 1.040 \varepsilon_{L,\text{to1}}^2$  for the velocity and  $\bar{e}_{\text{rcv},\text{to1}}^2 = 1.073 \varepsilon_{L,\text{to1}}^2$  for the pressure, accordingly.

Following the same procedure of offline training and online tests as in Case 1, the reduced-order solutions for the four test values of  $Re$  are obtained, and their relative error are shown in Figure 12, compared with those of Case 1. In both cases the solutions have a similar accuracy, as a results of sharing similar  $\bar{e}_{\text{rcv},\text{to1}}^2$ 's. On the other side, the rank of the SVD-truncations,  $Q_l$ , for both velocity and pressure in the two cases are shown in Figure 13. 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. Setting these 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 14, 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 [10, 11] is used as a high-fidelity solver for this problem. The quantities in Figure 14 are given as: Young's modulus  $E = 210$  GPa, unit force  $F_0 = 1$  N and unit displacement  $\Delta_0 = 1$  mm.

**Case 1: hyperelasticity**

With a hyperelastic constitutive law and a uniform Young's modulus  $E$  in the whole structure, the equilibrium paths, referred to as the load-displacement curve, is shown in Figure 15, together with some configurations of the frame at different loading stages.

In this case, there are two parameters  $\mu_1$  and  $\mu_2$ :  $\mu_1 \in [0.5, 1.5]$  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 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  $t_i = (i - 1)/(N_t^d - 1)$ ,  $i = 1, 2, \dots, N_t^d$ . All full-order samples will be used 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  $\varepsilon_{L, \text{to1}} = 0.1\%$  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 four new parameter locations that are not included in the database. We 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 four test locations of  $(\mu_1, \mu_2)$  are shown in Figure 16, matching very well with those obtained by full-order simulations. Computational times in different stages are compared in Table 1, showing the significantly enhanced efficiency of the regression-based reduced model.

	Time / s
One run of offline full-order time series in average	2.28E+01
Construction of all regression models	4.88E+00
One run of online reduced-order time series in average	2.23E-02

Table 1: Comparison of computational times

### Case 2: hyperelasto-plasticity

We consider a constitutive law of hyperelasto-plasticity in this case. One characterizing parameter  $\mu_3 = \tau_y$ , the value of yielding stress measured by the unit N/mm<sup>2</sup>, is taken into account, and the parameter domain is  $\mathcal{P} = [5000, 6000]$ . The elasto-plastic deformation along the equilibrium path is again considered as a pseudo-time-dependent problem. The full-order database and the pseudo-time are constructed in the same way as Case 1. The time series at each of the  $N_\mu^d = 20$  uniformly distributed parameter locations are taken as the 100 loading increments with fixed arc-length, i.e.  $N_t^d = 101$ . Similarly to Case 1, we take  $\varepsilon_{L, \text{to1}} = 0.1\%$  and  $\alpha = 1.0$ , and train regression-based reduced models for the equilibrium path.

The equilibrium path of the hyperelasto-plastic structure is not as complex as that of the hyperelastic case and only has two monotone stages – a hardening stage and a softening stage. Due to much stronger nonlinearity in the constitutive law, however, the large-deformation simulation of the structure is more challenging than that in the previous case. Online tests are again performed at four new parameter locations that are not included in the database. As shown in Figure 17, reduced- and full-order solutions of the equilibrium paths are matching well for these test parameter values. It can be seen that the reduced model gives good results for both the hardening and softening stages, and accurately detects both the location and the value of the ultimate load.

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.

Furthermore, we would like to refer to [19] for some quantitative comparisons of the online accuracy and efficiency between the conventional RB method and the regression-based RB scheme, and [15] for a comparison of computational times at different stages in the regression-based framework.

## 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 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 the 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.

## Acknowledgment

This work was supported by the Commission for Technology and Innovation (CTI) under Grant no. 25964.2 PFIW-IW.

## 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] C. Audouze, F. De Vuyst, and P. B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.
- [5] L. D. Avendaño-Valencia, E. N. Chatzi, K. Y. Koo, and J. M. Brownjohn. Gaussian process time-series models for structures under operational variability. *Frontiers in Built Environment*, 3:69, 2017.
- [6] 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 Mathématique*, 339(9):667–672, 2004.
- [7] K.-J. Bathe. *Finite Element Procedures*. Klaus-Jurgen Bathe, 2006.
- [8] 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.
- [9] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [10] J. Bonet, A. J. Gil, and R. D. Wood. FLagSHyP software. <http://www.flagshyp.com/>, 2016.
- [11] J. Bonet, A. J. Gil, and R. D. Wood. *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. Cambridge University Press, 2016.
- [12] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [13] 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.
- [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 J. S. Hesthaven. Reduced order modeling for nonlinear structural analysis using gaussian process regression. *Computer Methods in Applied Mechanics and Engineering*, 341:807–826, 2018.
- [16] 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.
- [17] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [18] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.
- [19] 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.
- [20] F. Kopsaftopoulos, R. Nardari, Y.-H. Li, and F.-K. Chang. A stochastic global identification framework for aerospace structures operating under varying flight states. *Mechanical Systems and Signal Processing*, 98:425–447, 2018.

- [21] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*, volume 149. SIAM, 2016.
- [22] 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.
- [23] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. Proper orthogonal decomposition and its applications, part i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.
- [24] 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.
- [25] C. V. Mai, M. D. Spiridonakos, E. N. Chatzi, and B. Sudret. Surrogate modeling for stochastic dynamical systems by combining nonlinear autoregressive with exogenous input models and polynomial chaos expansions. *International Journal for Uncertainty Quantification*, 6(4), 2016.
- [26] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.
- [27] F. Negri. redbKIT Version 2.2. <http://redbkit.github.io/redbKIT/>, 2016.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] C. J. Paciorek and M. J. Schervish. Nonstationary covariance functions for gaussian process regression. In *Advances in neural information processing systems*, pages 273–280, 2004.
- [32] 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.
- [33] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.
- [34] 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.
- [35] C. Plogemann, K. Kersting, and W. Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 204–219. Springer, 2008.
- [36] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.
- [37] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT press Cambridge, 2006.
- [38] 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.
- [39] J. Sakellariou and S. Fassois. Functionally pooled models for the global identification of stochastic systems under different pseudo-static operating conditions. *Mechanical Systems and Signal Processing*, 72:785–807, 2016.
- [40] 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.
- [41] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. i. teil: Entwicklung willkrlicher funktionen nach systemen vorgeschriebener. *Mathematische Annalen*, 63:433–476, 1907.
- [42] 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.
- [43] 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.
- [44] C. K. Williams and C. E. Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.
- [45] K. Worden, G. Manson, and E. J. Cross. On gaussian process narx models and their higher-order frequency response functions. In *Solving Computationally Expensive Engineering Problems*, pages 315–335. Springer, 2014.
- [46] D. Xiao, F. Fang, C. Pain, and I. Navon. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering*, 317:868–889, 2017.

