



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 317 (2017) 868–889

**Computer methods
in applied
mechanics and
engineering**

www.elsevier.com/locate/cma

A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications

D. Xiao^{a,*}, F. Fang^{a,*}, C.C. Pain^a, I.M. Navon^b

^a *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK*¹

^b *Department of Scientific Computing, Florida State University, Tallahassee, FL, 32306-4120, USA*

Received 16 August 2016; received in revised form 14 December 2016; accepted 22 December 2016

Available online 16 January 2017

Abstract

A novel parameterized non-intrusive reduced order model (P-NIROM) based on proper orthogonal decomposition (POD) has been developed. This P-NIROM is a generic and efficient approach for model reduction of parameterized partial differential equations (P-PDEs). Over existing parameterized reduced order models (P-ROM) (most of them are based on the reduced basis method), it is non-intrusive and independent on partial differential equations and computational codes. During the training process, the Smolyak sparse grid method is used to select a set of parameters over a specific parameterized space ($\Omega_p \in \mathcal{R}^P$). For each selected parameter, the reduced basis functions are generated from the snapshots derived from a run of the high fidelity model. More generally, the snapshots and basis function sets for any parameters over Ω_p can be obtained using an interpolation method. The P-NIROM can then be constructed by using our recently developed technique (Xiao et al., 2015 [41,42]) where either the Smolyak or radial basis function (RBF) methods are used to generate a set of hyper-surfaces representing the underlying dynamical system over the reduced space.

The new P-NIROM technique has been applied to parameterized Navier–Stokes equations and implemented with an unstructured mesh finite element model. The capability of this P-NIROM has been illustrated numerically by two test cases: flow past a cylinder and lock exchange case. The prediction capabilities of the P-NIROM have been evaluated by varying the viscosity, initial and boundary conditions. The results show that this P-NIROM has captured the quasi-totality of the details of the flow with CPU speedup of three orders of magnitude. An error analysis for the P-NIROM has been carried out.

© 2016 Elsevier B.V. All rights reserved.

Keywords: Parameterized; Non-intrusive ROM; PDE; RBF; POD; Smolyak sparse grid

* Corresponding authors.

E-mail addresses: dh.xiao@imperial.ac.uk (D. Xiao), f.fang@imperial.ac.uk (F. Fang).

¹ URL: <http://amcg.ese.imperial.ac.uk>.

1. Introduction

Reduced order models (ROMs) are popular and powerful techniques for circumventing the intensive computational burden in large complex numerical simulations in engineering and science, for example, ocean modelling, weather prediction, uncertainty quantification, sensitive analysis, data assimilation, sensor placement optimization, porous media, structural problem, convection diffusion reaction equations, molecular dynamics simulation and optimal control [1–16]. The basic idea of reduced order modelling is to find an approximate solution by a linear combination of a set of basis functions. The weighting coefficients of the basis functions are determined by error minimization or the Galerkin projection method [17].

One of challenges in reduced order modelling is to generate a robust ROM for different parameters/inputs, which can represent the physical dynamics of parameterized partial differential equations (P-PDEs) as the model parameters vary. Recently reduced basis method in combination with projection-based methods has been introduced and proven to be a very powerful means in model reduction of P-PDEs [18–24,17,25]. The high dimensional parameterized PDEs can be projected onto a low dimensional space which consists of a number of reduced basis functions. The construction of the reduced basis functions is based on snapshots (solutions of the original PDEs). These reduced basis functions can be constructed by either global or local approaches [18]. The ‘global’ reduced basis functions can be constructed by global snapshot matrices over the parameter space while the ‘local’ reduced basis functions can be obtained by interpolating the local snapshot matrices associated with a set of selected parameters over the parameter space. These reduced basis functions can then be used for constructing a P-ROM. The original matrices in the discretized PDEs and variables can be decomposed as a weighted linear combination of the reduced basis functions. The weighting coefficients for the reduced basis functions are dependent on time and model parameters.

Most of the existing P-ROMs are intrusive and dependent on original PDEs and codes (e.g. numerical schemes). In most cases, modifications are needed to generate the intrusive P-ROM. These modifications are difficult or even impossible in commercial software [26]. In addition, the intrusive ROM suffers from non-linear inefficiency and instability issues [27–29]. The methods of improving the stability of the ROM can be found in [30–33,4,34]. The approaches of enhancing the non-linearity efficiency have been developed in the work of [35–40].

In order to tackle these issues in intrusive ROMs, a number of non-intrusive reduced order models (NIROMs) have been developed recently [41–44]. However, very little work can be found addressing non-intrusive model reduction for parameterized PDEs, where inputs (e.g. initial and boundary conditions) and parameters (e.g. viscosity, material property) vary in space and time. Audouze et al. presented a proper orthogonal decomposition (POD) non-intrusive reduced order model for part of nonlinear parameterized PDEs [17,45]. The key idea underpinning the proposed method in [17] is to split the reduced-order approximation into two terms. The first term was the approximate solution of an auxiliary parabolic linear PDE, which enforces satisfaction of the boundary and initial conditions whereas the second term is a linear combination of a tensor product of adapted spatial and temporal basis functions obtained using the POD method. A non-intrusive approach based on radial basis function (RBF) (in contrast to Galerkin projection) was introduced to calculate the coefficients (weights) at the second term. However it is noted that to approximate the solutions of the auxiliary parabolic linear PDE, the classical POD-Galerkin approach was used, which is intrusive.

More recently, we have developed three types of NIROMs based on Taylor series expansion, Smolyak sparse grid and radial basis function (RBF) methods [41,42]. The NIROMs have been successfully applied to some realistic problems such as fluid and solid interaction [46] and porous media multi-phase problems [47]. In this work, we have further extended the NIROMs to parameterized physical problems (described by PDE’s).

A generic parameterized NIROM (P-NIROM) technique has been developed here for parameterized time-dependent linear/nonlinear physical problems. The P-NIROM developed here is independent of equations and easy to implement, and there is no need for an auxiliary parabolic linear PDE in contrast to the method proposed by Audouze et al. [17]. The whole procedure can be divided into the two steps: the offline process-training and online process-constructing and solving P-NIROM. During the training procedure, the parameter training points are first chosen using the Smolyak sparse grid method. The parameter vector $\mu \in \mathcal{R}^P$ is P -dimensional and each variable parameter denotes one dimension. The Smolyak sparse grid is constructed from a tensor product grid obtained over the parameter space. The solution snapshots on the parameter training points (sparse grids) are then obtained by running the original high fidelity model and the basis functions are computed using singular value decomposition (SVD)/POD. From these snapshots obtained on the parameterized training points, a set of POD basis functions on each parameterized training point is generated in an optimal sense that represents the fluid dynamics.

