



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

**ScienceDirect**

Comput. Methods Appl. Mech. Engrg. 373 (2021) 113495

**Computer methods  
in applied  
mechanics and  
engineering**

[www.elsevier.com/locate/cma](http://www.elsevier.com/locate/cma)

# Data-driven nonintrusive reduced order modeling for dynamical systems with moving boundaries using Gaussian process regression

Zhan Ma, Wenxiao Pan\*

*Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA*

Received 7 April 2020; received in revised form 19 August 2020; accepted 8 October 2020

Available online xxxx

---

## Abstract

We present a data-driven nonintrusive model order reduction method for dynamical systems with moving boundaries. The proposed method draws on the proper orthogonal decomposition, Gaussian process regression, and moving least squares interpolation. It combines several attributes that are not simultaneously satisfied in the existing model order reduction methods for dynamical systems with moving boundaries. Specifically, the method requires only snapshot data of state variables at discrete time instances and the parameters that characterize the boundaries, but not further knowledge of the full-order model and the underlying governing equations. The dynamical systems can be generally nonlinear. The movements of boundaries are not limited to prescribed or periodic motions but can be free motions. In addition, we numerically investigate the ability of the reduced order model constructed by the proposed method to forecast the full-order solutions for future times beyond the range of snapshot data. The error analysis for the proposed reduced order modeling and the criteria to determine the furthest forecast time are also provided. Through numerical experiments, we assess the accuracy and efficiency of the proposed method in several benchmark problems. The snapshot data used to construct and validate the reduced order model are from analytical/numerical solutions and experimental measurements.

© 2020 Elsevier B.V. All rights reserved.

**Keywords:** Reduced order modeling; Nonintrusive model order reduction; Data-driven model reduction; Gaussian process; Proper orthogonal decomposition; Moving boundaries

---

## 1. Introduction

Many natural and engineering systems can exhibit complex dynamics with a wide range of temporal and spatial features. In the analysis of these systems, a first step is to seek and extract dominant features or modes [1–4]. This step typically starts with a modal decomposition of a data set of state variables of interest attained at discrete time instances from experiments or computations. The proper orthogonal decomposition (POD) is one of the most widely used techniques for accomplishing modal decomposition, which can decompose the set of data into spatially dependent POD bases and temporally dependent coefficients. The dominant POD modes can effectively capture the major dynamical evolution of the non-stationary state variables and thereby provide a means to describe a complex dynamical system in a low-dimensional form, the so-called reduced-order model (ROM).

---

\* Corresponding author.

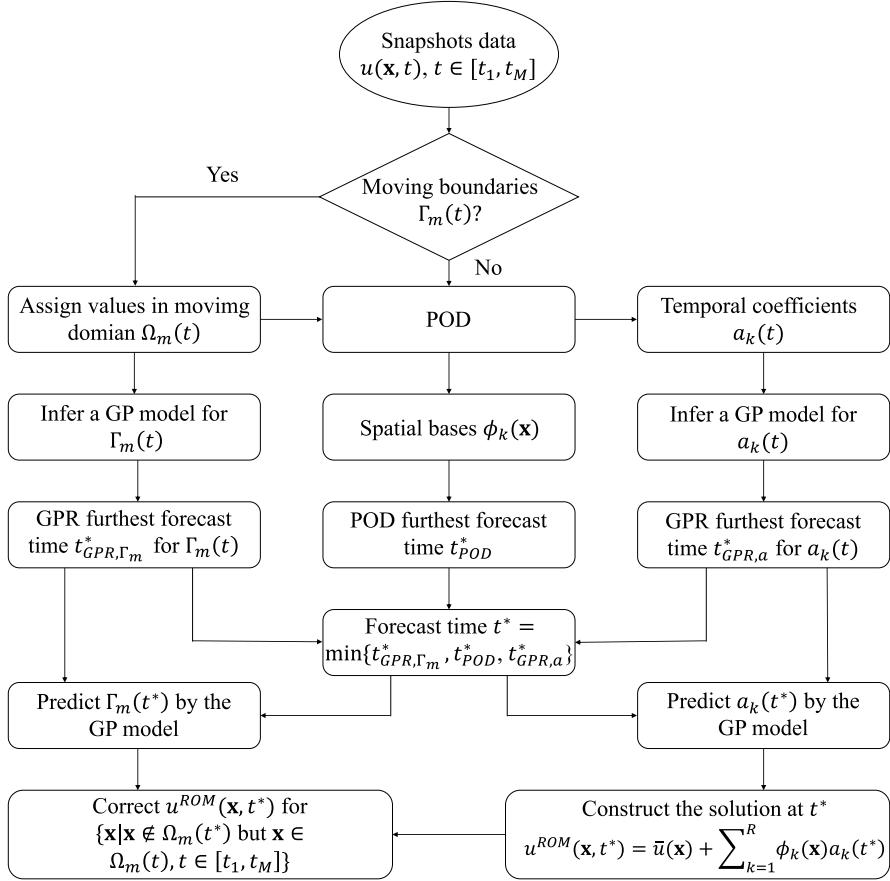
E-mail address: [wpan9@wisc.edu](mailto:wpan9@wisc.edu) (W. Pan).

To determine the temporal coefficient for each POD mode retained, both intrusive and non-intrusive approaches have been developed. The Galerkin projection projects the full model (usually partial differential equations (PDEs)) onto the space of truncated POD modes. Through the orthonormality and energy-optimality attributes of the POD bases, a truncated set of coupled ordinary differential equations (ODEs) is obtained, which constitutes a low-order dynamical system and governs the evolution of the temporal coefficients. By solving the ODEs, usually numerically, the temporal coefficients can be determined. Along with the spatial POD basis functions, such constructed ROM can then be used to reproduce the full-order solutions. Since the Galerkin projection requires the prior knowledge of the governing PDEs, this approach is referred to as intrusive model reduction. The quadratic nonlinearity and triadic interactions in the ODEs derived from Galerkin projection call for a computational load in the order of  $\mathcal{O}(R^3)$ , where  $R$  is the number of POD modes retained in the ROM. Alternative to the Galerkin projection, nonintrusive approaches can be employed to determine the temporal modes. For example, replace the set of ODEs with a set of hyper surfaces, which is constructed using the interpolation methods such as the Smolyak sparse grid [5] and radial basis function (RBF) [6,7]. The work in [2] proposed to infer the operators of the ODEs via the least squares optimization [2]. Wan et al. employed the Gaussian Processes to model the temporal modes [3]. In addition, several efforts adopted deep learning techniques and trained a neural network model for the temporal modes in the form of the artificial [8] or deep feedforward neural network [9], the long short-term memory (LSTM) recurrent neural network [4,10], or the temporal convolutional neural network [11]. These nonintrusive approaches can effectively capture the generally nonlinear time evolution of the temporal coefficients without performing the Galerkin projection. Among them, some are limited to predictions within the database range [2]; and some can predict both inside and outside the database range [3,4,9]. The non-intrusive approaches do not necessarily require the exact form of the full-order equations and hence are applicable to experimental data where the governing equations are often not well established or the associated parameters have considerable uncertainties.

The above-mentioned model order reduction methods are applicable to fixed-domain problems. For many systems with moving objects/boundaries, e.g., in fluid–solid interactions, additional care is needed to derive the ROM. The efforts reported in literature also fall into either of two categories: intrusive or nonintrusive. And those efforts mainly focus on fluid–solid interaction problems. In the intrusive category, the Navier–Stokes (NS) equations that govern the fluid flow are extended to the solid domain. The POD modes are computed for the combined fluid–solid domain. The extended NS equations are then projected via Galerkin projection onto the POD modes to yield the low-order dynamical system with extra terms related to the solid motion. Liberge and Hamdouni proposed a multiphase method similar to the fictitious domain method to extend the NS equations to the solid domain by using a penalization method and a Lagrangian multiplier [12]. Gao and Wei proposed to add extra body-force terms to the NS equations, similar to the immersed boundary method [13,14]. For generally unprescribed solid motion, the additional terms contributed by the unsteady solid motion must be recomputed at each time step. The significantly increased computational cost by these terms can overshadow the benefit of reduced order modeling. To alleviate this issue, the most expensive nonlinear term was neglected in practice [13]. Those intrusive approaches were validated by the tests on a rigid cylinder or sphere oscillating in a fluid [12–14], and only prescribed motion was considered in [13,14]. In the non-intrusive category, Xiao et al. [15] proposed to employ the POD and RBF multi-dimensional interpolation to construct the ROM for fluid–solid interaction problems with a free-moving solid body in the fluid. In this work, the predictions of the ROM were examined for the time instances inside the database range.

In this paper, we present a new data-driven nonintrusive model order reduction method for dynamical systems involving moving boundaries. The dynamical systems can be generally nonlinear. The proposed method is based on the POD, Gaussian process regression (GPR) and moving least squares (MLS) interpolation. It combines three attributes that are not simultaneously satisfied in the existing reduced order modeling methods for dynamical systems with moving boundaries [12,13,15]. First, our method only needs snapshots of state variables at discrete time instances and the parameters that characterize the boundaries. Otherwise, it does not require any prior knowledge of the full-order model or the underlying governing equations for construction of the ROM. The snapshot data can be from simulations or experiments. Second, the moving boundaries are not limited to prescribed or periodic motions. Third, the ability of the ROM constructed to forecast the full-order solutions for future times beyond the range of snapshot data is studied. We provide the criteria to determine the furthest future time that the ROM constructed from a given set of data can predict. Fig. 1 summarizes the main components of the proposed method.

The paper is organized as follows. Section 2 explains in detail each component of the proposed method and provides algorithms for practical implementation. In Section 3, we outline the results of numerical experiments,



**Fig. 1.** Schematic of the proposed data-driven nonintrusive model order reduction method for dynamical systems with moving boundaries.

where the proposed method is assessed in several benchmark problems. We first present the benchmarks with fixed domains only and then proceed to the cases with moving boundaries. The data used to construct and validate the ROMs are from analytical/numerical solutions and experiments. Finally, we conclude in Section 4 and discuss the limitation and possible extensions of the present work.

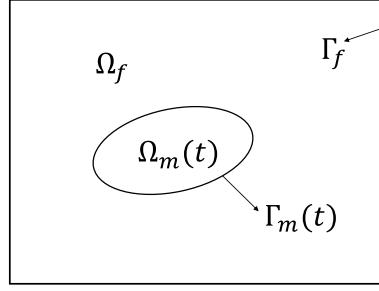
## 2. Methodology

Suppose  $u(\mathbf{x}, t)$  is the full-order solution for a dynamical system. The spatial domain and boundaries for this dynamical system generally consist of the fixed domain  $\Omega_f$ , the moving domain  $\Omega_m$  occupied by other phase(s) (e.g., a cavity or a moving solid body), the fixed boundary  $\Gamma_f$ , and the moving boundary  $\Gamma_m$ , as illustrated in Fig. 2.

### 2.1. POD

POD starts from the snapshot data of the full-order solutions of the dynamical system, which can be obtained from either numerical simulations or experimental measurements. The snapshot data correspond to  $u(\mathbf{x}, t_i)$  at  $M$  time instances with  $t_i \in \{t_1, t_2, \dots, t_M\}$ . Any  $u(\mathbf{x}, t_i)$  may be decomposed into

$$u(\mathbf{x}, t_i) = \bar{u}(\mathbf{x}) + \hat{u}(\mathbf{x}, t_i), \quad \bar{u}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M u(\mathbf{x}, t_i), \quad (1)$$



**Fig. 2.** Schematic of the setup of spatial domain and boundaries.

where  $\bar{u}$  is the temporal mean of snapshot data and  $\hat{u}$  is the fluctuating part. The task is to find a set of spatial bases  $\phi_k(\mathbf{x})$  and temporal coefficients  $a_k(t)$  such that the fluctuating part  $\hat{u}$  can be reproduced as:

$$\hat{u}(\mathbf{x}, t) = \sum_{k=1}^M a_k(t) \phi_k(\mathbf{x}). \quad (2)$$

To this end, the correlation matrix  $\mathbf{A} \in \mathbb{R}^{M \times M}$  of the fluctuating parts of snapshot data is constructed by:

$$A_{ij} = \int_{\Omega} \hat{u}(\mathbf{x}, t_i) \hat{u}(\mathbf{x}, t_j) d\mathbf{x}, \quad (3)$$

where  $i$  and  $j$  refer to the  $i$ th and  $j$ th snapshots, respectively. By definition,  $\mathbf{A}$  is symmetric and positive semi-definite. Thus, the set of spatial bases  $\phi_k(\mathbf{x})$  (i.e., POD basis function) can be obtained via eigendecomposition of  $\mathbf{A}$ :

$$\phi_k(\mathbf{x}) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^M w_i^k \hat{u}(\mathbf{x}, t_i), \quad (4)$$

where  $\{\lambda_1, \lambda_2, \dots, \lambda_M\}$  are eigenvalues in descending order and  $\mathbf{w}^k$  is the eigenvector corresponding to  $\lambda_k$ . The POD basis functions satisfy the condition of orthogonality:

$$(\phi_k, \phi_l) = \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases}, \quad (5)$$

where  $(\phi_k, \phi_l) = \int_{\Omega} \phi_k(\mathbf{x}) \phi_l(\mathbf{x}) d\mathbf{x}$  defines the inner-product of two functions.

For model order reduction, only the first  $R \ll M$  largest eigenvectors (or POD modes) are retained. As a result, a snapshot  $\hat{u}(\mathbf{x}, t_i)$  can be approximated as:

$$\hat{u}(\mathbf{x}, t_i) \approx \hat{u}^{POD}(\mathbf{x}, t_i) = \sum_{k=1}^R \sqrt{\lambda_k} w_k^i \phi_k(\mathbf{x}) \quad (6)$$

with the following truncation error:

$$\epsilon^{POD} = \frac{1}{M} \sum_{i=1}^M \left\| \hat{u}(\mathbf{x}, t_i) - \hat{u}^{POD}(\mathbf{x}, t_i) \right\|_{L_2} = \frac{1}{M} \sum_{i=1}^M \left\| \sum_{k=R+1}^M \sqrt{\lambda_k} w_k^i \phi_k(\mathbf{x}) \right\|_{L_2} = \sqrt{\sum_{k=R+1}^M \lambda_k}. \quad (7)$$

By increasing  $R$ , i.e., by keeping more POD modes, the truncation error of POD ( $\epsilon^{POD}$ ) can be reduced. Given Eq. (6), Eq. (2) can be truncated to  $R$  dominant POD modes as:

$$\hat{u}(\mathbf{x}, t) \approx \sum_{k=1}^R a_k(t) \phi_k(\mathbf{x}). \quad (8)$$

Define the relative root mean square (RRMS) truncation error as:

$$RRMS\ Error = \sqrt{\frac{\sum_{k=R+1}^M \lambda_k}{\sum_{k=1}^M \lambda_k}}. \quad (9)$$

By requiring  $RRMS\ Error < \alpha^{POD}$  with  $\alpha^{POD}$  the preset threshold, we can determine how many dominant POD modes ( $R$ ) to retain.