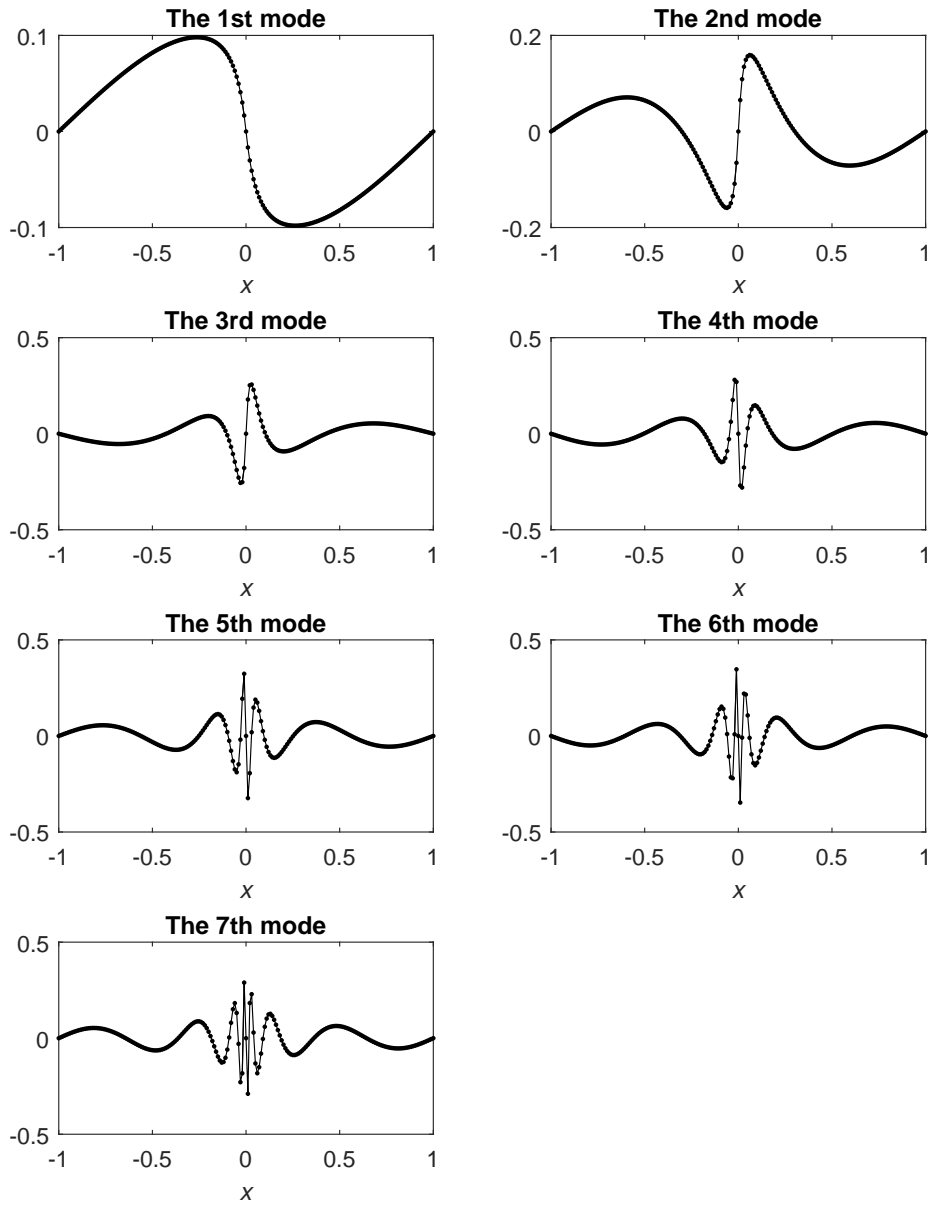


Figure 2: The RB modes for the parametrized viscous Burgers' equation



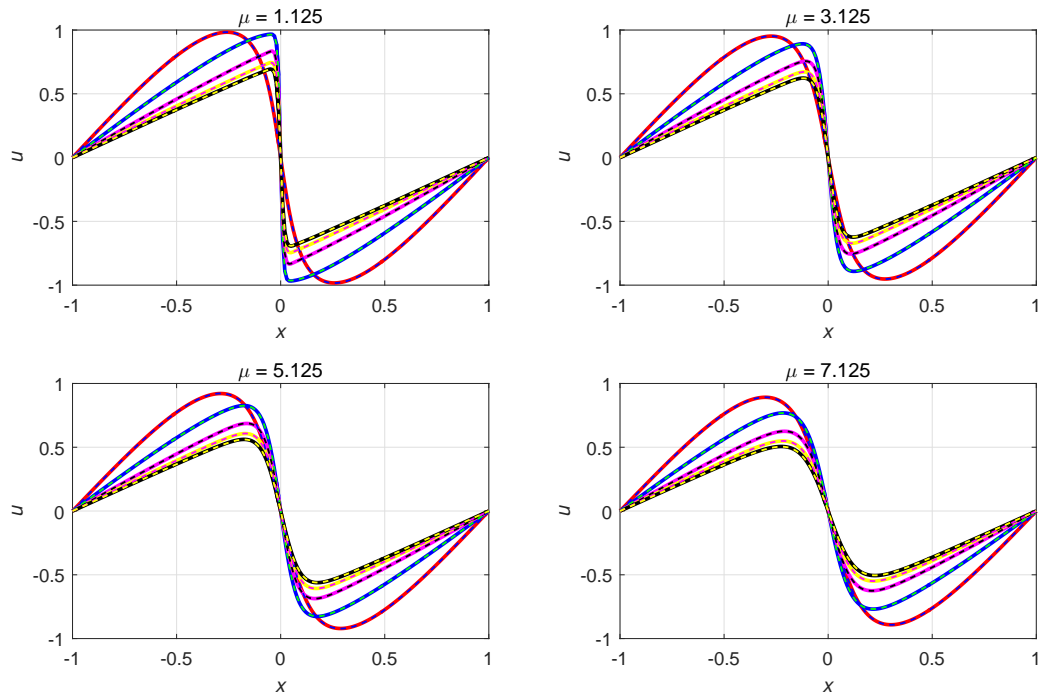


Figure 3: Comparison between the full-order solutions and the corresponding reduced-order solutions: solutions at different  $t \in \{0.25, 0.50, 0.75, 0.90, 1.00\}$  with different values of  $\mu$ : full-order solutions – dashed, reduced-order solution – solid.