During the online computation procedure, a two-level RBF interpolation method is used for constructing the P-NIROM. Firstly, for any given parameter $\mu \in \mathcal{R}^P$, a set of snapshots and POD bases can be obtained using the first level RBF interpolation. The second level RBF interpolation is then used to construct a set of hyper-surfaces representing the dynamics of the original time-dependent PDEs. After the hyper-surfaces are obtained, the solution of the ROM at the current time level can be obtained by giving reduced solution at previous time level into the hyper-surface functions.

The above P-NIROM has been implemented under the framework of an unstructured mesh finite element model (Fluidity). The capabilities of this new NIROM have been assessed for two test cases: a flow past a cylinder case and a 2-D lock exchange case. Comparisons have been made between the high fidelity model and the P-NIROM to investigate the accuracy of the P-NIROM methodology.

The structure of the paper is as follows: Section 2 presents the general reduced order model for parameterized PDEs; Section 3 provides the details of calculation of POD basis functions for any given parameter over \mathcal{R}^P ; Section 4 describes the non-intrusive methods for model reduction of parameterized PDEs; Section 5 illustrates the P-NIROM method derived by means of two numerical examples: flow past a cylinder and lock exchange problem. Finally in Section 6, summary and conclusions are presented, and error analysis is provided in the Appendix.

2. General parameterized reduced order PDEs

In general, the parameterized space–time linear/nonlinear PDEs can be written as follows:

$$F(\mathbf{u}(\mathbf{x}, t, \mu), \mathbf{x}, t, \mu) = s(\mathbf{x}, t, \mu), \quad (1)$$

where $\mathbf{u}(\cdot, t, \mu) \in \mathcal{R}^{D \times N}$ is the state variable vector (including, for example, velocity components, pressure, temperature, etc. here N is the number of nodes in a scalar grid used in the computational domain and D is the number of scalars), \mathbf{x} is the spatial coordinate system, s denotes source term, t is the time and $\mu \in \mathcal{R}^P$ is the parameter vector (constructing a P dimensional parameter space).

In reduced order modelling, the state variable \mathbf{u} can be expressed as an expansion of the basis functions $\Phi(\mathbf{x}, \mu) = (\Phi_1, \dots, \Phi_m, \dots, \Phi_M)$ ($m \in (1, \dots, M)$, M is the number of basis functions and $M \ll N$):

$$\mathbf{u}(\mathbf{x}, t, \mu) = \Phi \mathbf{u}^r, \quad (2)$$

where $\mathbf{u}^r(t, \mu) \in \mathcal{R}^M$ is the reduced state variable vector (the superscript r indicates a variable or operator associated with the reduced order model). By using POD, the basis functions Φ of the variable are extracted and derived optimally from the snapshots sampled at time instants $\{t_1, \dots, t_i, \dots, t_{N_t}\}$:

$$\Phi_m(\mathbf{x}, \mu) = \sum_{i=1}^{N_t} \mathbf{u}(\mathbf{x}, t_i, \mu) \Upsilon_{m,i}, \quad m \in (1, \dots, M), \quad (3)$$

subject to

$$\sum_{m=1}^K |\langle \Phi_m, \Phi_m \rangle_{L^2}|^2 = 1, \quad (4)$$

where $\langle \cdot, \cdot \rangle_{L^2}$ is the canonical inner product in L^2 norm, M is the number of basis functions to be chosen (here $M < N_t \ll N$), and $\Upsilon_{m,i}$ is obtained using singular value decomposition (SVD):

$$B \Upsilon_m = \lambda_m \Upsilon_m, \quad (5)$$

where $\Upsilon_m = (\Upsilon_{m,1}, \dots, \Upsilon_{m,i}, \dots, \Upsilon_{m,N_t})$ and the matrix B has the form of,

$$B_{k,n} = \frac{1}{N_t} \int_{\Omega} \mathbf{u}(\cdot, t_k, \cdot) \mathbf{u}(\cdot, t_n, \cdot)^* dx, \quad k, n \in (1, \dots, N_t). \quad (6)$$

The singular values $\lambda = (\lambda_1, \dots, \lambda_m, \dots, \lambda_M)$ are listed in decreasing order. Projecting (1) onto the reduced space, yields,

$$\Phi^T F(\Phi \mathbf{u}^r(\mu, \mathbf{x}, t), \mathbf{x}, t, \mu) = \Phi^T s(\mathbf{x}, t, \mu). \quad (7)$$

The parameterized reduced order model in (7) can be re-written as:

$$F^r(\mathbf{u}^r(\boldsymbol{\mu}, \mathbf{x}, t), \mathbf{x}, t, \boldsymbol{\mu}) = s^r(\mathbf{x}, t, \boldsymbol{\mu}). \quad (8)$$

As discussed above, the traditional method to implement the parameterized reduced order model (8) is intrusive based on reduced basis methods. In this work, we propose a non-intrusive method for constructing the parameterized reduced order PDE model.

3. Computation of basis functions over the parameter space

In this section, the details of calculating the basis functions over the parameter space using Smolyak sparse grid and RBF interpolation methods are provided. Firstly, a number of parameter training points can be chosen using the Smolyak sparse grid method. The solution snapshots are then obtained for each training parameter by running the high fidelity model. The corresponding POD basis functions can be calculated using SVD/POD. Finally the solution snapshots or basis functions can be obtained for any given parameter $\boldsymbol{\mu} \in \mathcal{R}^P$ using the RBF interpolation method.