Once the spatial POD bases  $\phi_k(\mathbf{x})$  are determined from Eq. (4), the next task is to construct the temporal coefficients  $a_k(t)$  such that the full-order solution can be effectively reproduced at a given time  $t'$  either within or beyond the database range:

$$u(\mathbf{x}, t') = \bar{u}(\mathbf{x}) + \hat{u}(\mathbf{x}, t') \approx u^{ROM}(\mathbf{x}, t') = \bar{u}(\mathbf{x}) + \sum_{k=1}^R a_k(t') \phi_k(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega_f \text{ and } t' \in [t_1, t_M] \text{ or } t' > t_M. \quad (10)$$

Eq. (10) is hence the ROM constructed via POD.

## 2.2. Classical intrusive model order reduction based on Galerkin projection

In this section, we review the classical, intrusive approach to determine the temporal coefficients  $a_k(t)$  based on the Galerkin projection. Assume the dynamical system of interest can be described by the following full-order model:

$$\dot{u}(\mathbf{x}, t) = \mathcal{F}(u), \quad (11)$$

where  $\mathcal{F}$  denotes general operators, e.g., spatial differential operators. To determine  $a_k(t)$ , the idea of Galerkin projection is to project the original full-order model (Eq. (11)) onto the POD manifold, by which and noting the orthogonality of POD basis functions a set of ordinary differential equations (ODEs) of  $a_k(t)$  can be derived as:

$$\dot{a}_k(t) = (\mathcal{F}(u), \phi_k), \quad k = 1, 2, \dots, R. \quad (12)$$

To close the above ODEs, the initial conditions can be given using the following projection:

$$a_k(t = 0) = (u(\mathbf{x}, t = 0) - \bar{u}(\mathbf{x}), \phi_k). \quad (13)$$

Hence, to solve for  $a_k(t)$ , one needs to evaluate the inner product in the right hand side of Eq. (12), which requires the knowledge of  $\mathcal{F}$ . If  $\mathcal{F}$  is not attainable for “black-box” problems, e.g., for experimental data, the Galerkin projection is not feasible. For problems with moving boundaries, even if  $\mathcal{F}$  is known, it needs to include the contributions from the moving boundaries, and hence evaluating the inner product  $(\mathcal{F}(u), \phi_k)$  demands expensive computation. For example, in fluid–solid interactions, the NS equations that govern the fluid flow are augmented with extra body-force terms to account for the coupling of fluid flow and solid motion [13,14]. For generally unsteady solid motion, the contributions of these extra terms in the inner product  $(\mathcal{F}(u), \phi_k)$  must be recomputed at each time step, which significantly increases the computational cost of the ROM. In addition, the ODEs in Eq. (12) usually must be numerically solved. Thus, the temporal integrator and time step size must be chosen properly for the desired accuracy and stability.

## 2.3. Gaussian process-enabled nonintrusive model order reduction

In this paper, we propose a Gaussian process-enabled nonintrusive model order reduction method. Different from the Galerkin projection discussed in Section 2.2, the proposed method does not rely on any prior knowledge of the full-model operator and is well suited for black-box problems or the problems requiring nontrivial evaluation and solution of Eq. (12). For each temporal coefficient  $a_k(t)$ , we infer a Gaussian process model from the dataset  $a_k(t_i)$  for  $i = 1, \dots, M$ . The obtained Gaussian process model is then employed to predict  $a_k(t')$  at a given time  $t'$ , where  $t' > t_1$  can be within ( $t' < t_M$ ) or beyond ( $t' > t_M$ ) the dataset range. We provide the criteria to determine the furthest time extrapolation permitted in this method. Once  $a_k(t')$  is predicted, the full-order solution can be reconstructed from Eq. (10).

### 2.3.1. Inference of temporal coefficients via Gaussian process

Consider the data set  $\mathbf{t} = [t_1, \dots, t_M]^T$  as the training inputs and  $\mathbf{t}' = [t'_1, t'_2, \dots, t'_N]^T$  as the inputs for prediction. For each  $k = 1, 2, \dots, R$ , the training outputs are  $\mathbf{y} = [a_k(t_1), a_k(t_2), \dots, a_k(t_M)]^T$ , and the predicted outputs are  $\mathbf{y}' = [a_k(t'_1), a_k(t'_2), \dots, a_k(t'_N)]^T$ . The Gaussian process model is given by:  $\mathbf{y}(\mathbf{t}) \sim \mathcal{GP}(\boldsymbol{\mu}(\mathbf{t}), \mathbf{C}(\mathbf{t}, \mathbf{t}))$ , where  $\boldsymbol{\mu}(\mathbf{t})$  is the mean function and  $\mathbf{C}(\mathbf{t}, \mathbf{t})$  is the covariance function. To predict  $\mathbf{y}'$ , the key is to determine the posterior distribution  $p(\mathbf{y}'|\mathbf{y})$ . Since the joint distribution of the training outputs  $\mathbf{y}$  and the predicted outputs  $\mathbf{y}'$  satisfy:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}' \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \boldsymbol{\mu}(\mathbf{t}) \\ \boldsymbol{\mu}(\mathbf{t}') \end{bmatrix}, \begin{bmatrix} \mathbf{C}(\mathbf{t}, \mathbf{t}) + \sigma^2 \mathbf{I} & \mathbf{C}(\mathbf{t}', \mathbf{t})^T \\ \mathbf{C}(\mathbf{t}', \mathbf{t}) & \mathbf{C}(\mathbf{t}', \mathbf{t}') \end{bmatrix} \right), \quad (14)$$

where  $\sigma^2$  is the variance of identically independent normally distributed noise (with zero mean) assumed in the Gaussian process model, the posterior distribution can be determined as:

$$p(\mathbf{y}'|\mathbf{y}) = \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\mathbf{C}}), \quad (15)$$

where

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \mathbf{C}(\mathbf{t}', \mathbf{t}) [\mathbf{C}(\mathbf{t}, \mathbf{t}) + \sigma^2 \mathbf{I}]^{-1} [\mathbf{y} - \boldsymbol{\mu}(\mathbf{t})] + \boldsymbol{\mu}(\mathbf{t}'), \\ \hat{\mathbf{C}} &= \mathbf{C}(\mathbf{t}', \mathbf{t}') - \mathbf{C}(\mathbf{t}', \mathbf{t}) [\mathbf{C}(\mathbf{t}, \mathbf{t}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{C}(\mathbf{t}', \mathbf{t})^T. \end{aligned} \quad (16)$$

In this work, the covariance function  $\mathbf{C}(\mathbf{t}, \mathbf{t}')$  is assumed a squared exponential form, i.e.,

$$C_{ij}(t_i, t'_j; \boldsymbol{\theta}) = \theta_f^2 \exp \left[ -\frac{1}{2} \theta_l^2 (t_i - t'_j)^2 \right], \quad (17)$$

where  $\boldsymbol{\theta} = (\theta_f, \theta_l)$  denotes the hyper-parameters. To determine the hyper-parameters  $\boldsymbol{\theta}$  as well as  $\sigma^2$  in Eq. (16), we minimize the negative log marginal likelihood [16]:

$$-\log p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2) = \frac{1}{2} \mathbf{y}^T \mathbf{C}^{-1} \mathbf{y} + \frac{1}{2} \log |\mathbf{C}| + \frac{M}{2} \log(2\pi), \quad (18)$$

where  $\mathbf{C} = \mathbf{C}(\mathbf{t}, \mathbf{t}) + \sigma^2 \mathbf{I}$ , via the Quasi-Newton optimizer L-BFGS [17]. Here,  $M$  is the total number of training data; and  $|\mathbf{C}|$  is the determinant of matrix  $\mathbf{C}$ . The marginal likelihood as in Eq. (18) is chosen because it entails a trade-off between data-fit and model complexity: while the term  $\frac{1}{2} \mathbf{y}^T \mathbf{C} \mathbf{y}$  targets better fitting the training data, the term  $\log |\mathbf{C}|$  penalizes the model complexity. A key advantage of GPR is that uncertainty bounds can be analytically derived from the hyper-parameters.

The method proposed herein is different from that in [3], where the time derivative  $\dot{a}_k = f(a_1, \dots, a_R)$  is modeled as a Gaussian process, instead of  $a_k(t)$  as in this work. To obtain an accurate model for the time derivative, it requires to approximate the time derivative by a numerical integrator of high-order accuracy in the training process. To predict  $a_k(t' > t_M)$  from the trained model of  $\dot{a}_k$ , it needs to march from  $a_k(t_M)$  by the increment of  $\delta t$  (time step size) each time and finally reaches  $t'$ . Also, compared with training a model for  $a_k(t)$ , it needs more training data. In addition, if the data are noisy, e.g., from experimental measurements, training an accurate model for the time derivative can be challenging. Due to these considerations, we choose to model  $a_k(t)$  itself as a Gaussian process.

### 2.3.2. Criteria for furthest forecast time

In this section, we propose two criteria to determine the furthest time extrapolation that the ROM can achieve. Since the ROM is constructed via POD and GPR, each of them poses a constraint on the time extrapolation.

To forecast the solution at  $t' > t_M$ , the same POD subspace is assumed, which is constructed by the dominant spatial bases extracted from the snapshot data. The error arising from this assumption constrains the time extrapolation of the ROM. While in some cases, the dominant POD bases vary slowly in time, e.g., flow dynamics at low Reynolds numbers described by the Burgers equation; in other cases, they can change rapidly, e.g., shock wave and advection problems. Theoretically, the decay rate of Kolmogorov  $n$ -width [18,19] can measure how fast the POD spatial bases change in time for a dynamical system. The Kolmogorov  $n$ -width is defined as [20]:

$$d_n(\mathcal{M}) = \inf_{\mathcal{S}_n} \sup_{u \in \mathcal{M}} \min_{v \in \mathcal{S}_n} \|u - v\|, \quad (19)$$

where  $\mathcal{M}$  is the manifold of the full-order solutions over the entire time considered, including the snapshot data;  $\mathcal{S}_n$  denotes  $n$ -dimensional linear subspaces constructed from the snapshot data;  $u$  is a full-order solution in  $\mathcal{M}$ ; and

$v$  represents a reduced-order solution in  $\mathcal{S}_n$ . In the context of this paper,  $\mathcal{S}_n$  refers to the subspaces formed by the first  $R$  POD spatial bases. The Kolmogorov  $n$ -width as in Eq. (19) provides the worst-case error resulting from a projection onto the best-possible linear subspace of a given dimension  $n$  [20]. If the Kolmogorov  $n$ -width decays faster with respect to  $n$ , the reduced-order solution in the subspace constructed by the first  $R$  POD bases can be an accurate approximation of the full-order solution for longer time beyond the range of snapshot data. In practice, a direct evaluation of the Kolmogorov  $n$ -width and its decay rate is difficult. In this paper, we instead estimate the decay rate of dominant eigenvalues, i.e.,  $\frac{\ln \lambda_2 - \ln \lambda_{R+2}}{R}$ . Note that  $\lambda_2$  dominates the POD truncation error if only the first POD mode is retained in the ROM;  $\lambda_{R+2}$  dominates the POD truncation error if the first  $R+1$  modes are retained in the ROM. We employ this decay rate of dominant eigenvalues to characterize how fast the POD spatial bases change in time for a dynamical system. Based on that, we propose the following constraint posed by POD on the furthest forecast time of the ROM:

$$\frac{\Delta t_{\text{POD}}^*}{\Delta t_{\text{S snapshots}}} \leq \beta^{\text{POD}} \frac{\ln \lambda_2 - \ln \lambda_{R+2}}{R}, \quad (20)$$

where  $\Delta t_{\text{S snapshots}} = t_M - t_1$  denotes the time span of snapshot data;  $\Delta t_{\text{POD}}^* = t_{\text{POD}}^* - t_M$  defines how long the furthest forecast time  $t_{\text{POD}}^*$  is beyond  $t_M$  (the latest time of snapshot data); and  $\beta^{\text{POD}} < 1$  is a non-dimensional constant related to the tolerance for the error induced by the change of POD spatial bases. Eq. (20) provides the criterion to determine the furthest forecast time  $t_{\text{POD}}^*$  permitted in time extrapolation using the dominant POD spatial bases extracted from the  $M$  snapshot data. Note that although the eigenvalues can be over/under-estimated by picking different snapshots in a fixed range, the decay rate of eigenvalues has little change if we pick different snapshots in the same range. Thus, the criterion in Eq. (20) does not require snapshots to be uniformly sampled.

The second criterion is from the uncertainty level of GPR for predicting each  $a_k$  at a given time  $t'$ . The standard deviation (or uncertainty level)  $\hat{\sigma}_k(t')$  grows when the forecast time  $t'$  is further from the snapshot data, and hence it poses a constraint on the furthest forecast time  $t_{\text{GPR},a}^*$  permitted by GPR. Considering each temporal mode  $a_k$  contributes differently in the total energy of the dynamical system, the weighted average of standard deviation is defined to measure the uncertainty level:

$$\hat{\sigma}(t') = \frac{\sum_{k=1}^R \lambda_k \hat{\sigma}_k(t')}{\sum_{k=1}^R \lambda_k}. \quad (21)$$

And based on that, we propose the following constraint posed by GPR on the furthest forecast time of the ROM:

$$\frac{\sum_{k=1}^R \lambda_k \hat{\sigma}_k(t_{\text{GPR},a}^*)}{\sum_{k=1}^R \lambda_k |\hat{\mu}_k(t_{\text{GPR},a}^*)|} \leq \beta^{\text{GPR},a} \quad (22)$$

where  $\beta^{\text{GPR},a}$  denotes the preset tolerance. Larger  $\beta^{\text{GPR},a}$  means larger tolerance for the uncertainty associated in GPR. From Eq. (22), the furthest forecast time permitted in time extrapolation by GPR,  $t_{\text{GPR},a}^*$ , can be adaptively determined according to the preset tolerance.

### 2.3.3. Algorithm for Gaussian process-enabled nonintrusive model order reduction

The procedure of the proposed nonintrusive model order reduction method based on the POD and GPR is summarized in Algorithm 1.

When the numerical simulations or experimental measurements are demanding or expensive for a dynamical system, one can employ the ROM to forecast the full-order solutions at future times beyond the database range. As we have discussed in Section 2.3.2, the furthest forecast time is constrained by the POD and GPR. Thus, after the furthest forecast time of the ROM is reached, the consecutive prediction would still need numerical simulations or experimental measurements. Thus, for long-time prediction of a dynamical system, the numerical simulations/experimental measurements and the ROM can be called alternatively in an automated process:

1. Generate  $M$  snapshot data using numerical simulations/experimental measurements.
2. Construct a ROM from the snapshot data and determine the furthest forecast time  $t^*$  by the proposed criteria.
3. Call the ROM to predict the full-order solutions at any desired times until  $t = t^*$ . The full-order solution at  $t^*$  is taken as the initial condition for the consecutive numerical simulations/experimental measurements.
4. Repeat Steps 1–3 until the target prediction time.