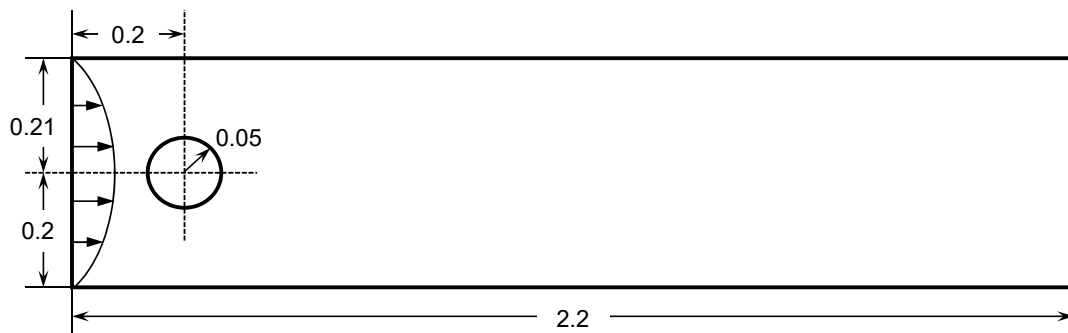


Figure 4: Underlying geometry for the problem of fluid flow around a cylinder

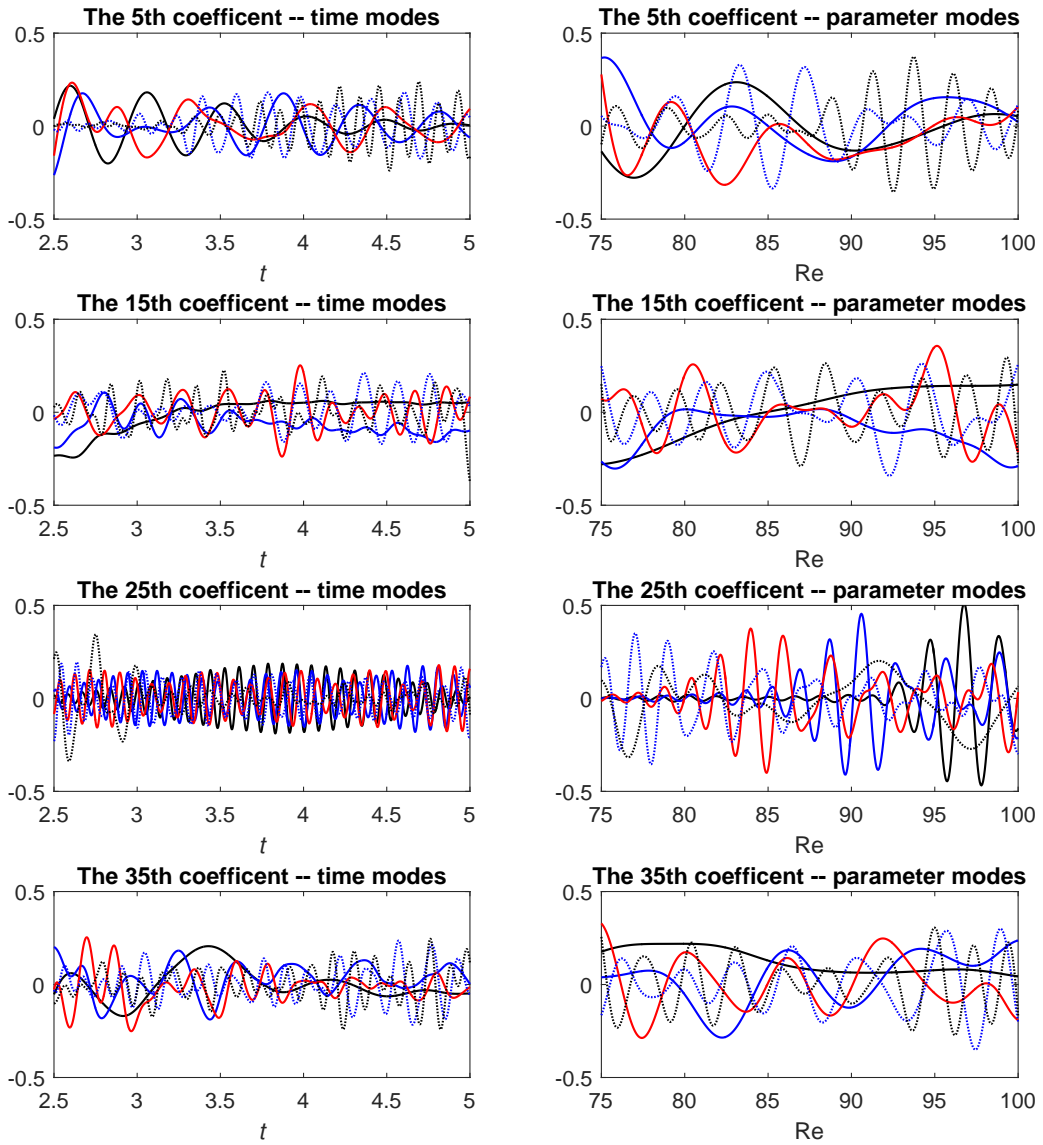


Figure 5: Time- and parameter-modes for the 5th, 15th, 25th and 35th projection coefficients for the velocity field: the 1st modes – black, the 5th modes – blue, the 10th modes – red, the 15th modes – blue dashed, the 20th modes – black dashed