3.1. Choice of the parameter interpolation points using the Smolyak sparse grid

The Smolyak sparse grid is a numerical technique to interpolate or integrate high dimensional functions. It was developed by the Russian mathematician Smolyak, and it was designed to tackle the problem of ‘curse of dimensionality’ [48]. The key idea of the Smolyak sparse grid is that it selects a relatively small number of nodes on the full tensor-product grid in terms of potential importance of the nodes, thus resulting in great computational efficiency. In this case, only a small number of Smolyak nodes are involved in calculation rather than all the nodes on the full tensor-product grid. There is a approximation level, that controls how many nodes on the full tensor-product grid are selected. The higher the approximation level is chosen, the larger number of nodes will be used and higher approximation quality will be obtained, for more details, see [48,49]. Smolyak presented a rule that selects nodes from tensor product grid. The process of sampling the parameter interpolation points over the parameter space \mathcal{R}^P can be summarized as follows:

Let $Q_l^1 f$ be a quadrature rule on dimension 1 with N_l parameter points, it assumes the following form,

$$Q_l^1 f = \sum_{i=1}^{N_l} f(\boldsymbol{\mu}_l^i) \cdot \eta_l^i, \quad (9)$$

where l denotes the approximation level of sparse grid and f is the function on the interval $[0, 1]$ to be approximated. The η denotes the weight corresponding to the parameter point $\boldsymbol{\mu}_l^i$ and i denotes the i th points on the dimension.

In order to construct the sparse parameter interpolation points, a multi-index I is introduced and has the following form of,

$$I = \sum_{i=1}^P l_i. \quad (10)$$

The multi-index I determines the number of points selected from the tensor product grid and it satisfies the condition given in (14). Using the multi-index, the d -dimensional sparse grid quadrature formulation $Q_l^P f$ on the space $[0, 1]^P$ then can be defined as,

$$Q_l^P f = \sum_{|\mathbf{l}| \leq I+P-1} (\Delta_{l_1}^1 \otimes \cdots \otimes \Delta_{l_P}^P) f, \quad (11)$$

where Δ_l^1 is a difference quadrature rule, and is defined as,

$$\Delta_l^1 = (Q_l^1 - Q_{l-1}) f, \quad (12)$$

with

$$Q_0^1 f = 0, \quad (13)$$

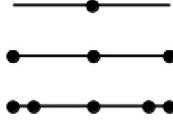


Fig. 1. The graph shows the 1-D Smolyak grid with level 0, level 1 and full tensor product grid (top to bottom) respectively.

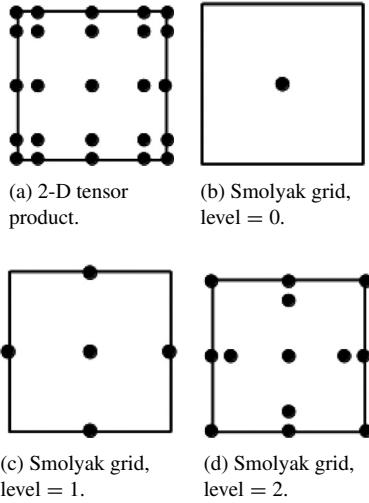


Fig. 2. The figures displayed above shows the full tensor product grid and 2-D Smolyak sparse grid with level 0, 1 and 2.

The Smolyak sparse grid satisfies the following condition:

$$P \leq i_1 + i_2 + \cdots + i_P \leq P + l, \quad (14)$$

which i_1, i_2 and i_P are indices corresponding to dimension 1, 2 and d respectively, and each one varies from 1 to the number of parameter points in one dimension. For example, in one dimension case, if there are three parameter points on the dimension, then $i_1 \in \{1, 2, 3\}$. Examples of Smolyak sparse grids of approximation levels 0, 1, 2 with dimension size 1 and 2 are illustrated in Figs. 1 and 2.

As can be seen in Figs. 1 and 2, the Smolyak sparse grid has a considerably decreased number of nodes. In one-dimensional cases, see Fig. 1, the full tensor product has 5 nodes while the Smolyak sparse grid has only 1 or 3 nodes depending on the level zero or one. In two-dimensional cases, see Fig. 2, the full tensor product has 25 nodes (5×5), while the Smolyak sparse grids with levels 0, 1 and 2 only have 1, 9 and 13 nodes respectively. It is worth noting that the number of nodes ratio for the full tensor product and Smolyak sparse grid increases as the dimension size increases. More details regarding the construction of Smolyak sparse grid can be found in [50,49]. In this work, each varying parameter constitutes one dimension in the parameter space which can be a high dimension space.

3.2. Interpolation of basis function and snapshots over the parameter space

By running the original high fidelity model (1) for each parameter point $\mu_p \in \mathcal{R}^P$ (here $p = (1, \dots, P)$, P is the number of parameter interpolation points) sampled using the Smolyak sparse grid method, one can obtain a set of snapshots $\mathbf{u}(\cdot, \cdot, \mu_p)$ and basis functions $\Phi(\cdot, \mu_p)$. For any given parameter $\mu \in \mathcal{R}^P$, the snapshots $\mathbf{u}(\cdot, \cdot, \mu)$ can be given using an interpolation function \mathcal{I} :

$$\mathbf{u}(\cdot, \cdot, \mu) = \mathcal{I}(\mathbf{u}(\cdot, \cdot, \mu_1), \dots, \mathbf{u}(\cdot, \cdot, \mu_P)). \quad (15)$$

There are a lot of interpolation methods to choose from. For a high dimensional parameter space, we may choose the Smolyak sparse grid method described above. Here we will introduce the RBF interpolation method. The RBF is a function that its value depends on the distance from the origin or some other interpolation points. The RBF

interpolation method constructs an approximate function through a number of random data points, here the parameter interpolation points chosen using the Smolyak sparse grid method.