**Algorithm 1** Gaussian process-enabled nonintrusive model order reduction**Input:**  $M$  snapshot data of  $u(\mathbf{x}, t_i)$ ,  $i = 1, 2, \dots, M$ **Output:**  $u(\mathbf{x}, t^*)$  predicted at  $t^* > t_M$ 

- 1: Determine  $R$  dominant POD spatial bases  $\phi_k(\mathbf{x})$ ,  $k = 1, 2, \dots, R$ , with  $RRMS\ Error < \alpha^{POD}$
- 2: Project  $u(\mathbf{x}, t_i)$  to the reduced space and obtain  $a_k(t) = (\hat{u}(\mathbf{x}, t), \phi_k(\mathbf{x}))$
- 3: Determine the furthest forecast time permitted by the POD,  $t_{POD}^*$ , from Eq. (20)
- 4: **for**  $k = 1, 2, \dots, R$  **do**
- 5:     Infer a Gaussian Process model for  $a_k(t)$  following Section 2.3.1
- 6:     Determine the furthest forecast time permitted by the GPR,  $t_{GPR,a}^*$ , from Eq. (22)
- 7: **end for**
- 8: Determine the final furthest forecast time as  $t^* = \min\{t_{POD}^*, t_{GPR,a}^*\}$
- 9: Predict each  $a_k(t^*)$  by the inferred Gaussian Process models
- 10: Reconstruct the full-order solution  $u(\mathbf{x}, t^*)$  by Eq. (10)
- 11: **return**  $u(\mathbf{x}, t^*)$

This process adaptively combines numerical simulations/experimental measurements with ROMs and hence optimizes the efficiency for long-time prediction of a dynamical system.

## 2.4. Nonintrusive model order reduction for problems with moving boundaries

In the presence of time-evolving boundaries  $\Gamma_m(t)$ , reproducing the full-order solution from Eq. (10) needs additional efforts. First, the snapshot data are also needed in  $\Omega_m$ . Thus, we need to assign the values of  $\hat{u}(\mathbf{x}, t_i)$  for  $\mathbf{x} \in \Omega_m$  and  $t_i \in \{t_1, t_2, \dots, t_M\}$ . If a moving solid body is involved (in fluid–solid interaction), the values of the velocity in  $\Omega_m$  coincide with the motion of the solid body; and the values of the pressure in  $\Omega_m$  are assigned via interpolation of the surrounding fluid pressure in  $\Omega_f$ . If a cavity is involved, the values of the state variable in  $\Omega_m$  are extrapolated by least squares from the values in  $\Omega_f$  near the cavity. With the snapshot data given in both  $\Omega_f$  and  $\Omega_m$ , the POD can be conducted for the entire domain  $\Omega_f \cup \Omega_m$ .

### 2.4.1. Inference of the time-evolution of moving boundaries

The next task is to infer the time-evolution of the moving boundaries given a set of trajectory data. We assume the evolution of a moving boundary can be fully determined by a finite set of parameters, for example, the radius of a spherical surface (see Section 3.4) or the translational and rotational displacements and velocities of the center of mass of a rigid solid body (see Section 3.5). Without loss of generality, we denote the parameters that fully characterize a time-evolving boundary  $\Gamma_m(t)$  as  $\gamma(t) = [\gamma_1(t), \gamma_2(t), \dots]$ . For each  $\gamma_l(t)$ , we seek a Gaussian process model using  $\mathbf{t} = [t_1, \dots, t_M]^T$  as the training inputs and  $[\gamma_l(t_1), \gamma_l(t_2), \dots, \gamma_l(t_M)]^T$  as the training outputs. By the constructed Gaussian process model, we predict  $\gamma_l(\mathbf{t}')$  at a future time series  $\mathbf{t}' > t_M$ . We consider a Gaussian process model  $\gamma_l(\mathbf{t}) \sim \mathcal{GP}(\mu^{\gamma_l}(\mathbf{t}), \mathbf{C}^{\gamma_l}(\mathbf{t}, \mathbf{t}'))$ , where  $\mu^{\gamma_l}(\mathbf{t})$  is the mean function and  $\mathbf{C}^{\gamma_l}(\mathbf{t}, \mathbf{t}')$  is the covariance function. The procedure to construct this Gaussian process model is the same as described in Section 2.3.1. The furthest forecast time for  $\Gamma_m(t)$  is determined by  $t_{GPR, \Gamma_m}^* = \min\{t_{GPR, \gamma_1}^*, t_{GPR, \gamma_2}^*, \dots\}$ . And each  $t_{GPR, \gamma_l}^*$  for  $\gamma_l(t)$  is determined from:

$$\frac{\hat{\sigma}^{\gamma_l}(t_{GPR, \gamma_l}^*)}{|\hat{\mu}^{\gamma_l}(t_{GPR, \gamma_l}^*)|} \leq \beta^{GPR, \Gamma_m}, \quad (23)$$

where  $\beta^{GPR, \Gamma_m}$  is the preset tolerance.

### 2.4.2. Correction near the moving boundaries

We note that when reproducing the full-order solution in  $\Omega_f$  at a future time  $t' > t_M$ , the largest errors appear in the regions near the moving boundaries on the downstream side. The large errors stem from the fact that those regions were part of  $\Omega_m$  but not in  $\Omega_f$  during  $[t_1, t_M]$ , and the gradient of the solution (e.g., velocity gradient) is discontinuous across  $\Gamma_m$ . Thus, we propose a correction step to recover the accuracy of the prediction for those

regions. In particular, the correction is achieved via high-order interpolation from their neighbor regions using the MLS [21].

More specifically, consider a point  $\mathbf{x}_p \in \Omega_m$  during  $[t_1, t_M]$  but  $\in \Omega_f$  at  $t > t_M$ , a kernel length  $h$  is set to search the neighbor points  $\mathbf{x}_q$  satisfying  $\|\mathbf{x}_q - \mathbf{x}_p\| < h$  and  $\mathbf{x}_q \in \Omega_f$  during  $[t_1, t^*]$ . Here, the furthest forecast time  $t^*$  is determined by  $t^* = \min\{t_{\text{POD}}^*, t_{\text{GPR},a}^*, t_{\text{GPR},\Gamma_m}^*\}$ . A polynomial  $u_h(\mathbf{x}) = \mathbf{P}^\top(\mathbf{x})\mathbf{c}^*$  of order  $s$  is sought to approximate  $u$  at  $\mathbf{x}_p$ , where  $\mathbf{P}(\mathbf{x})$  denotes the polynomial basis. The coefficient vector  $\mathbf{c}^*$  is determined by minimizing the following weighted residual functional:

$$\mathbf{c}_p^* = \underset{\mathbf{c}_p}{\operatorname{argmin}} \sum_q [u(\mathbf{x}_q) - \mathbf{P}^\top(\mathbf{x}_q)\mathbf{c}_p]^2 W_{pq}, \quad (24)$$

where  $q \in \mathbb{N}_{h,p} = \{\mathbf{x}_q \text{ s.t. } \|\mathbf{x}_q - \mathbf{x}_p\| < h, \mathbf{x}_q \in \Omega_f \text{ during } [t_1, t^*]\}$  and  $W_{pq} = W(\|\mathbf{x}_q - \mathbf{x}_p\|)$  with  $W$  a positive function with the compact support  $h$ . The choice of  $h$  is determined by the polynomial order  $s$  to ensure necessarily sufficient neighbor points  $\mathbf{x}_q$  to solve Eq. (24). Due to its polynomial consistency, the MLS interpolation can achieve high-order accuracy by taking large  $s$ , e.g.,  $s = 3$  used in this work. Following standard arguments for the minimization of a symmetric positive definite quadratic form, the solution of Eq. (24) is given by:

$$\mathbf{c}_p^* = \left( \sum_{j \in \mathbb{N}_{h,p}} \mathbf{P}_p(\mathbf{x}_j) W_{pj} \mathbf{P}_p^\top(\mathbf{x}_j) \right)^{-1} \left( \sum_{q \in \mathbb{N}_{h,p}} \mathbf{P}_p(\mathbf{x}_q) W_{pq} u(\mathbf{x}_q) \right). \quad (25)$$

The solution reproduced from the ROM (Eq. (10)) at  $\mathbf{x}_p$  is then replaced with the interpolated value, i.e.,  $u(\mathbf{x}_p) \approx u_h(\mathbf{x}_p) = \mathbf{P}_p^\top(\mathbf{x}_p)\mathbf{c}_p^*$ . By such, we improve the accuracy of the ROM's predictions for the regions that fall in  $\Omega_f$  at  $t > t_M$  but belong to  $\Omega_m$  during  $[t_1, t_M]$ .

#### 2.4.3. Algorithm

Algorithm 2 outlines the procedure of the proposed nonintrusive model order reduction method for problems with moving boundaries, which augments Algorithm 1 with the steps to infer the time evolution of the moving boundaries and to correct the solutions for the regions in  $\Omega_f$  at  $t^* > t_M$  but in  $\Omega_m$  during  $[t_1, t_M]$ .

---

#### Algorithm 2 Nonintrusive model order reduction for problems with moving boundaries

---

**Input:**  $M$  snapshot data of  $u(\mathbf{x}, t_i)$  and  $\gamma_l(t_i)$  with  $i = 1, 2, \dots, M$  and  $l = 1, 2, \dots$   
**Output:**  $u(\mathbf{x}, t^*)$  and  $\gamma_l(t^*)$  predicted at  $t^*$

- 1: **for**  $l = 1, 2, \dots$  **do**
- 2:     Infer a Gaussian Process model for  $\gamma_l(t)$  following Section 2.4.1
- 3:     Determine  $t_{\text{GPR},\gamma_l}^*$  by Eq. (23)
- 4: **end for**
- 5: Determine the furthest forecast time permitted for  $\Gamma_m$  as  $t_{\text{GPR},\Gamma_m}^* = \min\{t_{\text{GPR},\gamma_1}^*, t_{\text{GPR},\gamma_2}^*, \dots\}$
- 6: Generate snapshot data for  $\mathbf{x} \in \Omega_m(t)$ ,  $t \in [t_1, t_M]$
- 7: Determine  $R$  dominant POD spatial bases  $\phi_k(\mathbf{x})$ ,  $k = 1, 2, \dots, R$ , with  $RRMS\ Error < \alpha^{\text{POD}}$
- 8: Project  $u(\mathbf{x}, t_i)$  to the reduced space and obtain  $a_k(t) = (\hat{u}(\mathbf{x}, t), \phi_k(\mathbf{x}))$
- 9: Determine the furthest forecast time permitted by the POD,  $t_{\text{POD}}^*$ , from Eq. (20)
- 10: **for**  $k = 1, 2, \dots, R$  **do**
- 11:     Infer a Gaussian Process model for  $a_k(t)$  following Section 2.3.1
- 12:     Determine the furthest forecast time permitted by the GPR,  $t_{\text{GPR},a}^*$ , from Eq. (22)
- 13: **end for**
- 14: Determine the final furthest forecast time as  $t^* = \min\{t_{\text{POD}}^*, t_{\text{GPR},a}^*, t_{\text{GPR},\Gamma_m}^*\}$
- 15: Predict  $\gamma_l(t^*)$  by the inferred Gaussian Process models for  $\gamma_l(t)$
- 16: Predict each  $a_k(t^*)$  by the inferred Gaussian Process models for  $a_k(t)$
- 17: Reconstruct the full-order solution  $u(\mathbf{x}, t^*)$  by Eq. (10)
- 18: Correct  $u(\mathbf{x}, t^*)$  for  $\{\mathbf{x} | \mathbf{x} \in \Omega_f \text{ at } t^* \text{ but } \mathbf{x} \in \Omega_m \text{ during } [t_1, t_M]\}$
- 19: **return**  $u(\mathbf{x}, t^*)$  and  $\gamma_l(t^*)$

---

## 2.5. Error analysis

The error of the proposed nonintrusive model order reduction is originated from three resources: the truncation error in POD, the error caused by GPR, and the interpolation error introduced in Section 2.4.2, i.e.,

$$\epsilon^{\text{ROM}} = \|u(\mathbf{x}, t) - u^{\text{ROM}}(\mathbf{x}, t)\|_{L_2} = \epsilon^{\text{POD}} + \epsilon^{\text{GPR}} + \epsilon^{\text{MLS}}. \quad (26)$$

Here,  $\epsilon^{\text{POD}}$  is the truncation error of POD given by Eq. (7).  $\epsilon^{\text{GPR}}$  is the error caused by GPR and can be indicated by the uncertainty level  $\hat{\sigma}(t)$ :

$$\epsilon^{\text{GPR}} = \|u^{\text{POD}}(\mathbf{x}, t) - u^{\text{ROM}}(\mathbf{x}, t)\|_{L_2} \approx C_1 \hat{\sigma}(t), \quad (27)$$

where  $u^{\text{POD}}(\mathbf{x}, t)$  is the projection of the full-order solution  $u(\mathbf{x}, t)$  onto the reduced space formed by  $R$  POD bases;  $C_1 > 0$  is a constant; and  $\hat{\sigma}(t)$  is given by Eq. (21).  $\epsilon^{\text{MLS}}$  denotes the error of the MLS interpolation and can be estimated by [22]:

$$\epsilon^{\text{MLS}} = \|u(\mathbf{x}, t) - u_h(\mathbf{x}, t)\|_{L_2} \leq C_2 h^{s+1} |u|_{C^{s+1}(\Omega_f)}, \quad (28)$$

where  $C_2 > 0$  is a constant; and  $|u|_{C^{s+1}(\Omega_f)} := \max_{\zeta \leq s+1} \|D^\zeta u\|_{L_\infty}$  with  $D^\zeta u$  the  $\zeta$ -th order spatial derivative of  $u$  for  $\mathbf{x} \in \Omega_f$ . The relative error is then defined as:

$$\epsilon_r^{\text{ROM}} = \frac{\epsilon^{\text{ROM}}}{\|u(\mathbf{x}, t)\|_{L_2}} = \frac{\|u(\mathbf{x}, t) - u^{\text{ROM}}(\mathbf{x}, t)\|_{L_2}}{\|u(\mathbf{x}, t)\|_{L_2}}. \quad (29)$$

## 3. Numerical experiments

We assessed the accuracy and efficiency of the proposed data-driven nonintrusive model order reduction method in several benchmark dynamical systems. We started with three benchmarks without moving boundaries and then moved to problems with moving boundaries. The data used to construct and validate the ROM are either from analytical/numerical solutions or experimental data. The values of variables or parameters are all non-dimensional.