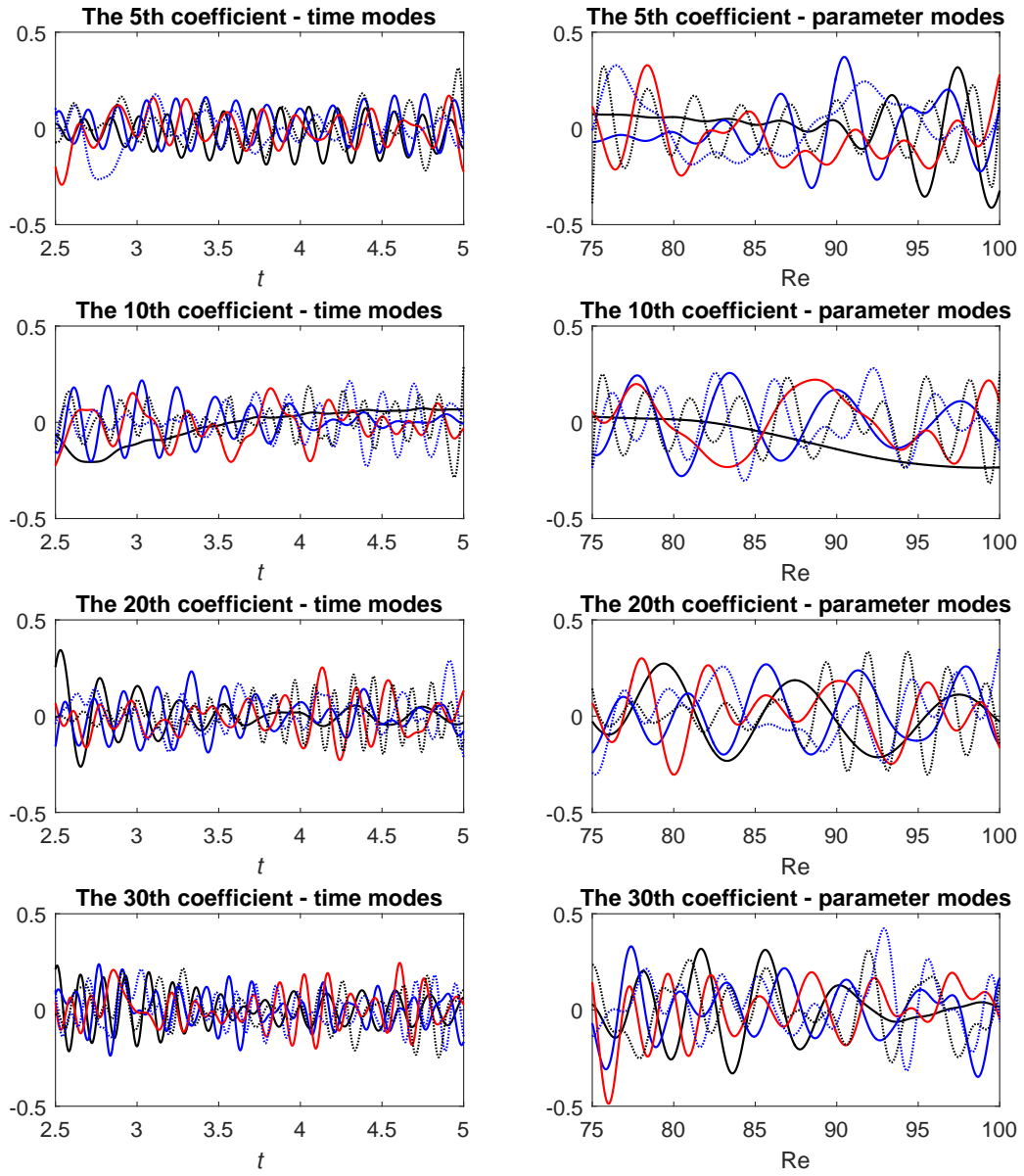


Figure 6: Time- and parameter-modes for the 5th, 10th, 20th and 30th projection coefficients for the pressure field: the 1st modes – black, the 5th modes – blue, the 10th modes – red, the 15th modes – blue dashed, the 20th modes – black dashed

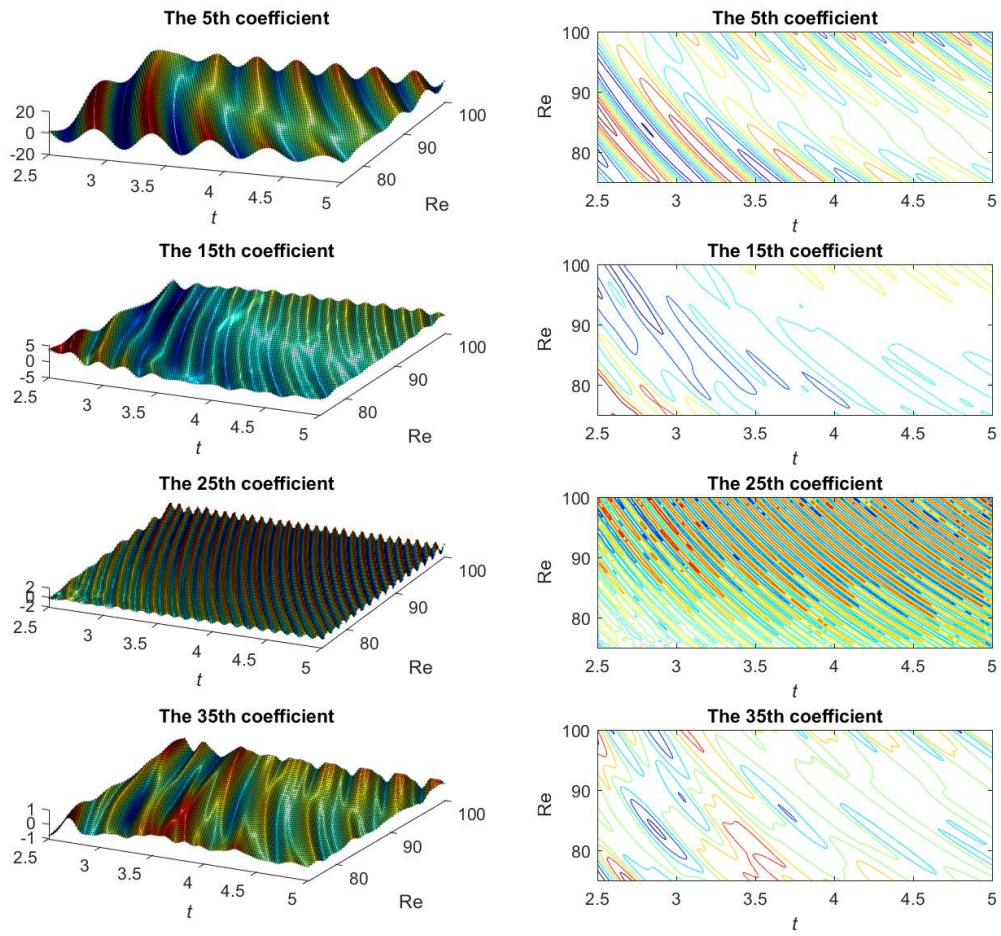


Figure 7: Regression results for the 5th, 15th, 25th and 35th projection coefficients for the velocity field