Let $\mathcal{I}(\boldsymbol{\mu})$ denote the interpolation function representing $\mathbf{u}(\cdot, \cdot, \boldsymbol{\mu})$ and $\Phi(\cdot, \boldsymbol{\mu})$, and has the form of,

$$\mathcal{I}(\boldsymbol{\mu}) = \sum_{p=1}^P w_p \phi(\|\boldsymbol{\mu} - \boldsymbol{\mu}_p\|), \quad (16)$$

where $\mathcal{I}(\boldsymbol{\mu})$ denotes the approximating function, and is a sum of P radial basis functions ϕ , each RBF associated with a different centre $\boldsymbol{\mu}_p$, and weighted by a coefficient w_p . P is the number of training data points. The norm is usually chosen to be Euclidean distance. The frequently used RBFs can be either multi-quadric, inverse quadratic, Gaussian, plate spline or inverse multi-quadric. In this work, the Gaussian RBF is chosen, which has a form of $\phi(r) = e^{-(r/\sigma)^2}$ (r being radius and σ being the shape parameter). The weights $w = (w_1, \dots, w_p, \dots, w_P)^T$ can be obtained by solving the linear equation (17),

$$Aw = b, \quad (17)$$

where b is a vector containing real functional values on the training parameter points,

$$A = \begin{bmatrix} \phi(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1\|) & \phi(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|) & \cdots \phi(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_P\|) \\ \phi(\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|) & \phi(\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_2\|) & \cdots \phi(\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_P\|) \\ \vdots & \vdots & \vdots \\ \phi(\|\boldsymbol{\mu}_P - \boldsymbol{\mu}_1\|) & \phi(\|\boldsymbol{\mu}_P - \boldsymbol{\mu}_2\|) & \cdots \phi(\|\boldsymbol{\mu}_P - \boldsymbol{\mu}_P\|) \end{bmatrix}. \quad (18)$$

The process of constructing a set of snapshots over the parameter space is summarized in algorithm 1.

4. Construction of P-NIROM for parameterized PDEs

In this section, we provide the details of constructing and solving the P-NIROM. The basic idea is to construct a set of hyper-surfaces for representing the physical dynamics of the parameterized PDEs. The parameterized ROM in (8) can be re-written at each time level t^{n+1} :

$$\mathbf{u}^{r,n+1}(\mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{u}_1^{r,n}(\mathbf{x}, \boldsymbol{\mu}), \dots, \mathbf{u}_m^{r,n}(\mathbf{x}, \boldsymbol{\mu}), \dots, \mathbf{u}_M^{r,n}(\mathbf{x}, \boldsymbol{\mu})), \quad (19)$$

where the superscript n represents the time level. In this work, the recently developed non-intrusive ROM approach is used for constructing the P-NIROM of (19). Using the RBF or Smolyak methods, the hyper-surface sets f_m ($m \in (1, \dots, M)$) are constructed to represent the physical dynamics of the original PDEs over the reduced space:

$$\mathbf{u}_m^{r,n+1} = f_m(\mathbf{u}_1^{r,n}, \dots, \mathbf{u}_m^{r,n}, \dots, \mathbf{u}_M^{r,n}), \quad m \in (1, \dots, M), \quad (20)$$

where $f_m \in \mathcal{R}^{M+1}$ is a $M + 1$ dimensional surface. Using the RBF, a set of hyper-surfaces f_m for any parameter set $\boldsymbol{\mu}$ over the parameter space can be expressed below:

$$f_m(\mathbf{u}^{r,n}(\boldsymbol{\mu})) = \sum_{n_t=1}^{N_t} w_m^{n_t} \phi(\|\mathbf{u}^{r,n}(\boldsymbol{\mu}) - \mathbf{u}^{r,n_t}(\boldsymbol{\mu})\|), \quad m \in (1, \dots, M), \quad (21)$$

where the weights $\mathbf{w}_m^{n_t}$ can be obtained by solving:

$$\begin{bmatrix} \phi\left(\left\|\mathbf{u}^{r,1} - \mathbf{u}^{r,1}\right\|_2\right) & \phi\left(\left\|\mathbf{u}^{r,1} - \mathbf{u}^{r,2}\right\|_2\right) & \cdots \phi\left(\left\|\mathbf{u}^{r,1} - \mathbf{u}^{r,N_t}\right\|_2\right) \\ \phi\left(\left\|\mathbf{u}^{r,2} - \mathbf{u}^{r,1}\right\|_2\right) & \phi\left(\left\|\mathbf{u}^{r,2} - \mathbf{u}^{r,2}\right\|_2\right) & \cdots \phi\left(\left\|\mathbf{u}^{r,2} - \mathbf{u}^{r,N_t}\right\|_2\right) \\ \vdots & \vdots & \vdots \\ \phi\left(\left\|\mathbf{u}^{r,N_t} - \mathbf{u}^{r,1}\right\|_2\right) & \phi\left(\left\|\mathbf{u}^{r,N_t} - \mathbf{u}^{r,2}\right\|_2\right) & \cdots \phi\left(\left\|\mathbf{u}^{r,N_t} - \mathbf{u}^{r,N_t}\right\|_2\right) \end{bmatrix} \begin{pmatrix} w_m^1 \\ w_m^2 \\ \vdots \\ w_m^{N_t} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_m^{r,1} \\ \mathbf{u}_m^{r,2} \\ \vdots \\ \mathbf{u}_m^{r,N_t} \end{pmatrix}, \quad (22)$$

Algorithm 1: Constructing a set of snapshots and basis functions over the parameter space**Offline procedure: Calculating the snapshots or basis functions over the parameter interpolation points**

- (1) Construct a parameter space \mathcal{R}^P . That is, determine the dimensional size P of the parameter space and the parameter range along each dimensional direction. The varying parameters then constitute a tensor product grid;
- (2) Generate a Smolyak sparse grid over the parameter space, $\mu_1, \dots, \mu_p, \dots, \mu_P$;
- (3) Generate snapshots $\mathbf{u}(\cdot, \cdot, \mu_p)$ for each parameter vector μ_p ($p \in (1, \dots, P)$) by solving the high fidelity model over the simulation time period $[0, T]$;
- (4) Calculate POD basis functions $\Phi(\mu_p)$ for each node through a truncated SVD of the snapshots matrix;

Online procedure: Construct the basis functions for any parameter set μ over the parameter space

- (1) Calculate snapshots $\mathbf{u}(\cdot, t, \mu) \in \mathcal{R}^N$ for a new arbitrary parameter point μ within the domain of the tensor product grid through the interpolation surface using the following loop:

for $j = 1$ to N **do**

- (i) Calculate the weights $\mathbf{w}_j = (w_{j,1}, \dots, w_{j,P})^T$ by solving:

$$A\mathbf{w}_j = b_j, \quad b_j = (\mathbf{u}(j, t, \mu_1), \dots, \mathbf{u}(j, t, \mu_P))^T.$$