### 3.1. Burgers equation

First, we constructed the ROM to predict the solution of the Burgers equation, which is a typical benchmark used in literature for validating model order reduction methods (e.g., in [23–25]). The Burgers equation considered herein is:

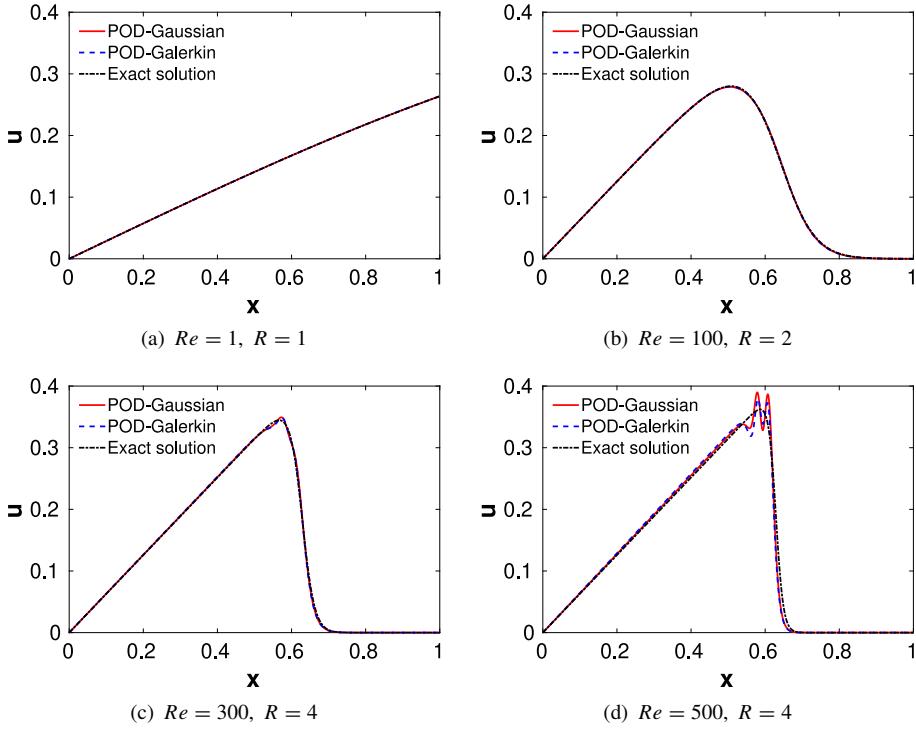
$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} &= \frac{1}{Re} \frac{\partial^2 u}{\partial x^2}, \quad x \in (0, 1) \\ u(0, t) &= 0, \quad u(1, t) = 0, \quad t \geq 0 \\ u(x, 0) &= \frac{x}{1 + \sqrt{\frac{1}{t_0}} \exp(Re \frac{x^2}{4})}, \quad x \in (0, 1) \end{aligned} \quad (30)$$

where  $Re$  is the Reynolds number and  $t_0 = \exp(\frac{Re}{8})$ . The analytical solution of Eq. (30) is given by:

$$u(x, t) = \frac{\frac{x}{t+1}}{1 + \sqrt{\frac{t+1}{t_0}} \exp(Re \frac{x^2}{4t+4})}. \quad (31)$$

The analytical solution (Eq. (31)) of the Burgers equation was used to generate the snapshot data to construct the ROM with the time step  $\delta t = 10^{-2}$  and spatial grid length  $\delta x = 10^{-3}$ . In particular, 20 snapshots from  $t = 0.3$  to  $t = 0.5$  were used to extract the dominant POD modes. We set the truncation threshold  $\alpha^{\text{POD}} = 0.01$ , which means the POD modes retained dominate at least 99.99% of the fluctuating kinetic energy. To satisfy this threshold, the number of POD modes retained ( $R$ ) varies with the Reynolds number, which is indicated in Fig. 3.

To construct the temporal coefficients  $a_k(t)$ , we employed both the Galerkin projection and GPR and compared their performance. With the constructed  $a_k(t)$ , we predicted the solution  $u(\mathbf{x}, t^*)$  of the Burgers equation at a future



**Fig. 3.** Burgers equation: The solution of  $u(x, t)$  at a future time  $t^* = 0.6$  predicted by the ROM built from POD and Gaussian process (POD–Gaussian) or Galerkin projection (POD–Galerkin), compared with the exact solution.

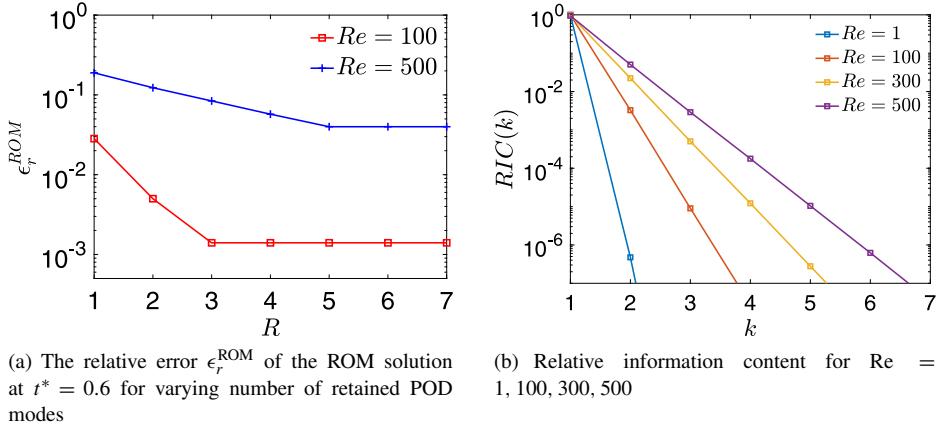
time  $t^* = 0.6$  for different Reynolds numbers from  $Re = 1$  to  $Re = 500$ . The results are shown in Fig. 3. The formulation used in the Galerkin projection to determine  $a_k(t)$  is given as below:

$$\dot{a}_k(t) = \left( \frac{1}{Re} \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}, \phi_k \right) = B_k + \sum_{i=1}^R L_{ik} a_i(t) + \sum_{i=1}^R \sum_{j=1}^R N_{ijk} a_i(t) a_j(t), \quad \text{for } k = 1, 2, \dots, R, \quad (32)$$

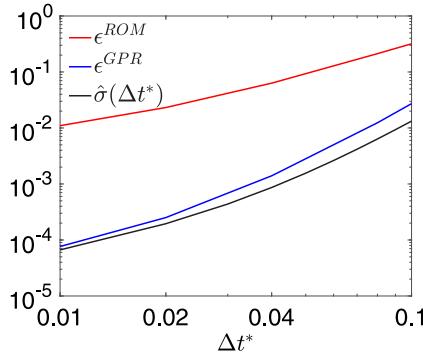
where  $B_k = \left( \frac{1}{Re} \frac{\partial^2 \bar{u}}{\partial x^2} - \bar{u} \frac{\partial \bar{u}}{\partial x}, \phi_k \right)$ ,  $L_{ik} = \left( \frac{1}{Re} \frac{\partial^2 \phi_i}{\partial x^2} - \phi_i \frac{\partial \bar{u}}{\partial x} - \bar{u} \frac{\partial \phi_i}{\partial x}, \phi_k \right)$ , and  $N_{ijk} = \left( -\phi_i \frac{\partial \phi_j}{\partial x}, \phi_k \right)$ .

From Fig. 3, we find that the solution forecasted at the future time  $t^* = 0.6$  by the constructed ROM is accurate for lower Reynolds numbers, either based on the Gaussian process or Galerkin projection. For higher Reynolds numbers, the accuracy deteriorates, and oscillations occur near the shock wave singularity. This is because the dominant spatial bases  $\phi_k(x)$  would change rapidly upon the development of shock wave, which violates the assumption of POD-based model order reduction: the dominant spatial bases are the same for the training data and forecasted solutions. The accuracy of the ROM constructed using the GPR is comparable with that using the Galerkin projection. However, while the Galerkin projection relies on both the data and knowledge of the Burgers equation to establish the ODEs (Eq. (32)) for the temporal coefficients  $a_k(t)$ , the Gaussian process only requires data.

We further examined the behavior of relative error  $\epsilon_r^{\text{ROM}}$  (Eq. (29)) of the ROM constructed using the POD and GPR. The solution predicted at  $t^* = 0.6$  with the snapshots from  $t_1 = 0.3$  to  $t_M = 0.5$  was considered. As expected, the constructed ROM is more accurate with more POD modes retained, as depicted in Fig. 4(a) for  $Re = 100$  and  $Re = 500$ . However, the error decay stalls when  $R$  (the number of POD modes retained) reaches a certain number, which is because  $\epsilon^{\text{GPR}}$  begins to dominate when  $\epsilon^{\text{POD}}$  becomes small. In addition, the error decreases with respect to  $R$  at a faster rate for lower Reynolds number ( $Re = 100$ ) and at a slower rate for higher Reynolds number ( $Re = 500$ ). This can be explained through the analysis of eigenvalues. Recall the truncation error of POD is determined from the eigenvalues of the correlation matrix of the snapshot data, as specified in Eq. (7). The energy



**Fig. 4.** Burgers equation: Error analysis for different number of POD modes retained.



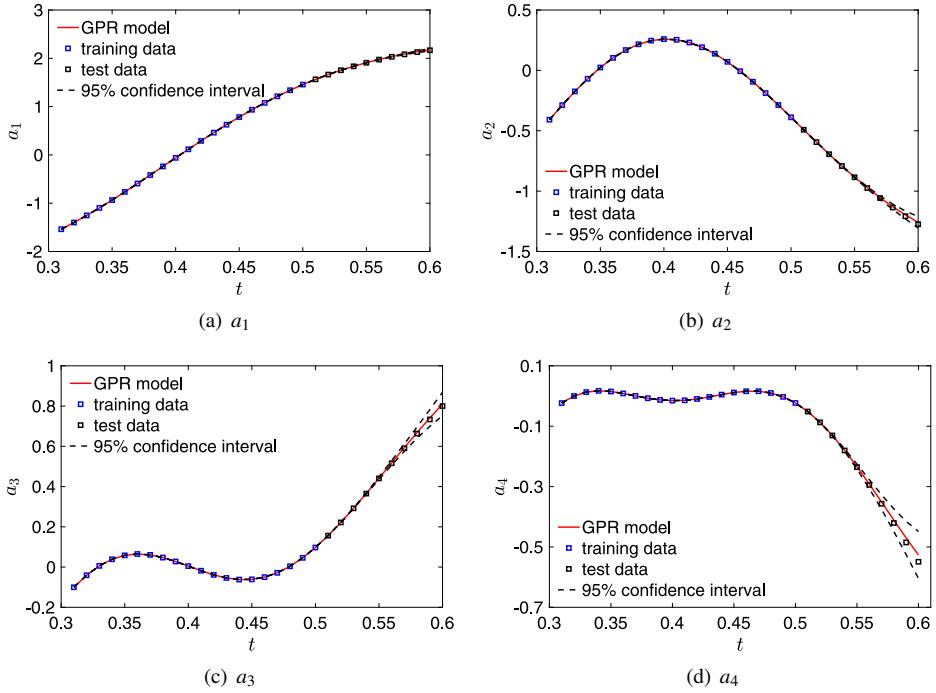
**Fig. 5.** Burgers equation:  $\epsilon^{\text{ROM}}$  at different forecast times beyond the dataset range  $\Delta t^* = t^* - t_M$ . Here,  $\epsilon^{\text{GPR}}$  and  $\hat{\sigma}$  are also shown for comparison.

contributed by each mode in the total energy can be evaluated by the relative information content (RIC), expressed as:

$$RIC(k) = \left( \frac{\lambda_k}{\sum_{j=1}^M \lambda_j} \right) \times 100\%. \quad (33)$$

In Fig. 4(b), we compare the RIC of  $k$ th POD mode for different Reynolds numbers. The RIC demonstrates how the contribution of each POD mode decays. The smaller the Reynolds number is, the decay slope is more steep, and thereby the POD truncation error decreases faster with respect to  $R$ . The decay rate of eigenvalues, as defined in Eq. (20) is also used to determine the furthest forecast time. Fig. 4(b) indicates smaller Reynolds number permits longer time extrapolation (i.e., larger  $\Delta t^* = t^* - t_M$ ) given the dataset range  $[t_1, t_M]$  is fixed.

In Fig. 5, we assessed the errors  $\epsilon^{\text{ROM}}$  and  $\epsilon^{\text{GPR}}$  and the uncertainty level of GPR  $\hat{\sigma}$  at different forecast times  $t^*$  with the same snapshots from  $t_1 = 0.3$  to  $t_M = 0.5$ . Here,  $Re = 500$  was considered and 4 POD modes were retained. The growth of  $\hat{\sigma}$  as  $\Delta t^*$  increases can be explained by Fig. 6, where the GPR model trained for each dominant  $a_k(t)$  is presented. It can be seen that the 95% confidence interval enlarges, indicating the uncertainty level (or  $\hat{\sigma}(t^*)$ ) increases, as time extrapolation goes further. This result also supports the use of  $\hat{\sigma}(t^*)$  as a criterion to adaptively determine the furthest forecast time  $t^*$ , as given in Eq. (22).  $\epsilon^{\text{GPR}}$  varies following the trend of  $\hat{\sigma}(\Delta t^*)$ . Finally,  $\epsilon^{\text{ROM}}$  grows with  $\Delta t^*$ , i.e., with the forecast time  $t^*$  going further beyond the dataset range, resulting from the behavior of  $\epsilon^{\text{GPR}}$ .