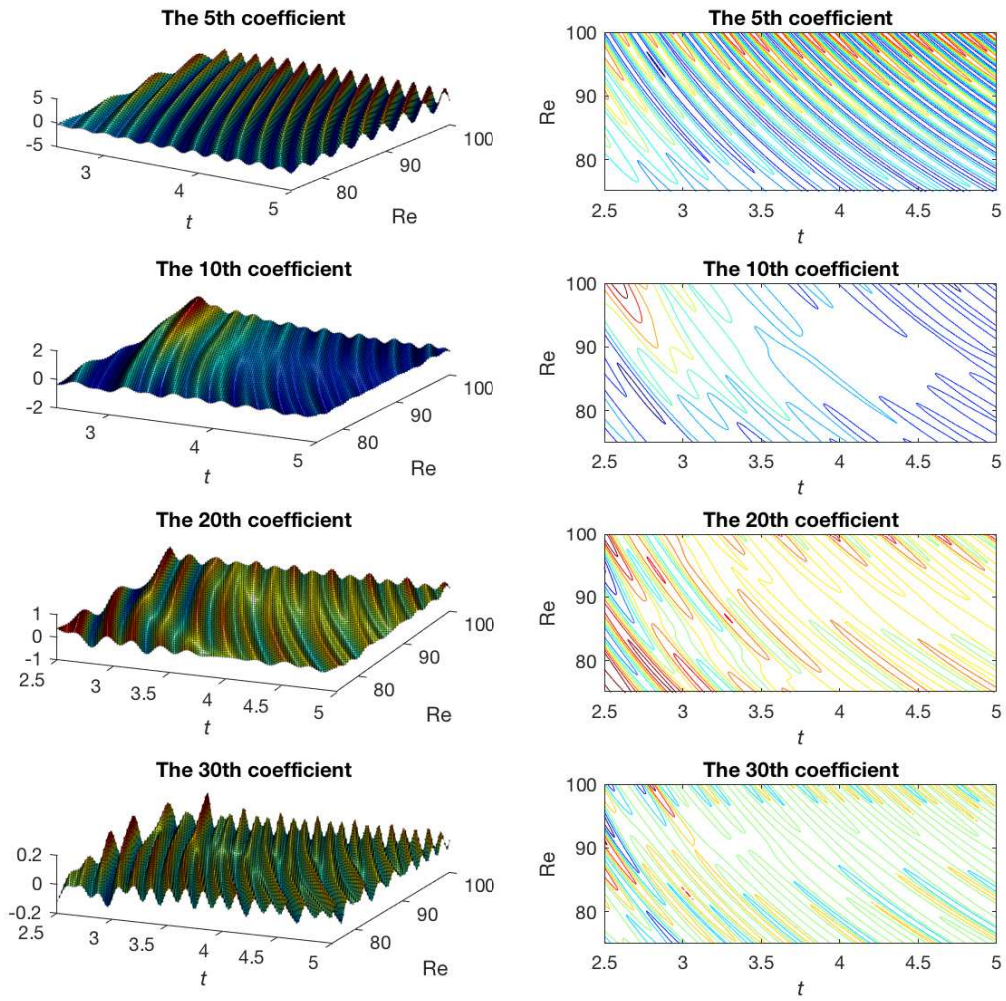
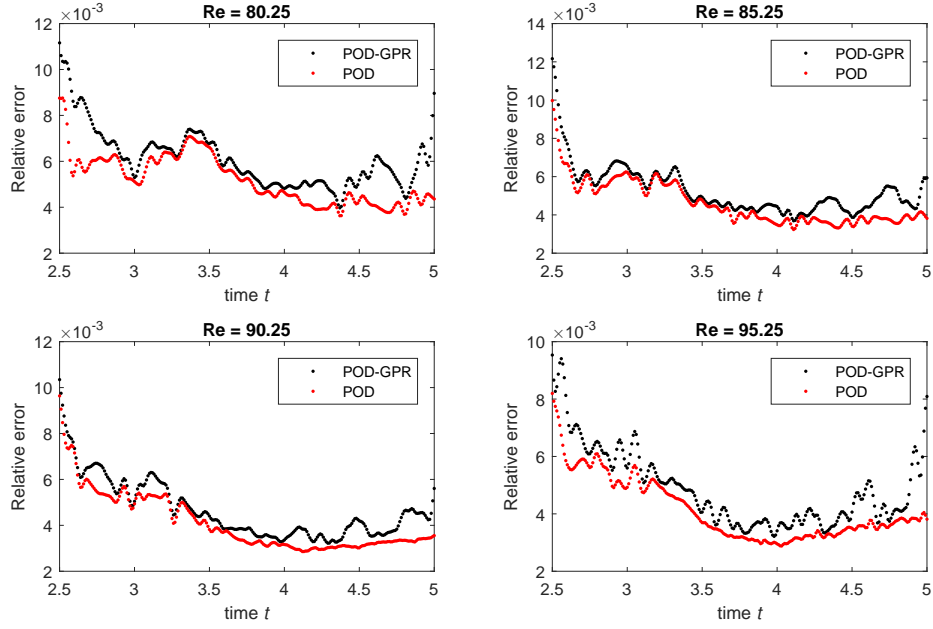
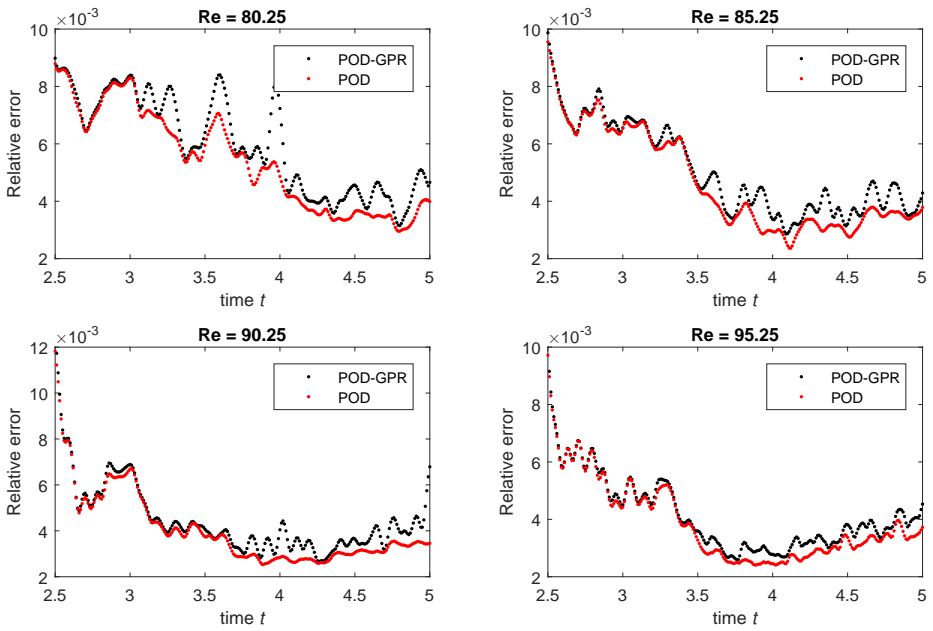


Figure 8: Regression results for the 5th, 10th, 20th and 30th projection coefficients for the pressure field



(a)



(b)

Figure 9: Relative errors of the solutions at the time steps with different  $Re$  values: (a) the velocity field; (b) the pressure field.