- (ii) Obtain an interpolation function $(\mathbf{u}(j, t, \mu) = \mathcal{I}_j(\mu))$ for calculating the snapshots by substituting the weights into following equations,

$$\mathcal{I}_j(\mu) = \sum_{p=1}^P w_{j,p} \phi(\|\mu - \mu_p\|).$$

- (iii) Obtain the snapshots $\mathbf{u}(\cdot, t, \mu_k)$ for any given parameter $\mu_k \in \mathcal{R}^P$ using:

$$\mathcal{I}_j(\mu_k) = \sum_{p=1}^P w_{j,p} \phi_j(\|\mu_k - \mu_p\|).$$

endfor

- (2) Calculate the basis functions $\Phi(\mathbf{x}, \mu_k)$ based on the snapshots $\mathbf{u}(\cdot, t, \mu_k)$ using SVD described (3) - (5) in section 2.

where $\mathbf{u}^{r,n_t}(\mu) = (\mathbf{u}_1^{r,n_t}(\mu), \dots, \mathbf{u}_M^{r,n_t}(\mu))$ ($n_t \in (1, \dots, N_t)$) are the reduced numerical solution for any parameter set $\mu \in \mathcal{R}^P$, which can be obtained:

- (1) Using algorithm 1, we obtain a set of snapshots $\{\mathbf{u}^{n_t}(\mathbf{x}, \mu)\}$ and basis function $\Phi(\mathbf{x}, \mu)$;
- (2) Projecting $\{\mathbf{u}^{n_t}(\mathbf{x}, \mu)\}$ over the reduced space which is constituted by $\Phi(\mathbf{x}, \mu)$, the reduced order solution, $\mathbf{u}^{r,n_t}(\mu)$, is calculated.

The construction of P-NIROM is summarized in algorithm 2, the procedure of solving the P-NIROM is provided in algorithm 3. By projecting the reduced solution $\mathbf{u}^{r,n+1}$ at time level $n + 1$, we can obtain the approximation of the high fidelity solution using Eq. (2).

5. Numerical examples

Two examples are presented in this section to illustrate the capabilities of the P-NIROM in resolving flow problem governed by the Navier–Stokes equations. In the first example a flow past a cylinder is solved. This is used to demonstrate the predictive capabilities of the P-NIROM as the boundary condition (inlet velocity) and model parameter (viscosity) vary. In the second example a 2-D lock exchange problem is solved. This example involves varying the initial conditions (initial temperature distribution). Both examples were simulated under the framework of an advanced unstructured mesh finite element model (Fluidity) [51].

Algorithm 2: Online procedure: constructing a P-NIROM for parameterized PDEs

-
- (1) Obtain the full solution snapshots \mathbf{u}^{n_t} ($n_t = 1, \dots, N_t$) for any parameter set $\boldsymbol{\mu} \in \mathcal{R}^P$ by interpolating $\mathbf{u}^{n_t}(\boldsymbol{\mu}_p)$ over the parameter training points $\boldsymbol{\mu}_p$ (here $p \in (1, \dots, P)$);
 - (2) Calculate the reduced solution $\mathbf{u}^{r,n_t}(\mathbf{u})$ by projecting \mathbf{u}^{n_t} onto the reduced space;
 - (3) Obtain a set of hyper surfaces $\{f_m\}$, $m \in (1, \dots, M)$ for each basis function $\{\Phi_m\}$ through the following loop:


```
for m = 1 to M do
  (i) Calculate the weights  $w_m^{n_t}$  by solving (22);
  (ii) Obtain a hyper-surface  $f_m$  for the basis function  $\Phi_m$  using the RBF;
endfor
```
 - (4) Construct the P-NIROM for calculating $\{\mathbf{u}_m^{r,n+1}\}$, ($m \in (1, \dots, M)$) at time level $n + 1$:

$$\mathbf{u}_m^{r,n+1}(\boldsymbol{\mu}) = f_m(\mathbf{u}_1^{r,n}(\boldsymbol{\mu}), \dots, \mathbf{u}_m^{r,n}(\boldsymbol{\mu}), \dots, \mathbf{u}_M^{r,n}(\boldsymbol{\mu}))$$
-

Algorithm 3: Online procedure: solve the P-NIROM and obtain the approximation of the high fidelity solution

- (1) Initialization.


```
for m = 1 to m do
  Initialize  $\mathbf{u}_m^0$ ;
endfor
```
 - (2) Calculate reduced numerical solutions at the current time step (here NT is the number of time levels):


```
for n = 1 to NT do
  for m = 1 to M do
    (i) Assign a complete set of the reduced solution  $\mathbf{u}^{r,n} = (\mathbf{u}_1^{r,n}, \dots, \mathbf{u}_M^{r,n})$  at previous time level  $n$  into the hyper-surface  $f_m$ :
        
$$f_m \leftarrow (\mathbf{u}_1^{r,n}, \dots, \mathbf{u}_m^{r,n}, \dots, \mathbf{u}_M^{r,n})$$

    (ii) Calculate  $\mathbf{u}_m^{r,n+1}$  at the current time level  $n + 1$  using:
        
$$\mathbf{u}_m^{r,n+1}(\boldsymbol{\mu}) = f_m(\mathbf{u}_1^{r,n}(\boldsymbol{\mu}), \dots, \mathbf{u}_m^{r,n}(\boldsymbol{\mu}), \dots, \mathbf{u}_M^{r,n}(\boldsymbol{\mu}))$$

  endfor
  Obtain the approximation of the high fidelity solution at the current time level  $n + 1$  by projecting  $\mathbf{u}^{r,n+1}(\boldsymbol{\mu})$  onto the full space using:
  
$$\mathbf{u}^{n+1}(\mathbf{x}, t, \boldsymbol{\mu}) = \sum_{m=1}^M \mathbf{u}_m^n \Phi_m$$


```
 - endfor**
-

5.1. Case 1: flow past a cylinder

In this example a two dimensional flow past a cylinder is numerically simulated. The problem domain is presented in Fig. 7 which shows a cylinder with a radius of 0.12 at location (0.2, 0.2) embedded within a rectangle with a length of 2 and a width of 0.4 (a non-dimensional unit). The fluid dynamics of the flow is driven by an inlet velocity, which enters from the left boundary of the domain. The fluid is allowed to flow past through the right boundary of the domain. No slip and zero outward flow conditions are applied to the lower and upper edges and the Dirichlet boundary conditions are applied to the cylinder's wall. The simulation time period is [0, 6], and the time step size is $\Delta t = 0.01$. 300 snapshots were sampled at an equal time interval of $\Delta t = 0.02$ during the time period [0, 6]. The computational domain consists of 3213 nodes.