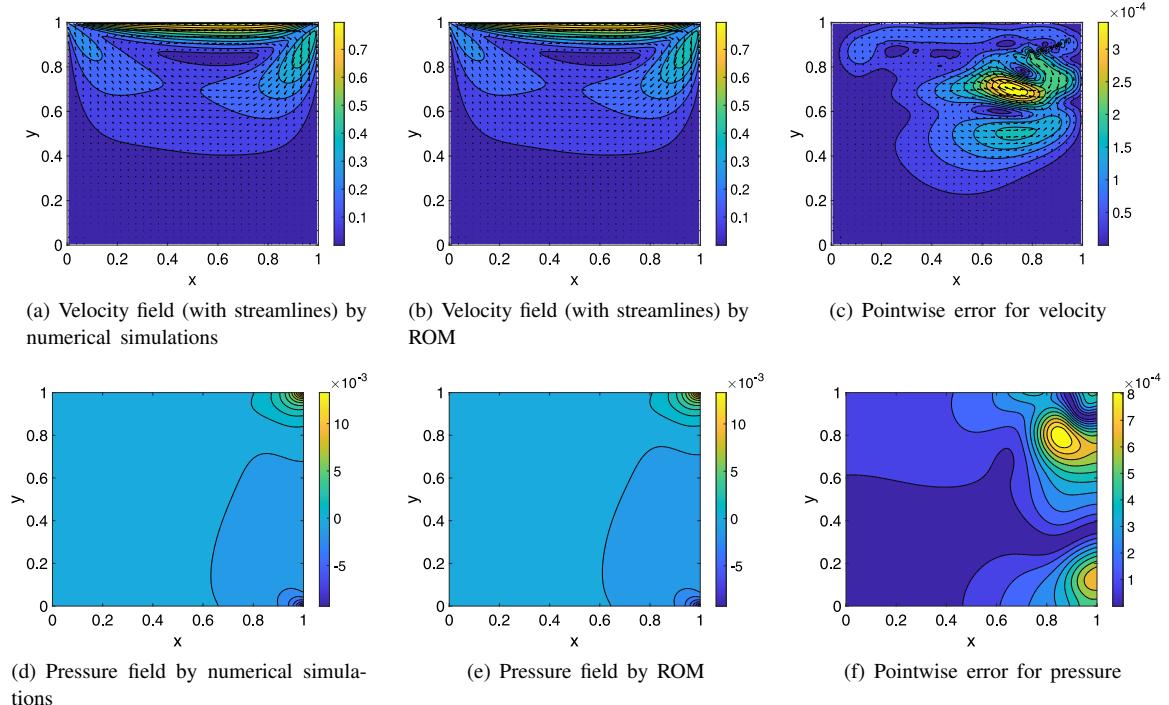
**Fig. 6.** Burgers equation: GPR for each dominant temporal coefficient  $a_k(t)$ .

### 3.2. Lid-driven cavity flow

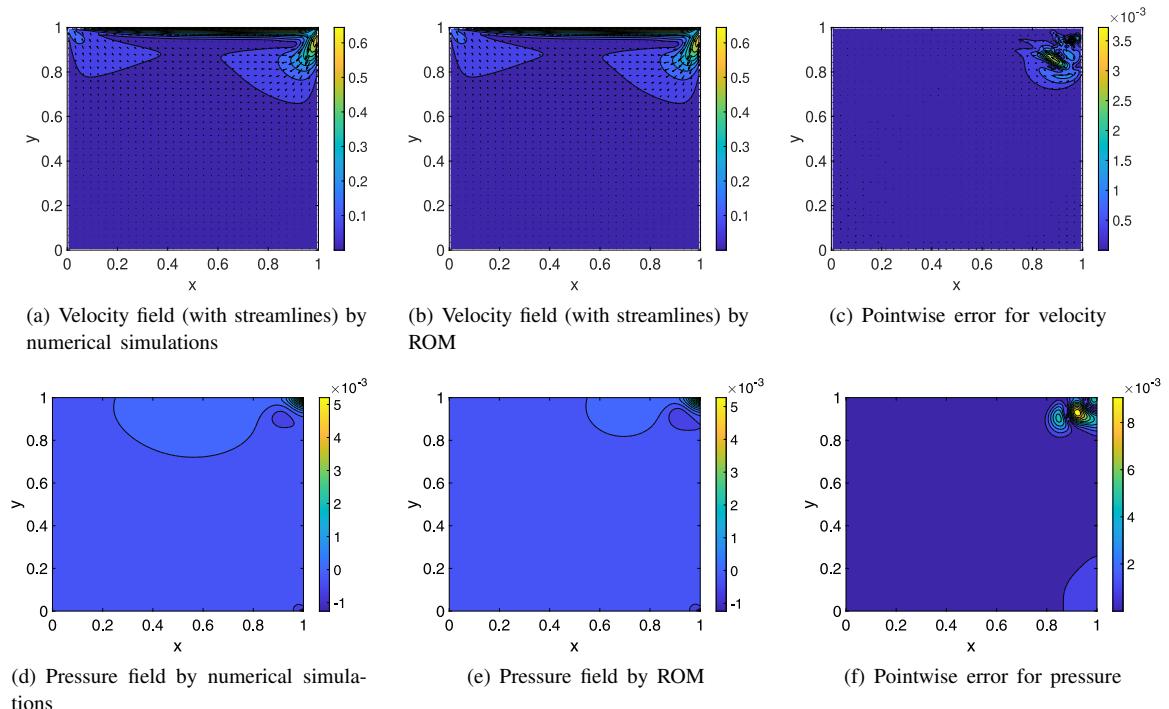
In this section, we studied the 2D lid-driven cavity flow, where a square cavity  $\Omega = [0, 1] \times [0, 1]$  consists of three stationary walls and a lid moving with a constant tangential velocity. To obtain the velocity and pressure fields for the lid-driven cavity flow, numerical simulations are usually employed for solving the full Navier–Stokes (NS) equation. Here, we assessed the ability of the proposed Gaussian process-enabled nonintrusive model order reduction to predict the velocity and pressure fields for this flow. To construct the ROM, the snapshot data were generated via simulation which is based on the staggered finite-volume spatial discretization, projection method to enforce incompressibility, and treating the nonlinear advection term explicitly and the viscous term implicitly [26,27].

Two different Reynolds numbers were considered:  $Re = 100$  and  $Re = 1000$ . In numerical simulations, the time step was set as  $\delta t = 0.01$ , and  $100 \times 100$  grids were used in the spatial discretization. We took the numerical solutions between  $t = 0.6 \sim 0.8$  as the snapshot data to construct the ROM using the POD and GPR. The POD truncation threshold was set as  $\alpha^{\text{POD}} = 0.01$ . The constructed ROM was then employed to predict the velocity and pressure fields of the flow for  $t > 0.8$ . Figs. 7 and 8 present the predicted solutions at  $t^* = 1.0$  for  $Re = 100$  and  $Re = 1000$ , respectively. This forecast time  $t^*$  was determined from Eqs. (20) and (22) with  $\beta^{\text{POD}} = 0.3$  and  $\beta^{\text{GPR},a} = 0.01$ . By comparison with the full-order solutions by simulations, we find the prediction of the constructed ROM is very accurate. The relative errors  $\epsilon_r^{\text{ROM}}$  for both velocity and pressure are less than 2% for either  $Re = 100$  or  $Re = 1000$ .

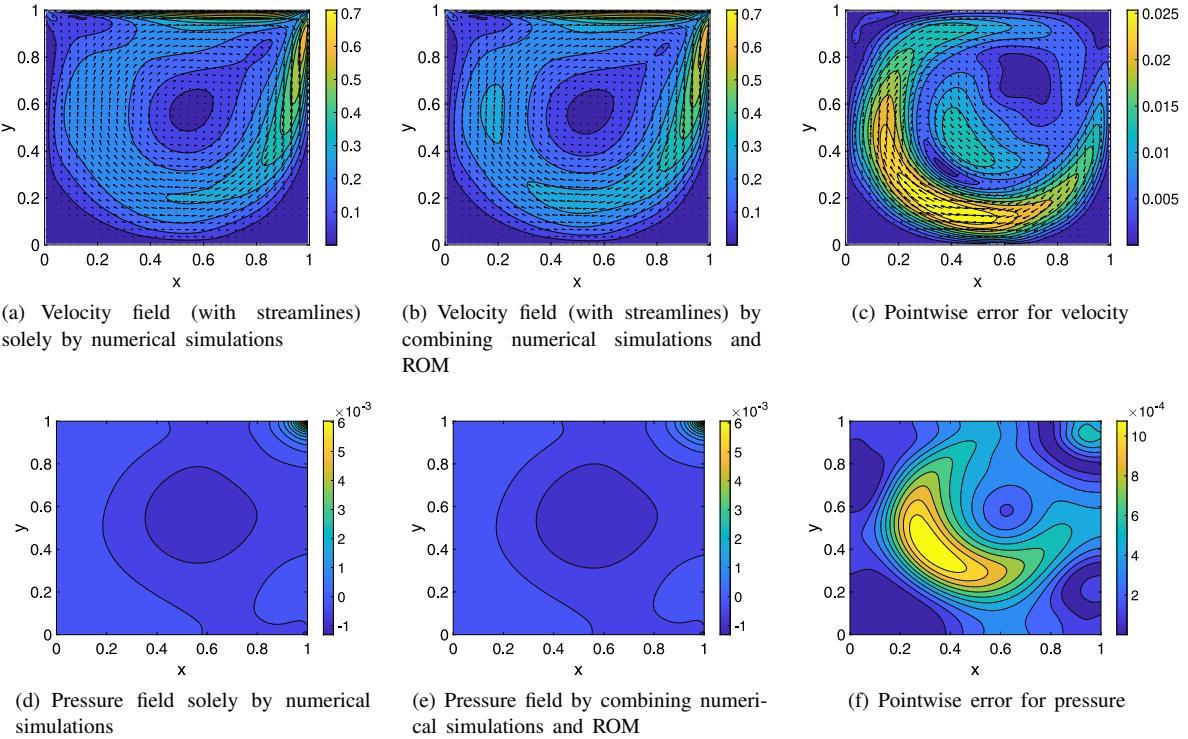
Next, we examined our implementation of the automated process that adaptively combines the numerical simulations and ROM (as described in Section 2.3.3) for long-time prediction of the solutions for the lid-driven cavity flow. Fig. 9 shows the velocity and pressure fields predicted at  $t = 20.0$  for  $Re = 1000$  by combining the numerical simulations and ROM, compared with the solutions obtained solely from numerical simulations. Good accuracy is achieved with the relative errors  $\epsilon_r^{\text{ROM}}$  for both velocity and pressure less than 5%.



**Fig. 7.** Lid-driven cavity flow: Comparison of the solutions predicted by the ROM with the solutions obtained from numerical simulations at  $t = 1.0$  for  $Re = 100$ . Here,  $R = 2$ , i.e., the first 2 POD modes are retained.



**Fig. 8.** Lid-driven cavity flow: Comparison of the solutions predicted by the ROM with the solutions obtained from numerical simulations at  $t = 1.0$  for  $Re = 1000$ . Here,  $R = 3$ , i.e., the first 3 POD modes are retained.



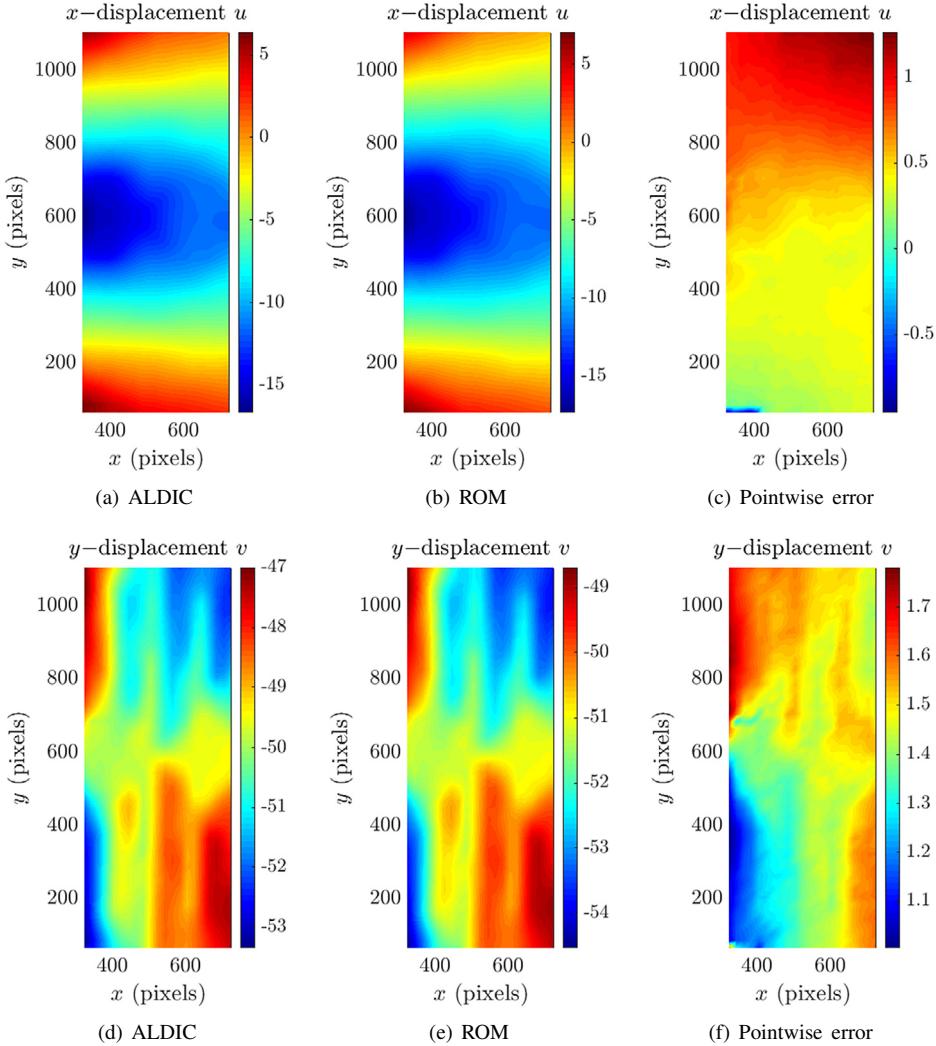
**Fig. 9.** Lid-driven cavity flow: Comparison of the solutions predicted by adaptively combining the numerical simulations and ROM with the solutions obtained solely from numerical simulations at  $t = 20.0$  for  $Re = 1000$ .

### 3.3. Heterogeneous fracture deformation

In this section, we applied the proposed nonintrusive model order reduction method to experimental data, where the full-order model is unknown. The experimental data are the 2D displacement fields for heterogeneous fracture deformation. From the snapshots of the displacement fields at different times, we constructed the ROM using the POD and GPR. The constructed ROM was then employed to forecast the displacement fields of future times beyond the snapshot data.

In experiments, the data of displacement fields were obtained by employing the augmented-Lagrangian digital image correlation (ALDIC) method [28] on the images from the experiment of Avellar [29]. The images recorded the gray scale of a speckle pattern painted on the surface of a material specimen. Upon deformation, the gray scale of the speckle pattern changed. Hence, by comparison of the gray scale in the images before and after deformation, the displacement field can be determined by solving an optimization problem. Although effective, the ALDIC method is quite expensive and requires storage of massive image data. Instead, the ROM, once constructed, can efficiently reproduce the displacement field at any given time until the furthest forecast time  $t^*$  allowed by the ROM is reached. To demonstrate the proposed reduced order modeling, we chose to consider the data for a material specimen with heterogeneous stiffness and thereby complex displacement fields [28], for which the saving of computational cost by using the ROM compared with the ALDIC method is even more remarkable.