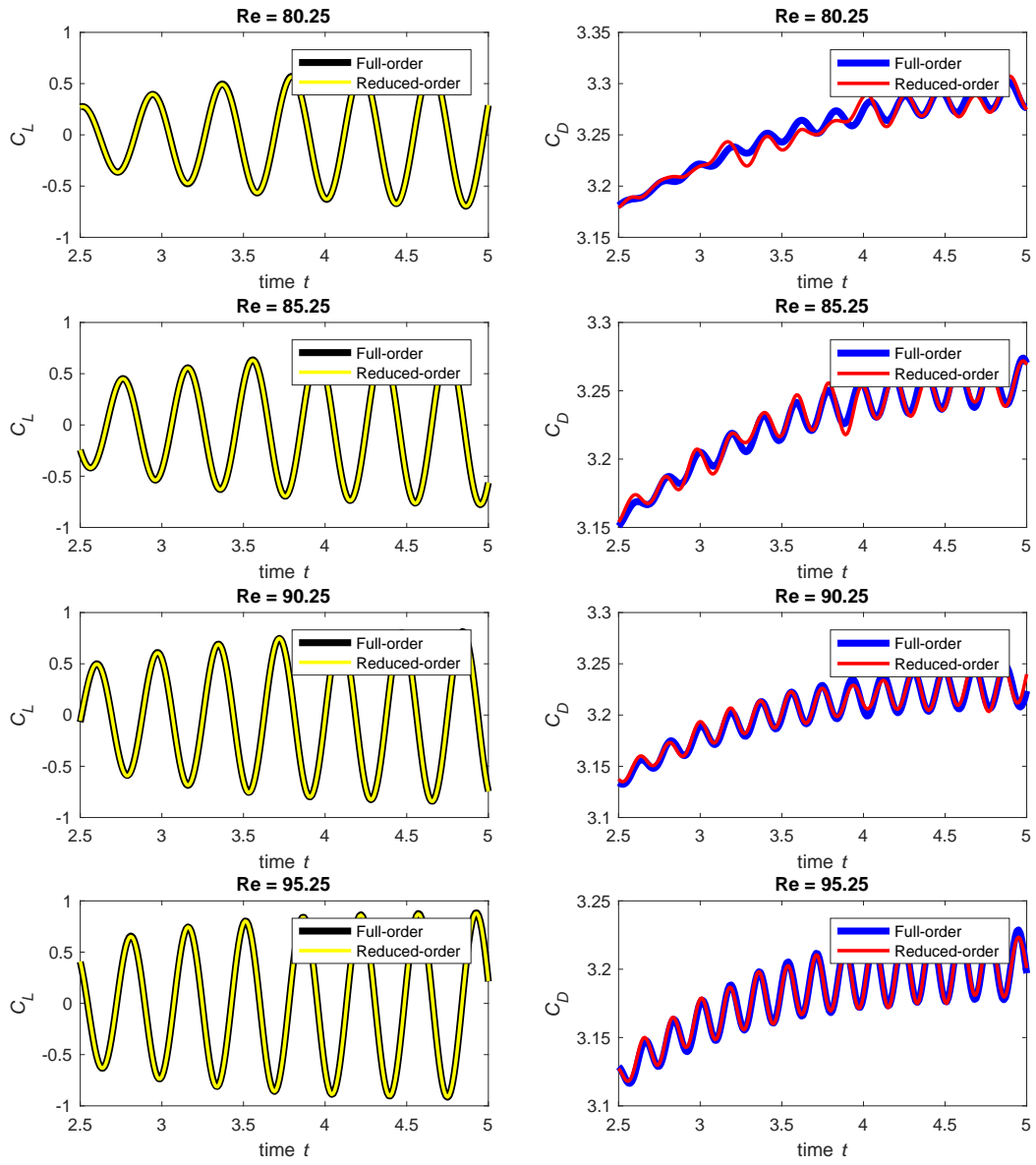
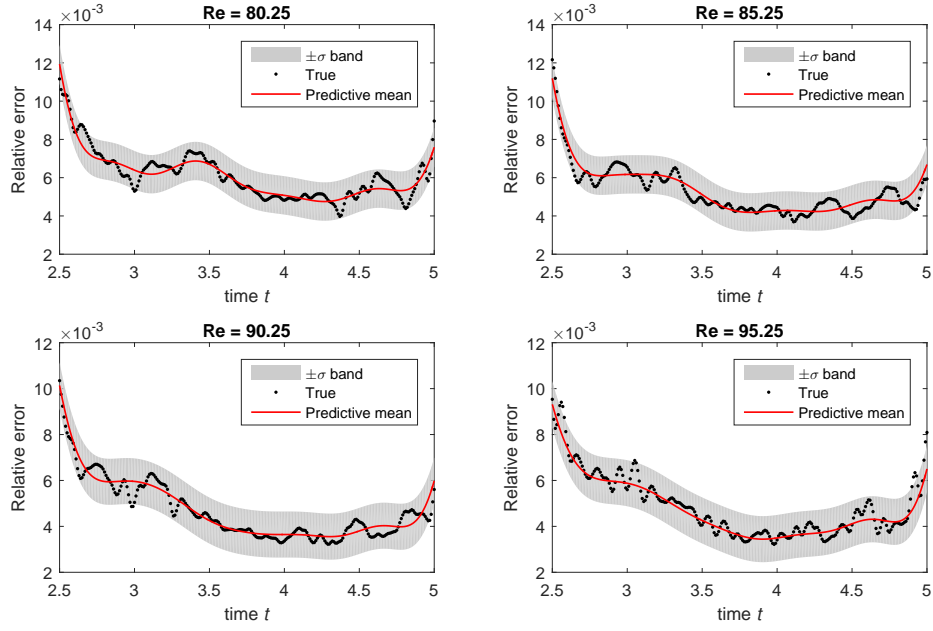
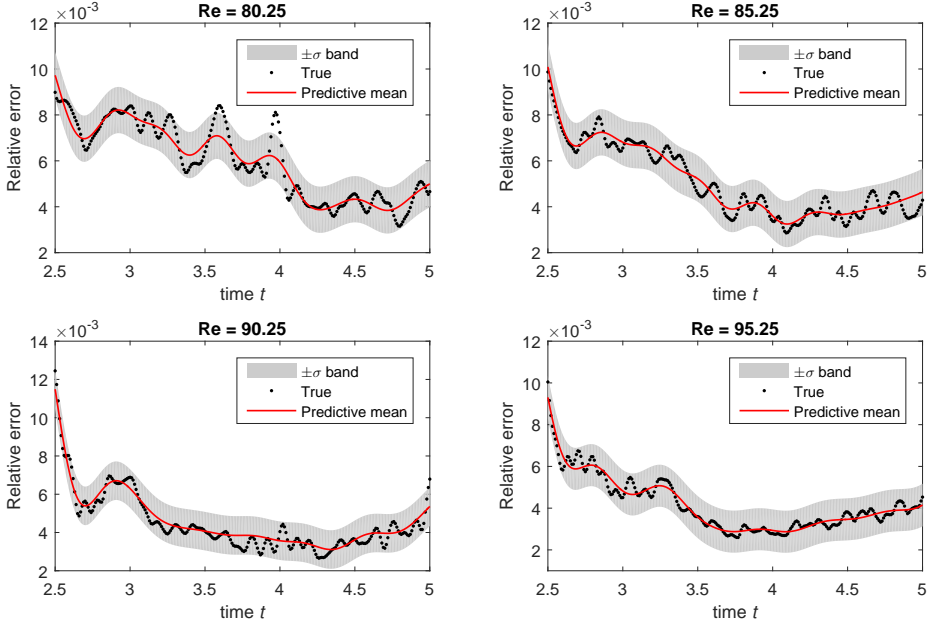


Figure 10: Comparisons of the drag and lift coefficients extracted from the full-order and reduced-order solutions under different  $Re$  values