5.1.1. Case 1a: one-dimensional parameter space (inlet velocity)

In this test case, the parameter to vary is the inlet velocity, i.e. $\boldsymbol{\mu} = \mathbf{u}|_x = 0$ (where $\boldsymbol{\mu} \in \mathcal{R}$, one-dimensional). The inlet velocity was ranged from 0.45 to 0.55. Three inlet velocities, 0.45, 0.5 and 0.55, were chosen as the training points within the parameter space $\Omega_p \in [0.45, 0.55]$ using the Smolyak sparse grid with an approximate level $l = 1$.

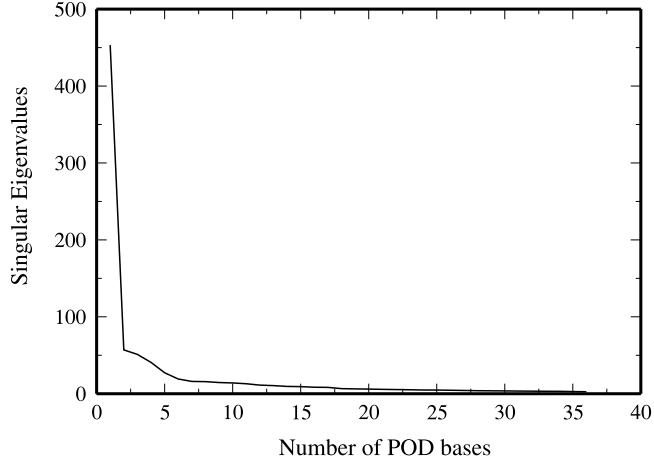


Fig. 3. Case 1a: the figure shows the first 36 eigenvalues in a decreasing order.

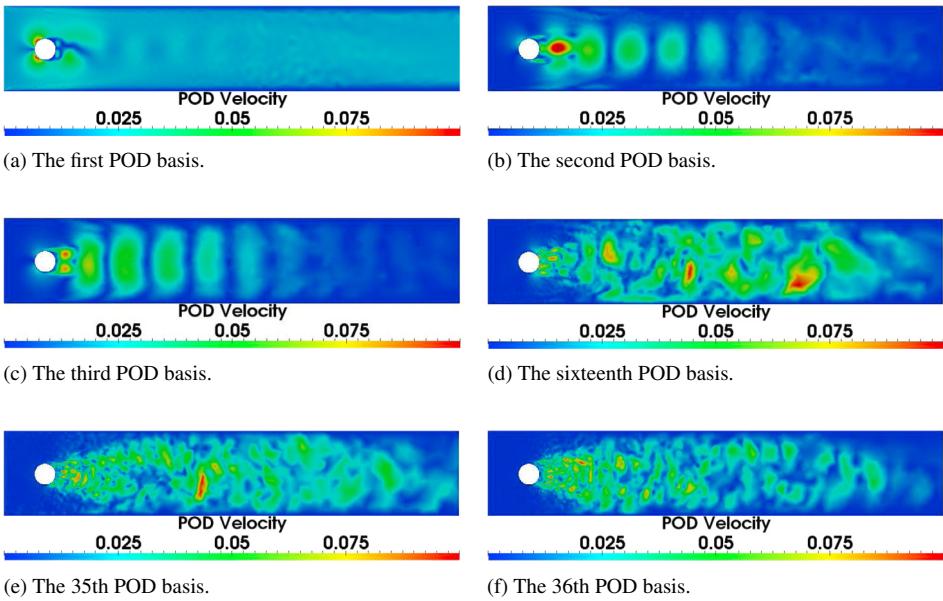


Fig. 4. Case 1a: the figure shows some of the first 36 POD basis functions of flow past a cylinder test case.

The solution snapshots ($\mathbf{u}(\cdot, n_t, \mu_p)$) over the training points were obtained by running the high fidelity model for each training inlet velocity. The P-NIROM was then constructed from these solution snapshots over the training inlet velocity points. To demonstrate the capability of the P-NIROM, a new (untrained) inlet velocity of 0.46, was chosen. The snapshots for the new inlet velocity were computed by interpolating $\mathbf{u}(\cdot, n_t, \mu_p)$ using the RBF method, where the multiquadric basis function was used. The POD basis functions were then obtained by POD–SVD. The P-NIROM for the new inlet velocity was constructed by a set of hyper-surfaces (see algorithm 3).

The singular values are presented in Fig. 3. It can be seen that there is a sharp drop in the first five singular values. The POD basis function associated with a larger eigenvalue can capture the more energy in the original flow dynamical system. This can be confirmed on inspection of the basis function in Fig. 4 which shows the first, second and third basis functions capturing 50.1%, 14.2% and 9.1% of the total energy respectively, and the 16th, 35th and 36th basis functions capturing 0.44%, 0.019% and 0.017% of the total energy respectively. Fig. 3 provides us a clear indication how to choose the number of basis functions to represent the original dynamic system.