More specifically, we used 15 snapshots of the displacement fields from  $t_1 = 0$  to  $t_M = 15$  to construct the ROM, which was then employed to predict the displacement field at  $t^* = 25$ . Here, the first 3 POD modes were retained ( $R = 3$ ) with the POD truncation threshold  $\alpha^{\text{POD}} = 0.01$ . The furthest forecast time  $t^*$  was determined from Eqs. (20) and (22) with  $\beta^{\text{POD}} = 0.3$  and  $\beta^{\text{GPR},a} = 0.1$ . The results are presented in Fig. 10. For comparison, we also show the displacement fields determined using the ALDIC method. From the comparison, we demonstrate the good accuracy of the predictions by the ROM. Taking the results by the ALDIC method as the “exact” solutions,



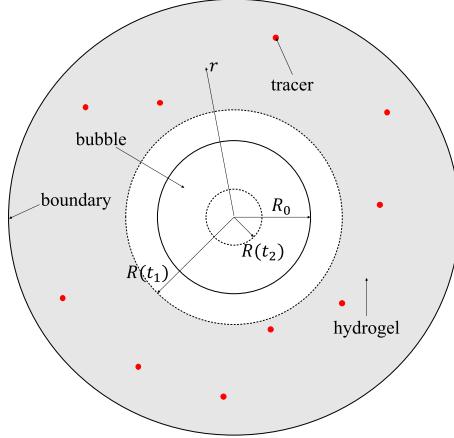
**Fig. 10.** Heterogeneous fracture deformation: The ((a)–(c))  $x$ - and ((d)–(f))  $y$ -displacement fields predicted by the ROM at  $t^* = 25$ , compared with the results by the ALDIC method.

the relative error  $\epsilon_r^{\text{ROM}}$  is less than 8% for the prediction of  $x$ -displacement field and less than 2% for the prediction of  $y$ -displacement field.

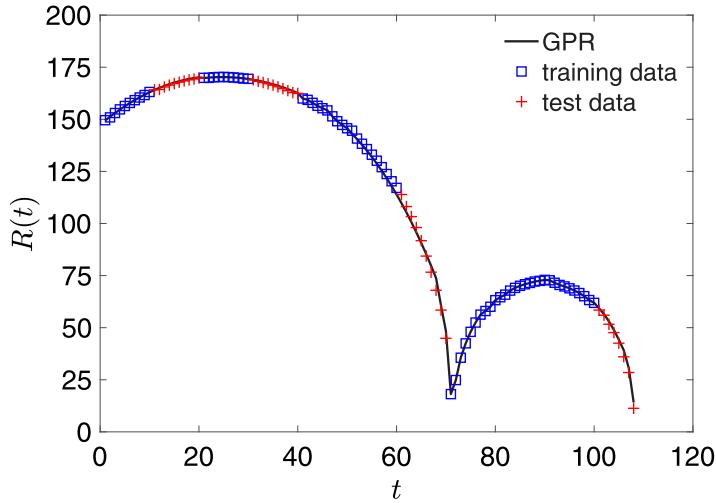
In regard to the computational cost, it typically takes the ALDIC method about 53 s to generate the displacement field from two images [28], which was tested in MATLAB using a workstation with Intel (R) Xeon(R) CPU E5-2650 v3 2.30 GHz (2 Processors). It only takes the ROM about 0.02s to predict the displacement field using MATLAB and comparable hardware. And the cost for constructing the ROM is about 0.70s. Thus, integrating the ALDIC method with the proposed reduced order modeling can significantly reduce the computational cost in experiments to determine the time-varying displacement fields.

### 3.4. Bubble cavitation in hydrogel

After the three benchmarks used to validate the nonintrusive model order reduction method based on the POD and GPR, we next considered a problem involving moving boundary. In particular, it concerned bubble cavitation in hydrogel. The data used to construct and validate the ROM were from the experiments [30], where a spherical



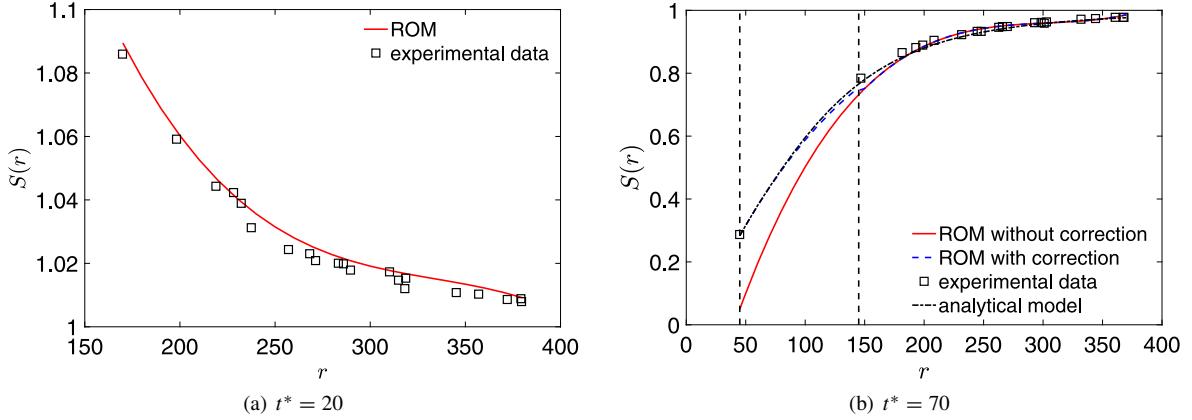
**Fig. 11.** Schematic of a bubble cavitation in hydrogel. The bubble can expand (e.g. at  $t_1$ ) or shrink (e.g., at  $t_2$ ).



**Fig. 12.** Bubble cavitation in hydrogel: The prediction of the bubble cavitation's radius  $R(t)$ .

bubble cavitation was formed in the center of a hydrogel material. Assume the initial radius of the bubble is  $R_0$ . Due to the plasma recombination and the volume change of the vapor and non-condensable gas within the bubble, the radius of the bubble is a function of time  $R(t)$ ; as a result, the interface between the bubble and hydrogel is a moving boundary. The expansion or shrinkage of the bubble led to compression or tension on the surrounding hydrogel material. A schematic of this problem is illustrated in Fig. 11. The distribution of the elastic strain  $S$  in the surrounding hydrogel as well as the bubble's radius  $R(t)$  were measured in the experiments. The elastic strain  $S$  was measured at each tracer position. Due to the spherical symmetry of this problem, we simplified it to a one-dimensional problem. Hence, the strain field only depends on  $r$ , the radial distance to the center of the bubble.

From  $M$  snapshots of the experimental data, we first constructed a GPR model for  $R(t)$ , the bubble's radius as a function of time. The constructed GPR model can forecast the bubble's radius at a given time beyond the range of data. The furthest forecast time  $t^*$  was determined from Eqs. (20), (22) and (23) with  $\beta^{\text{POD}} = 0.3$ ,  $\beta^{\text{GPR},a} = 0.1$  and  $\beta^{\text{GPR},\Gamma_m} = 0.1$ . After the furthest forecast time  $t^*$  was reached, new  $M$  snapshots of experimental data were collected and used to infer another GPR model to forecast  $R(t)$  for further times. Repeating this process and the resulting adaptive combination of experimental measurements and GPR modeling allowed us to efficiently predict the long-time evolution of the radius of the bubble cavitation with a complex dynamics of expansion and shrinkage, as shown in Fig. 12. Good agreement was achieved between the GPR model's predictions and the test data from experiments.



**Fig. 13.** Bubble cavitation in hydrogel: The prediction of the elastic strain  $S(r, t)$  in hydrogel.

Besides the bubble cavitation's radius, we also predicted the elastic strain field  $S(r, t)$  in the surrounding hydrogel. To this end, the ROM for  $S(r, t)$  was constructed using the POD and GPR from the experimental snapshot data. For demonstration, we predicted for two different times: one during the expansion of the bubble and the other during the bubble's shrinkage. More specifically, we used 10 snapshot data from  $t_1 = 1$  to  $t_M = 10$  to construct a ROM, which was then employed to forecast  $S(r, t)$  at  $t^* = 20$ ; and we used 10 snapshot data from  $t_1 = 51$  to  $t_M = 60$  to construct another ROM to predict  $S(r, t)$  at  $t^* = 70$ . Each prediction of  $S(r, t)$  by the constructed ROM, along with the comparison with the experimental data, is presented in Fig. 13. After  $t > 50$ , the bubble cavitation experienced shrinkage until  $t = 70$ . During this period of time, the surrounding hydrogel was stretched toward the bubble's center. Hence, the strain field  $S(r, t > t_M)$  included the range of  $r$  that was not covered in the snapshot data (from  $t_1 = 51$  to  $t_M = 60$ ). To construct the ROM covering that range of  $r$ , we extrapolated by least squares the snapshot data of  $S(r)$  until the minimum possible value of  $R$ . When we forecasted  $S(r, t)$  at  $t^* = 70$  by the constructed ROM, for the region of  $r$  not covered in the snapshot data (i.e., the region between two vertical dash lines in Fig. 13(b)), the correction was made using the method proposed in Section 2.4.2. To validate the prediction of  $S(r)$  in that region, since no experimental data were available, we relied on the following analytical model [30,31] for validation:

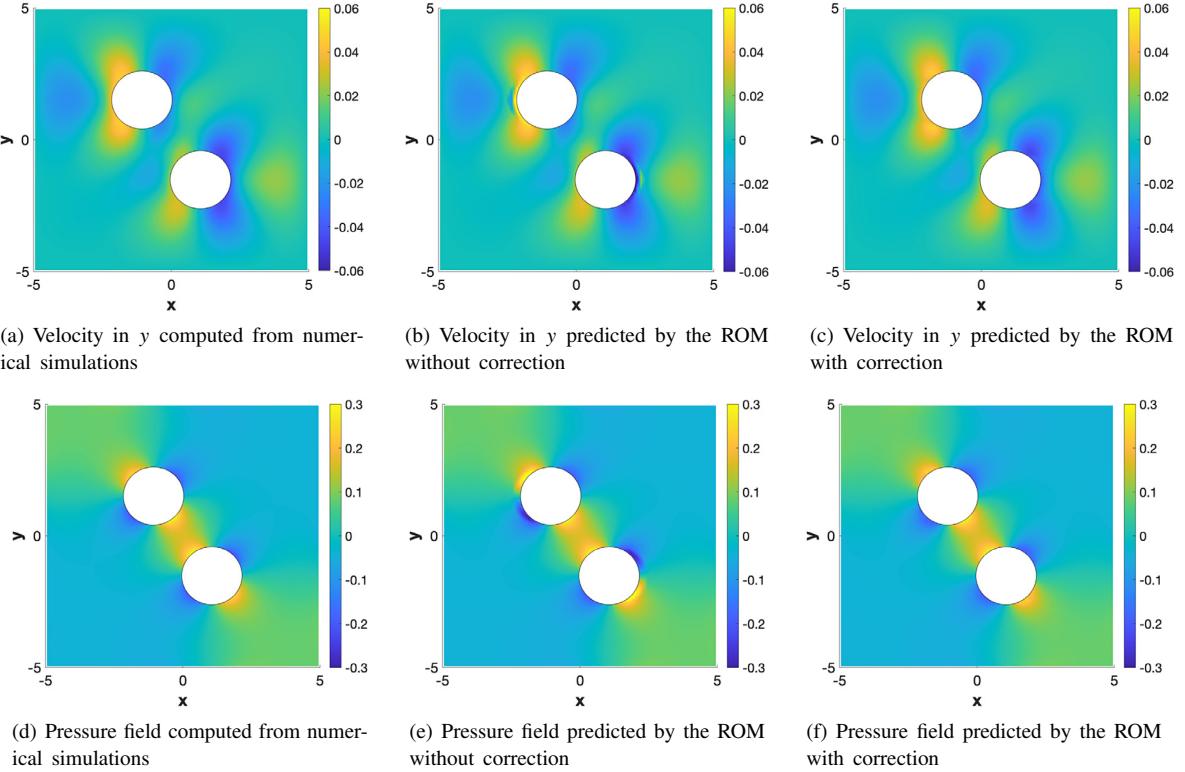
$$S(r, t) = \left( \frac{r}{r^3 + R(\bar{t})^3 - R(t)^3} \right), \quad (34)$$

where  $\bar{t} = 5$ . As can be seen in Fig. 13, the predictions of the strain field  $S(r, t)$  by the ROM reasonably agree with the experimental data and the analytical model. Thus, we anticipate the proposed reduced order modeling can be applied to predict the strain field for the regions and times that are not accessible in experiments or when the analytical model is not applicable (e.g., after the first collapse of the bubble).

### 3.5. Fluid–solid interactions

Finally, we studied the problems of fluid–solid interactions, where multiple solid bodies move in a fluid flow. Assume the domain  $\Omega_m$  consists of  $N_s$  solid bodies, and each of them has a boundary  $\Gamma_m^n$ ,  $n = 1, \dots, N_s$ , and a center-of-mass position  $X_n$  and orientation  $\Theta_n$ . In this study, we predicted the dynamics of each solid body as well as the velocity and pressure fields in the fluid by the ROM constructed. The snapshot data used to construct the ROM were generated via numerical simulations, where the incompressible Stokes flow subject to moving solid boundaries was numerically solved using the generalized moving least squares discretization method [32]. For simplicity, the solid bodies are subject to rigid-body kinematics. The evolution of a moving solid boundary  $\Gamma_m^n$  can be characterized by  $[X_n, \Theta_n]$ .