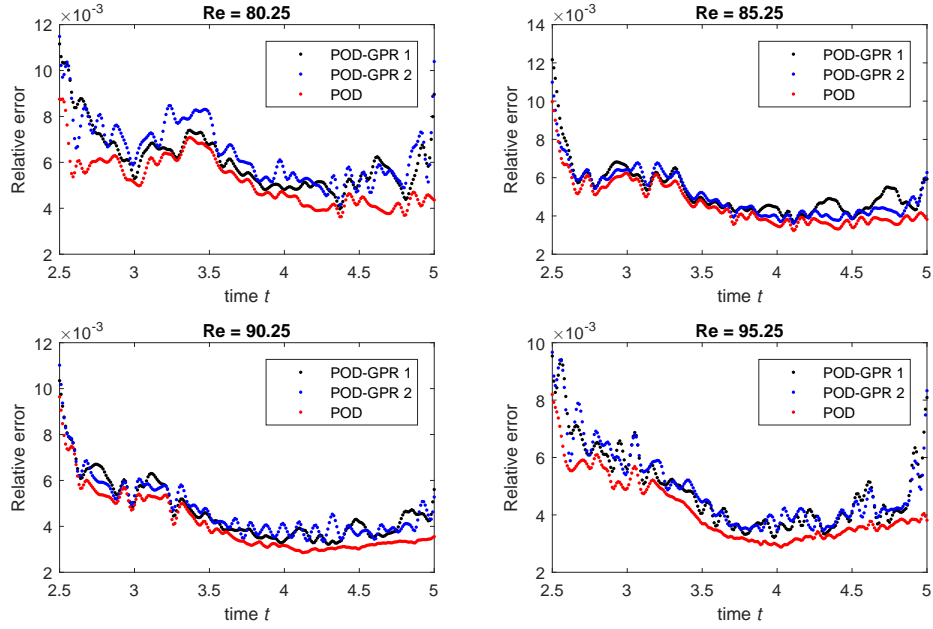
(a)



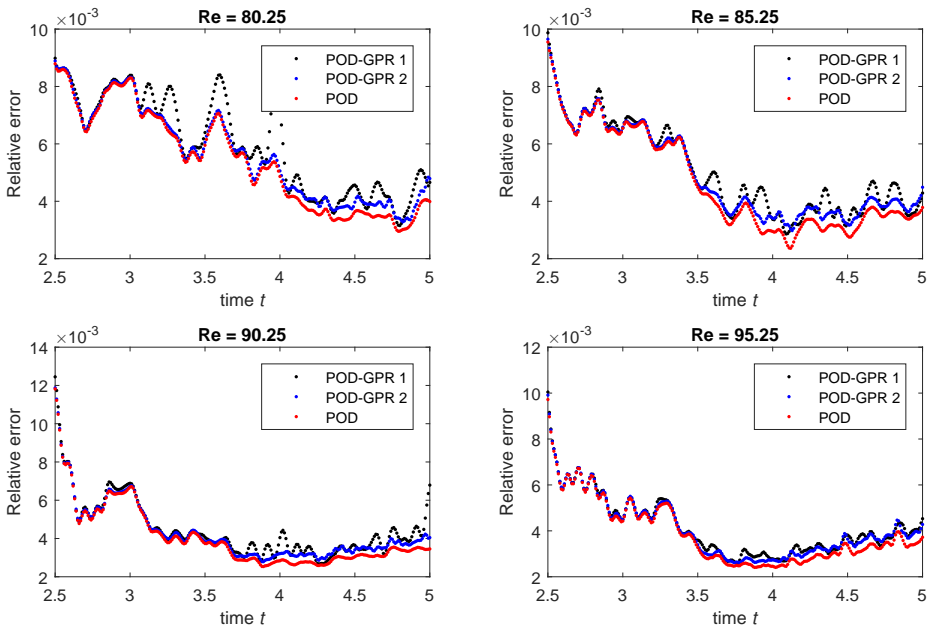
(b)

Figure 11: GP surrogate of the relative errors of the solutions with different  $Re$  values: (a) the velocity field; (b) the pressure field.





(a)



(b)

Figure 12: Relative errors of the solutions at the time steps with different  $Re$  values: (a) the velocity field; (b) the pressure field.

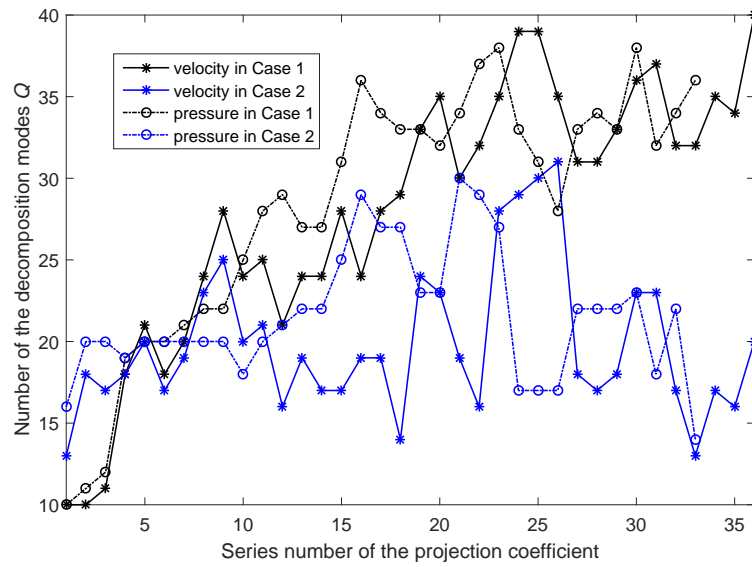


Figure 13: Comparison of the number of the decomposition modes  $Q$  for the projection coefficients