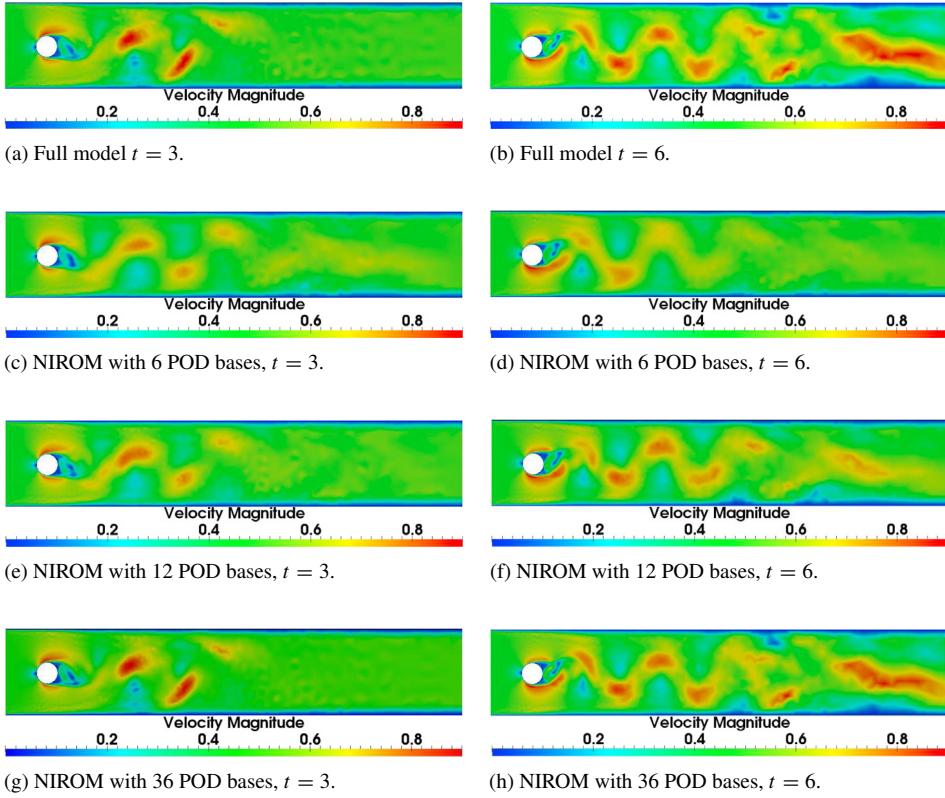


Fig. 5. Case 1a: the figure shows the velocity from full model and the NIROM with 6, 12 and 36 POD bases at time instances 3 and 6.

In this example, 6, 12 and 36 basis functions are selected to construct the P-NIROMs. Fig. 5 presents the velocity solution obtained from the high fidelity model and P-NIROMs using 6, 12 and 36 basis functions at time levels $t = 3$ and $t = 6$. Whilst there are visual differences between the high fidelity model and NIROM using 6 basis functions, the P-NIROM has still captured the dominant flow structure. With increased POD basis functions, the P-NIROM can capture the details of the flow very well in comparison with the high fidelity model. This can be confirmed on inspection of Fig. 6, which presents the error of solutions between the fidelity model and P-NIROMs with 6, 12 and 36 basis functions. The error is evidently decreased when the number of basis functions is increased to 36. Fig. 7 provides a comparison of the velocity profile from the high fidelity model with the P-NIROM using 6, 12 and 36 basis functions at a particular point ($x = 0.89514$, $y = 0.32519$). It shows that the P-NIROM solutions with 36 basis functions are in close agreement with the high fidelity solutions although there is a slight difference at the perturbation peaks.

The error analysis of NIROM has been further carried out using the root-mean-square error (RMSE) and correlation coefficient of solutions between the high fidelity model and P-NIROMs, which consider all the computational nodes on the mesh and all the simulation time levels. Fig. 8 shows the RMSE and correlation coefficient between the high fidelity model and NIROMs with 6, 12 and 36 basis functions. Again we can see that the P-NIROM with more basis functions exhibits higher prediction accuracy.

5.1.2. Case 1b: two-dimensional parameter space (inlet velocity and viscosity)

To further test the predictive capability of the P-NIROM, two parameters have been varied: the inlet velocity and viscosity, which construct a two-dimensional parameter space. Table 1 lists a combination of varying parameters in the case of the flow past a cylinder — labelled A1–A13 are the training parameter points, which are chosen using the Smolyak grid method with an approximate level of 2. In Table 1, μ_1 and μ_2 denote the inlet velocity and viscosity respectively. Two new (untrained) parameter points ($0.5, 0.833 \times 10^{-4}$) and ($0.525, 0.5 \times 10^{-4}$) — labelled T1 and T2 are chosen to demonstrate the capability of the P-NIROMs. The two parameters constitute a 2-D Smolyak sparse

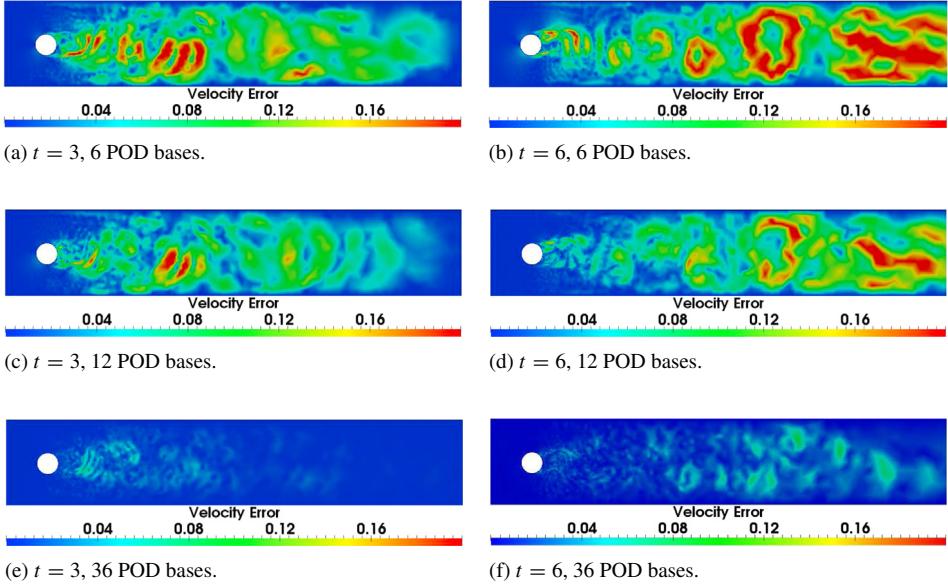


Fig. 6. Case 1a: the figure shows the velocity error between the high fidelity model and P-NIROMs with 6, 12 and 36 POD bases at time instances 3 and 6.