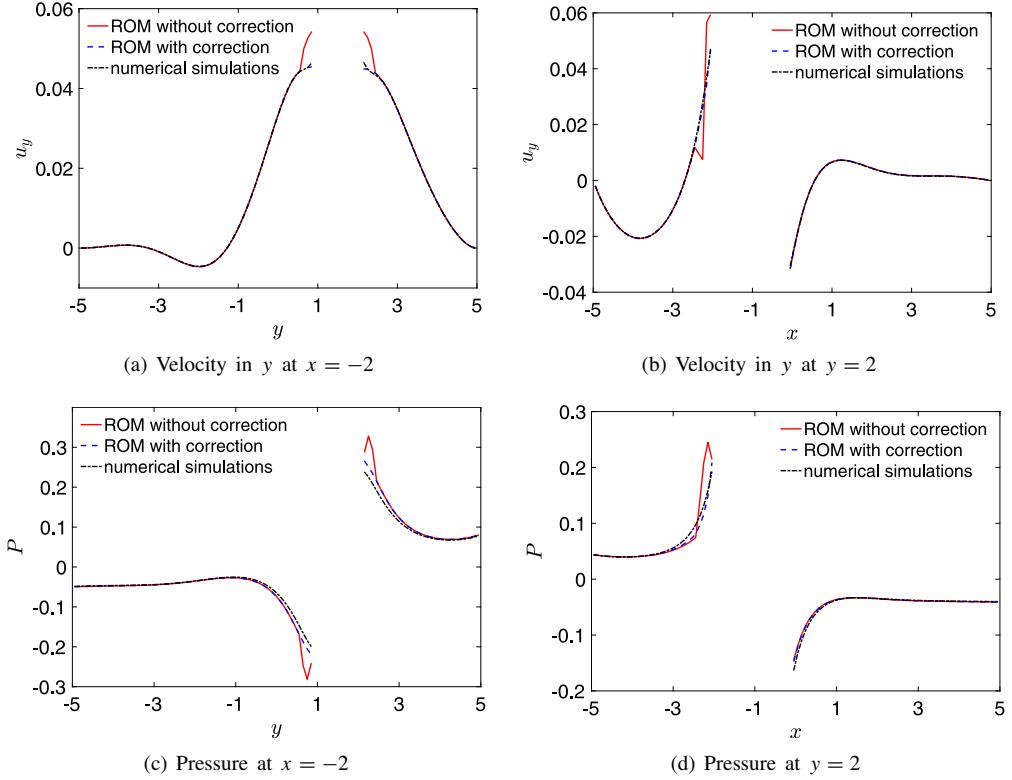
In all the cases studied herein, the fluid and solid bodies are confined in a 2D square box of  $[-5, 5] \times [-5, 5]$ . The top and bottom boundaries of the box are subject to identical velocities  $u_0 = 0.5$  but along opposite  $x$  directions so



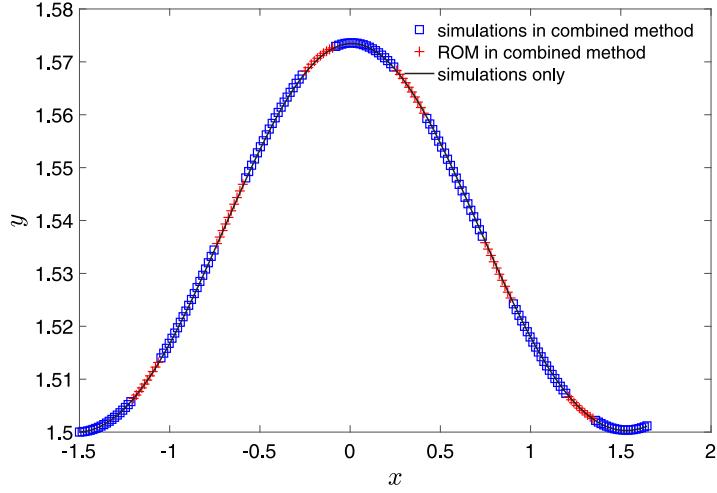
**Fig. 14.** Two cylinders under a shear flow: The velocity and pressure fields in the fluid predicted by the ROM at  $t^* = 3.0$ , compared with the full-order solutions by numerical simulations.

as to generate a shear flow. In numerical simulations, the time step was set  $\delta t = 0.1$ , and the spatial discretization was uniform with the spacing  $\delta x = 0.1$ . In the first case, there are two cylinders of equal size immersed in the fluid flow. We constructed the ROM from the first 20 snapshots from  $t_1 = 0$  to  $t_M = 2.0$ . The constructed ROM was then used to predict the velocity and pressure fields in the fluid and the positions of the two cylinders at  $t^* = 3.0$ , which is the furthest forecast time determined from Eqs. (20), (22) and (23) with  $\beta^{\text{POD}} = 0.8$ ,  $\beta^{\text{GPR},a} = 0.1$  and  $\beta^{\text{GPR},\Gamma_m} = 0.1$ . The first 4 POD modes were retained in the ROM by setting  $\alpha^{\text{POD}} = 0.05$ . Fig. 14 presents the velocity and pressure fields predicted by the ROM, compared with the full-order solutions by numerical simulations. Since the velocity in  $x$  direction is dominated by linear shear flow, the comparison is made for the velocity along  $y$  direction. We also compared the predictions with and without the correction near the moving solid boundaries, as discussed in Section 2.4.2. For a closer view, Fig. 15 depicts the velocity and pressure along two lines at  $x = -2$  and  $y = 2$ , respectively. It is clearly seen that the large errors appear in the regions near the moving solid boundaries on the downstream sides, which were previously occupied by the cylinders ( $\Omega_m$ ) but not part of the fluid domain ( $\Omega_f$ ). The correction method proposed in Section 2.4.2 can effectively improve the accuracy of the ROM's predictions for those regions.

For the problems of fluid–solid interactions, the numerical simulations can be rather demanding and expensive. Thus, we can adaptively combine the numerical simulations and ROM to save computational cost. In this case, we took the ROM's predictions (the velocity and pressure fields in the fluid and the positions of cylinders) at  $t^* = 3.0$  as the initial condition and restarted the numerical simulations until  $t = 5.0$ . Using the numerical solutions for  $t_1 = 3.1$  until  $t_M = 5.0$  as the snapshot data, we constructed a new ROM and predicted the solutions for  $t^* = 6.0$ . By repeating this procedure until the target time  $t = 20.0$ , we predicted the entire trajectory of one cylinder, as plotted in Fig. 16. By comparison with the result obtained solely from the numerical simulations, we find the prediction by adaptively combining the numerical simulations and ROM achieves good accuracy. In the meanwhile, the ROM's predictions replaced about  $\frac{1}{3}$  of the simulation. Compared with the cost of the numerical simulations, the

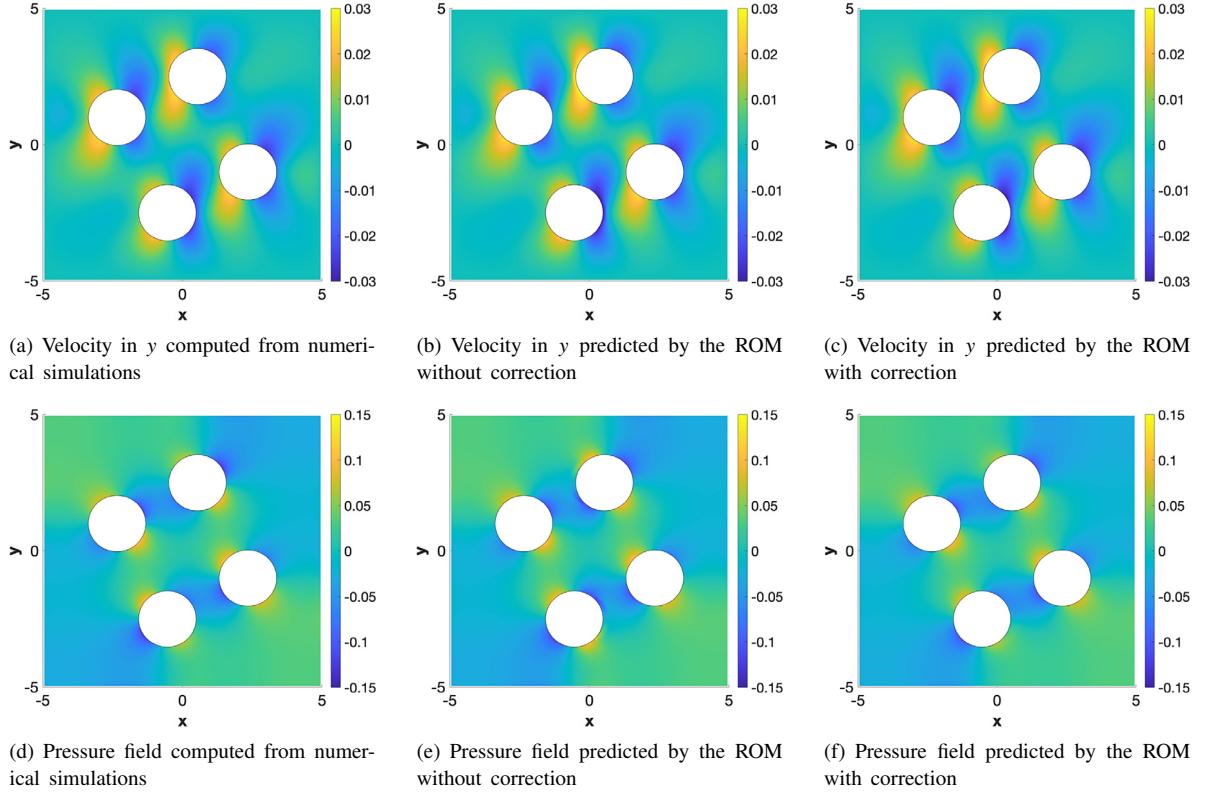


**Fig. 15.** Two cylinders under a shear flow: The velocity and pressure in the fluid along two lines  $x = -2$  and  $y = 2$  at  $t^* = 3.0$  predicted by the ROM with and without correction, compared with the full-order solutions by numerical simulations.



**Fig. 16.** Two cylinders under a shear flow: The trajectory of one cylinder predicted by the combination of the numerical simulations and ROM.

cost of constructing the ROM and employing it for prediction is trivial. For example, it took 21.2s for the numerical simulations to march from  $t = 2.0$  to  $t = 3.0$  with the time step  $\delta t = 0.1$ . The simulation was run using C++ with OpenMPI-based parallel implementation on one Intel(R) Xeon(R) E3-1231 v3 CPU @ 3.4 GHz with 4 cores. On



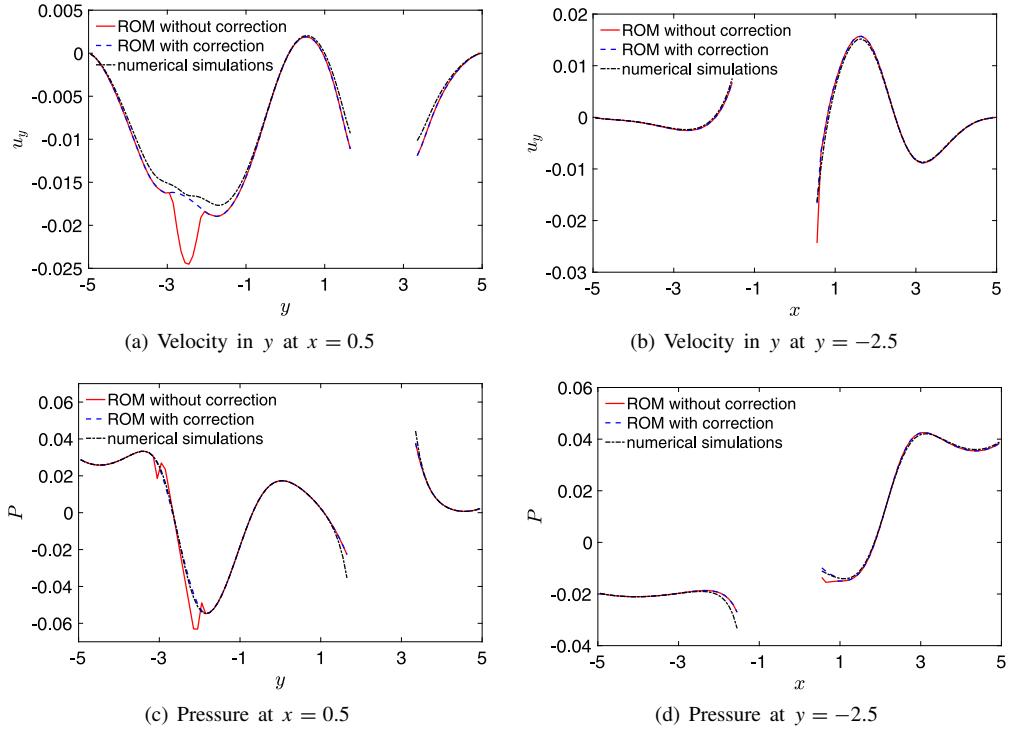
**Fig. 17.** Four cylinders under a shear flow: The velocity and pressure fields in the fluid predicted by the ROM at  $t^* = 3.0$ , compared with the full-order solutions by numerical simulations.

the same hardware, it only needed 4.65 s in total to construct the ROM and employ it for prediction by a serial MATLAB code.

To demonstrate the robustness of the proposed reduced order modeling method, we further examined a case of more cylinders in a shear flow. Figs. 17 and 18 summarize the velocity in  $y$  direction and pressure predicted by the ROM with and without the correction near the moving solid boundaries, compared with the full-order solutions by numerical simulations. Finally, to demonstrate the applicability of the proposed reduced order modeling method to moving boundaries of arbitrary geometries, we also solved the fluid–solid interaction problem with two squares. The results are presented in Figs. 19 and 20. In either case, the predictions by the ROM with the correction achieve good accuracy. Figs. 18 and 20 provide closer views for the difference between without and with the correction. We hence have demonstrated that the proposed reduced order modeling method is applicable to dynamical systems with general moving boundaries regardless of the number or geometry of the boundaries.

#### 4. Conclusion

We have presented a model order reduction method for dynamical systems with moving boundaries, which draws on the POD, GPR, and MLS interpolation. The method is nonintrusive and applicable to experimental data. Given a set of snapshot data of state variables at discrete time instances, the reduced space is constructed via the POD. The time-evolution of the temporal POD coefficients and the parameters characterizing the moving boundaries is inferred from the data via the GPR. The cost of GPR would not increase due to the nonlinearity of the system. The errors in the ROM’s prediction for the regions near the moving boundaries on the downstream side can be effectively reduced using the correction method based on the MLS interpolation. For a given set of snapshot data, the ROM constructed from the data can predict the full-order solution at a desired time inside or outside the dataset range. The forecast

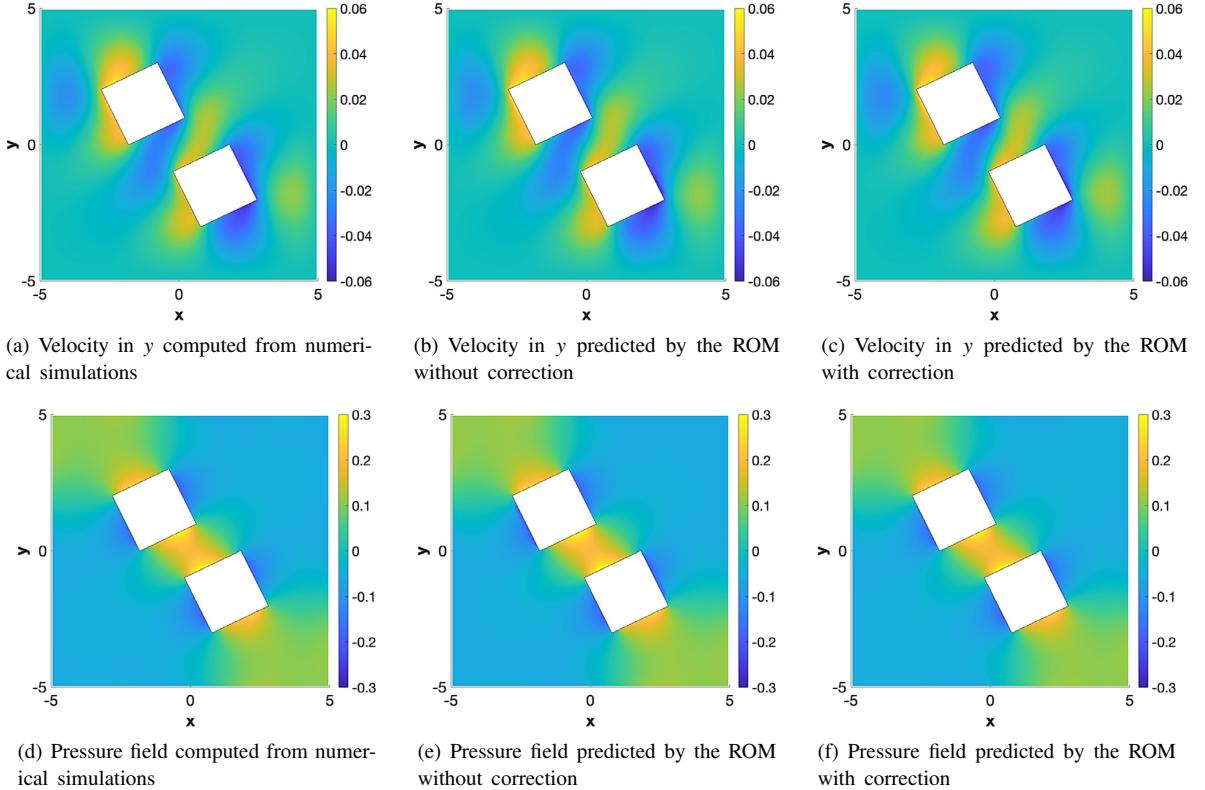


**Fig. 18.** Four cylinders under a shear flow: The velocity and pressure along the lines  $x = 0.5$  and  $y = -2.5$  at  $t^* = 3.0$  predicted by the ROM with and without correction, compared with the full-order solutions by numerical simulations.

beyond the range of snapshot data is constrained by the POD and GPR as well as how many snapshot data are available. To avoid a trial-and-error approach, we have provided the criteria for *a priori* determination of the furthest forecast time permitted in time extrapolation of the ROM. We have demonstrated the accuracy and efficiency of the proposed method in several benchmark problems, where the snapshot data used to construct and validate the ROMs were generated from analytical solutions, results of numerical simulations, and experimental data.