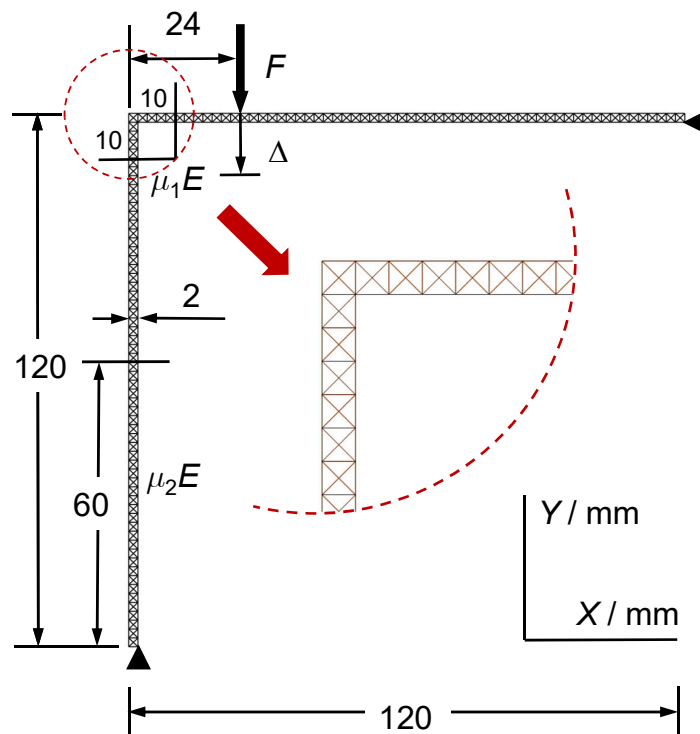


Figure 14: A trussed frame

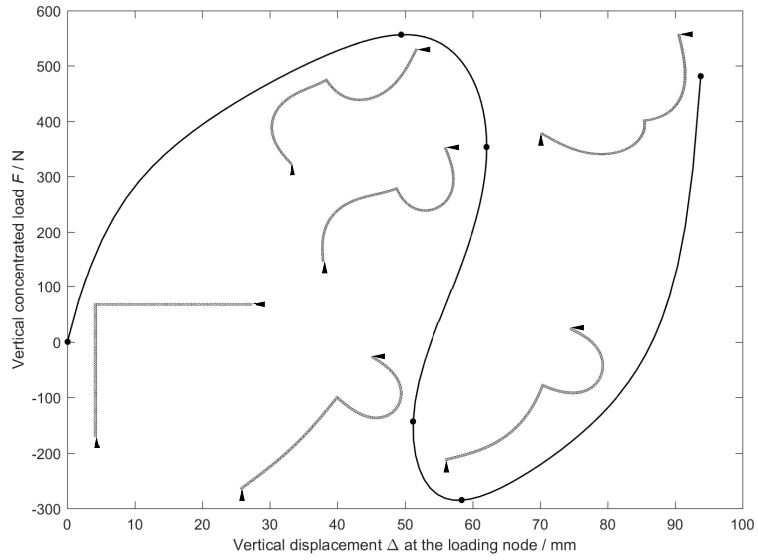


Figure 15: Configurations at different loading stages

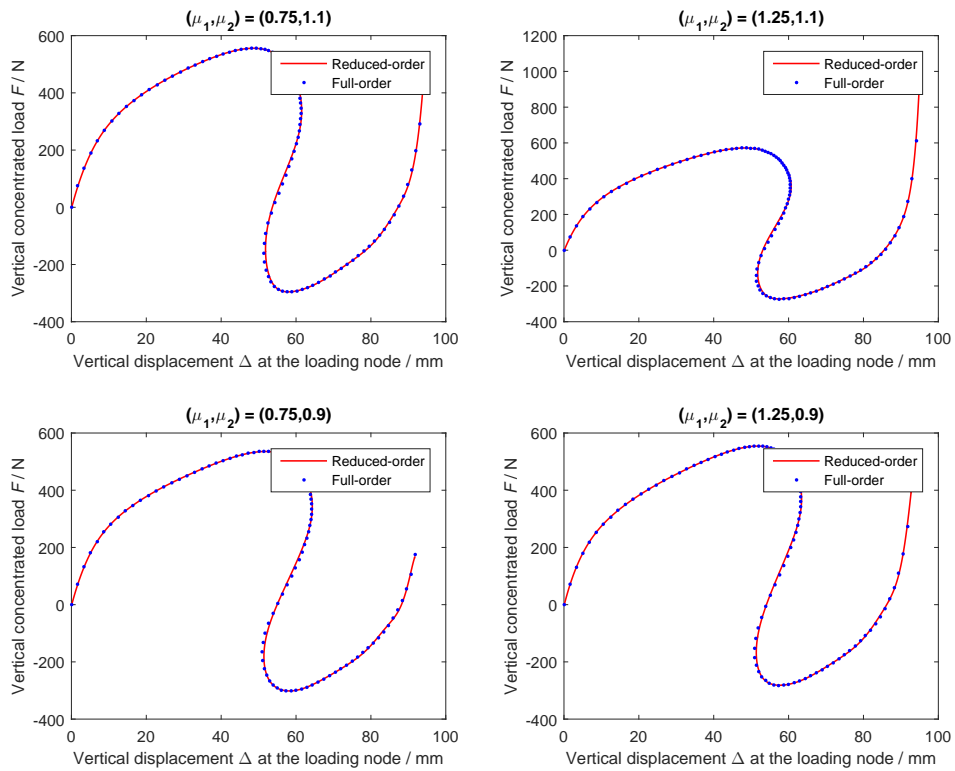


Figure 16: Comparison between the equilibrium paths extracted from the reduced- and the full-order solutions: the hyperelastic case

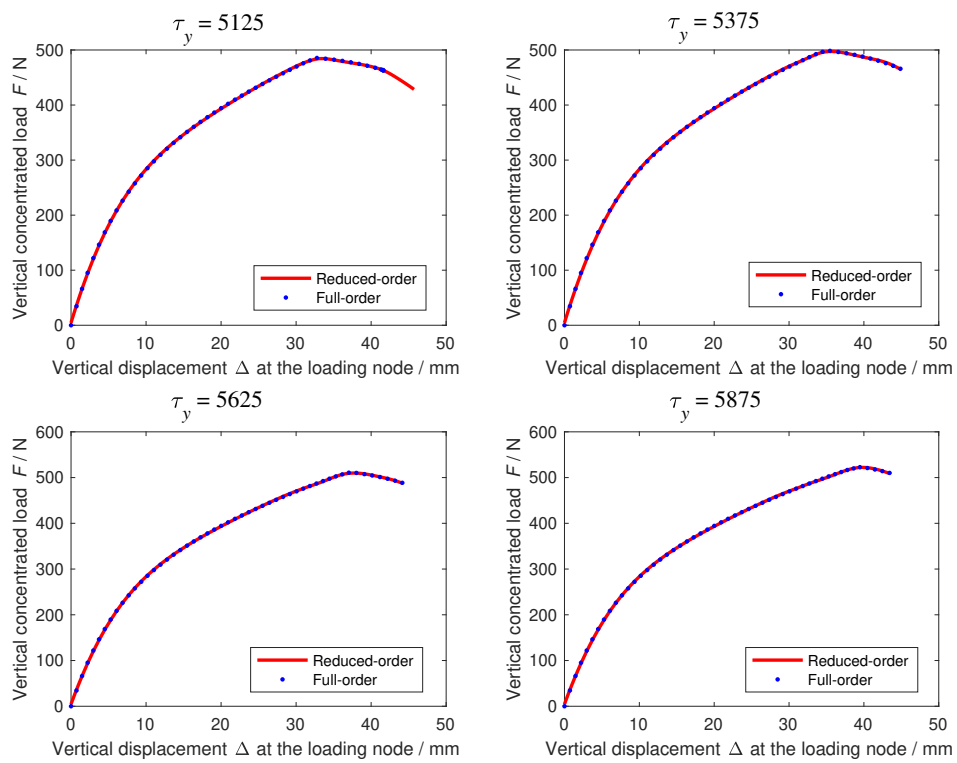


Figure 17: Comparison between the equilibrium paths extracted from the reduced- and the full-order solutions: the hyperelasto-plastic case