When numerical simulations or experimental measurements are demanding or expensive, reduced order modeling provides an attractive alternative means to predict the full-order solutions. In practice, numerical simulations/experimental measurements and ROMs can be adaptively combined to achieve an efficient long-time prediction for a dynamical system. With the furthest forecast time of the ROM determined *a priori* by the proposed criteria, the numerical simulations/experimental measurements and ROMs can be alternatively called in an automated process. We note that the “equation-free” approach [33,34] proposed by Kevrekidis and collaborators for multiscale modeling also couples two levels of predictions “on-the-fly”: microscopic simulations and macroscopic models. In particular, it extrapolates ensemble-averaged macroscale quantities obtained from the microscopic simulations. The extrapolation is through a projective integrator that advances the macro variable over a macro time step with the time derivative of the macro variable computed from the results of the last few steps of the microscale simulations using small time steps. The “equation-free” approach requires time-scale separation, i.e., the local relaxation time for the microscopic process is much smaller than the time scale for the macroscopic evolution of the system. The approach proposed in the present work uses Gaussian process regression for time extrapolation and does not require time-scale separation.

The slowly decaying Kolmogorov  $n$ -width is a major factor limiting the furthest forecast time of the ROM, which results from the fact that POD restricts the state to evolve in a linear subspace. To address this  $n$ -width limitation of linear subspaces, Lee and Carlberg proposed to construct ROMs in nonlinear manifolds that are computed based on convolutional autoencoders from deep learning [35]. The ROM constructed by this method was shown significantly outperforming the linear-subspace ROM when forecast the full-order solution at a future time for advection-dominated Burgers equation. Thus, it may be worthwhile in a future work to consider replacing the linear spatial bases by nonlinear manifolds in our method to achieve a longer forecast time for the ROM.



**Fig. 19.** Two squares under a shear flow: The velocity and pressure fields in the fluid predicted by the ROM at  $t^* = 3.0$ , compared with the full-order solutions by numerical simulations.

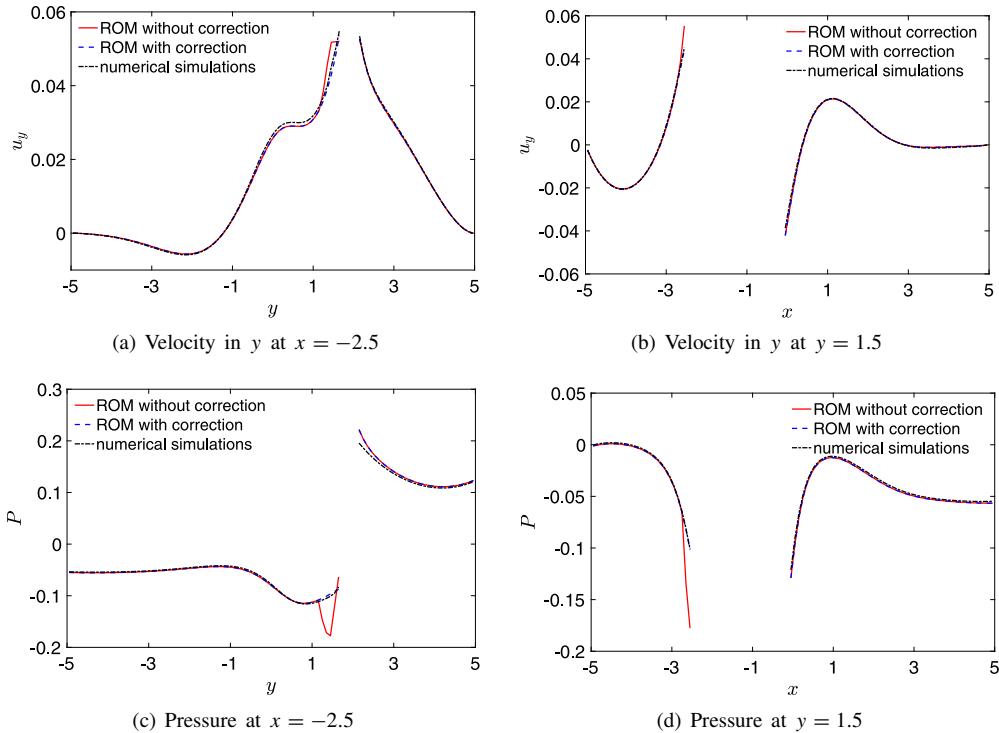
Our method can be potentially extended to construct ROMs to predict the full-order solutions for different parameters/inputs. Suppose  $u(\mathbf{x}, t; \boldsymbol{\eta})$  is the full-order solution for a parameterized dynamical system, where  $\boldsymbol{\eta} \in \mathcal{P}$  denotes the parameter, and  $\mathcal{P} \subset \mathbb{R}^d$  is the parameter space. The full-order solutions  $u(\mathbf{x}, t_i; \boldsymbol{\eta}_j)$  at sampled parameters  $\boldsymbol{\eta}_j \in \mathcal{P}_S \subset \mathcal{P}$  ( $j = 1, 2, \dots, M_\eta$ ) and time instances  $t_i$  ( $i = 1, 2, \dots, M_t$ ) are available to form the snapshot dataset. We aim to predict the solution  $u(\mathbf{x}, t'; \boldsymbol{\eta}^*)$  at  $t'$  for a new parameter  $\boldsymbol{\eta}^* \notin \mathcal{P}_S$ . For this type of problems, the coefficients of POD modes become  $a_k(t, \boldsymbol{\eta}) \in \mathbb{R}$ . Thus, a multivariate Gaussian process model for each  $a_k(t, \boldsymbol{\eta})$  needs to be constructed from the snapshot data with the training input  $\mathbf{z} = (t, \boldsymbol{\eta}) \in \mathbb{R}^{d+1}$ . The multivariate Gaussian process model constructed is then used to predict the POD coefficient  $a_k(t', \boldsymbol{\eta}^*)$  at  $t'$  and  $\boldsymbol{\eta}^*$ . Finally, the full-order solution can be predicted by:  $u(\mathbf{x}, t'; \boldsymbol{\eta}^*) \approx \bar{u}(\mathbf{x}) + \sum_{k=1}^R a_k(t', \boldsymbol{\eta}^*) \phi_k(\mathbf{x})$  with  $\bar{u}(\mathbf{x}) = \frac{1}{M_t M_\eta} \sum_{j=1}^{M_\eta} \sum_{i=1}^{M_t} u(\mathbf{x}, t_i; \boldsymbol{\eta}_j)$ . Detailed error analysis, criteria, and application to parameterized dynamical systems with moving boundaries will be investigated in our future work.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1761068. We gratefully acknowledge Dr. Christian Franck and Dr. Jin Yang in the Department of Mechanical Engineering at the University of Wisconsin-Madison for insightful discussions and providing us the experimental data in Sections 3.3 and 3.4. We also thank the two anonymous reviewers for their insightful comments and suggestions that helped improve the manuscript.



**Fig. 20.** Two squares under a shear flow: The velocity and pressure along the lines  $x = -2.5$  and  $y = 1.5$  at  $t^* = 3.0$  predicted by the ROM with and without correction, compared with the full-order solutions by numerical simulations.

## References

- [1] K. Taira, S.L. Brunton, S.T.M. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordelyev, V. Theofilis, L.S. Ukeiley, Modal analysis of fluid flows: An overview, *AIAA J.* 55 (12) (2017) 4013–4041.
- [2] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, *Comput. Methods Appl. Mech. Engrg.* 306 (2016) 196–215.
- [3] Z.Y. Wan, T.P. Sapsis, Reduced-space Gaussian process regression for data-driven probabilistic forecast of chaotic dynamical systems, *Physica D* 345 (2017) 40–55.
- [4] P.R. Vlachas, W. Byeon, Z.Y. Wan, T.P. Sapsis, P. Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks, *Proc. R. Soc. A* 474 (2213) (2018) 20170844.
- [5] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, A. Muggeridge, Non-intrusive reduced order modelling of the Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 293 (2015) 522–541.
- [6] S. Walton, O. Hassan, K. Morgan, Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions, *Appl. Math. Model.* 37 (20) (2013) 8930–8945.
- [7] D. Xiao, F. Fang, C. Pain, G. Hu, Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation, *Internat. J. Numer. Methods Fluids* 79 (11) (2015) 580–595.
- [8] O. San, R. Maulik, M. Ahmed, An artificial neural network framework for reduced order modeling of transient flows, *Commun. Nonlinear Sci. Numer. Simul.* 77 (2019) 271–287.
- [9] H.F.S. Lui, W.R. Wolf, Construction of reduced-order models for fluid flows using deep feedforward neural networks, *J. Fluid Mech.* 872 (2019) 963–994.
- [10] Z. Wang, D. Xiao, F. Fang, R. Govindan, C.C. Pain, Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, *Internat. J. Numer. Methods Fluids* 86 (4) (2018) 255–268.
- [11] P. Wu, J. Sun, X. Chang, W. Zhang, R. Arcucci, Y. Guo, C.C. Pain, Data-driven reduced order model with temporal convolutional neural network, *Comput. Methods Appl. Mech. Engrg.* 360 (2020) 112766.
- [12] E. Liberge, A. Hamdouni, Reduced order modelling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder, *J. Fluids Struct.* 26 (2) (2010) 292–311.
- [13] H. Gao, M. Wei, Domain decomposition in POD-Galerkin projection for flows with moving boundary, in: 54th AIAA Aerospace Sciences Meeting, 2016.

- [14] H. Gao, POD-Galerkin Based ROM for Fluid Flow with Moving Boundaries and the Model Adaptation in Parametric Space (Ph.D. thesis), Kansas State University, 2018.
- [15] D. Xiao, P. Yang, F. Fang, J. Xiang, C. Pain, I. Navon, Non-intrusive reduced order modelling of fluid–structure interactions, *Comput. Methods Appl. Mech. Engrg.* 303 (2016) 35–54.
- [16] C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- [17] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Program.* 45 (1989) 503–528.
- [18] A. Kolmogoroff, Über Die Beste Annäherung Von Funktionen Einer Gegebenen Funktionenklasse, *Ann. of Math.* 37 (1) (1936) 107–110.
- [19] P.W. Smith, N-widths in approximation theory (Allan Pinkus), *SIAM Rev.* 28 (2) (1986) 283–284.
- [20] C. Greif, K. Urban, Decay of the Kolmogorov n-width for wave problems, *Appl. Math. Lett.* 96 (2019) 216–222.
- [21] H. Wendland, *Scattered Data Approximation*, Vol. 17, Cambridge University Press, 2005.
- [22] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, *IMA J. Numer. Anal.* 32 (3) (2012) 983–1000.
- [23] H. Imtiaz, I. Akhtar, Closure modeling in reduced-order model of Burgers' equation for control applications, *Proc. Inst. Mech. Eng. G* 231 (4) (2017) 642–656.
- [24] K. Kunisch, S. Volkwein, Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition, *J. Optim. Theory Appl.* 102 (2) (1999) 345–371.
- [25] O. San, T. Iliescu, Proper orthogonal decomposition closure models for fluid flows: Burgers equation, *Int. J. Numer. Anal. Model.* 5 (2014) 217–237.
- [26] A.J. Chorin, Numerical solution of the Navier-Stokes equations, *Math. Comp.* 22 (104) (1968) 745–762.
- [27] M. Griebel, T. Dornseifer, T. Neunhoeffer, *Numerical Simulation in Fluid Dynamics: A Practical Introduction*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
- [28] J. Yang, K. Bhattacharya, Augmented Lagrangian digital image correlation, *Exp. Mech.* 59 (2019) 187–205.
- [29] L.T. Avellar, *Observations of Failure Phenomena in Periodic Media* (Ph.D. thesis), California Institute of Technology, 2018.
- [30] J. Yang, C. Franck, Strain stiffening effects of soft viscoelastic materials in inertial microcavitation, in: *Dynamic Behavior of Materials*, Vol. 1, Springer International Publishing, 2020, pp. 175–179.
- [31] X. Yang, C.C. Church, A model for the dynamics of gas bubbles in soft tissue, *J. Acoust. Soc. Am.* 118 (6) (2005) 3595–3606.
- [32] W. Hu, N. Trask, X. Hu, W. Pan, A spatially adaptive high-order meshless method for fluid–structure interactions, *Comput. Methods Appl. Mech. Engrg.* 355 (2019) 67–93.
- [33] I.G. Kevrekidis, C.W. Gear, J.M. Hyman, P.G. Kevrekidis, O. Runborg, C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: Enabling macroscopic simulators to perform system-level analysis, *Commun. Math. Sci.* 1 (4) (2003) 715–762.
- [34] I.G. Kevrekidis, G. Samaey, Equation-free multiscale computation: Algorithms and applications, *Ann. Rev. Phys. Chem.* 60 (1) (2009) 321–344.
- [35] K. Lee, K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, *J. Comput. Phys.* 404 (2020) 108973.