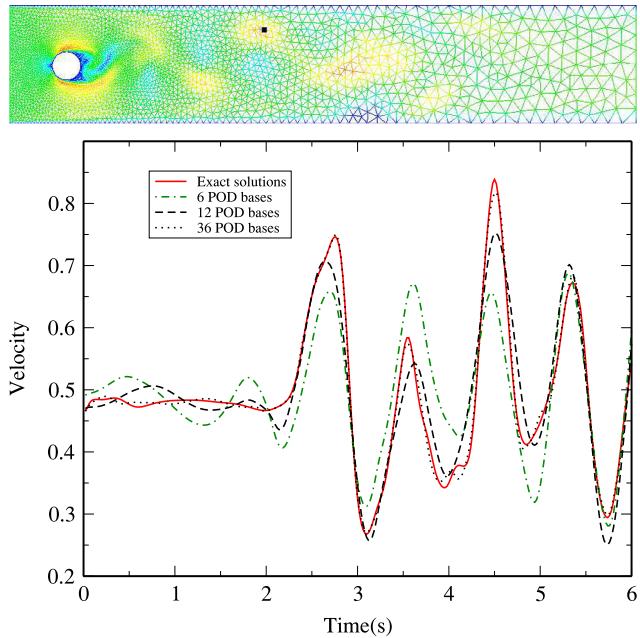


Fig. 7. Case 1a: the figure shows the velocity profile at location ($x = 0.89514$, $y = 0.32519$) from the high fidelity model and P-NIROM with 6, 12 and 36 basis functions.

grid — as shown in Fig. 9. In this figure, solid dotted points are training points (A1–A13) and the circled points (T1 and T2) are untrained points used to demonstrate the predictive capability of the P-NIROM.

Fig. 10 shows the velocity solution from the high fidelity model and P-NIROM at the untrained point T1 at time levels 3 and 4.6. It can be seen that the P-NIROM predicted the flow pattern well. Visually, there is little difference between the high fidelity model and P-NIROM. In order to see the difference, the velocity solutions obtained from the full model and NIROM at a particular point within the computational domain ($x = 0.32289$, $y = 0.34007$) are compared in Fig. 11. Again, there are slight differences at time instances 0.4–0.8 and 2.

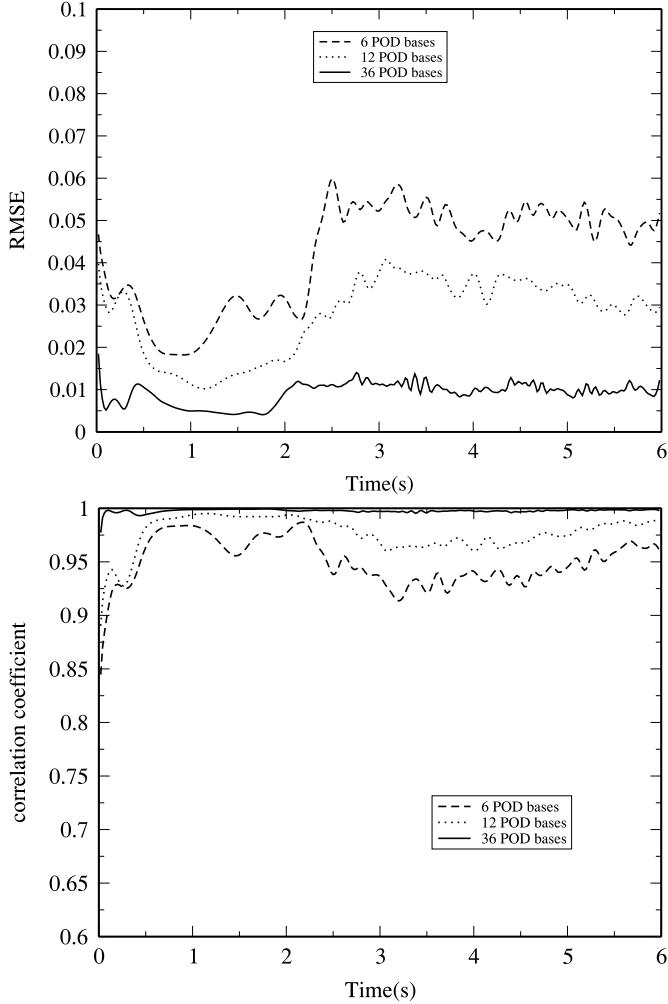


Fig. 8. Case 1a: the figure shows the root mean squared error (RMSE) and correlation coefficient between the high fidelity model and NIROMs with 6, 12 and 36 POD bases.

The velocity solutions from the high fidelity model and P-NIROMs at the untrained point T2 at time levels 3 and 4.6 are presented in Fig. 12. Again, the visual difference is not obvious. In order to see the little difference, Fig. 13 compares the velocity solutions between the high fidelity model and P-NIROM at a particular point in the computational domain ($x = 0.44274$, $y = 0.35188$) for the untrained point T2. As shown in this figure, the P-NIROM can predict the velocity solution at the untrained point well.

5.2. Lock exchange

In this case, the P-NIROM is used for resolving a lock exchange problem which involves two fluids of different temperature and density separated by a lock. When the lock is removed, two currents propagate along the tank horizontally. This laboratory-scale set up incorporates dynamics observed in gravity currents over a range of scales [52]. The problem computational domain is presented in Fig. 14 which consists of a non-dimensional rectangle (0.8×0.1). The initial conditions for the velocities and pressure are $\mathbf{u}_0 = 0$ and $p_0 = 0$ respectively. The isotropic value of viscosity is 1×10^{-10} .

The problem was resolved using the mesh which consisted of 4242 nodes, and 120 snapshots were obtained at regularly spaced time intervals during the time period [0–30] from the high fidelity solutions. In this test case, the parameter to vary is the temperature. The parameterized initial temperature condition, $\mu = T_0 \in \mathcal{R}$, was varied from

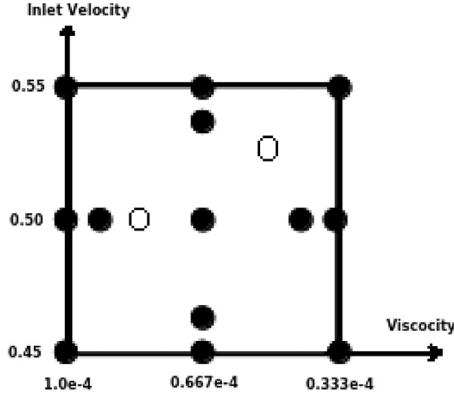


Fig. 9. Case 1b: two-dimensional parameter sparse grids, where solid dot: parameter training points; circle: new (untrained) parameter points.

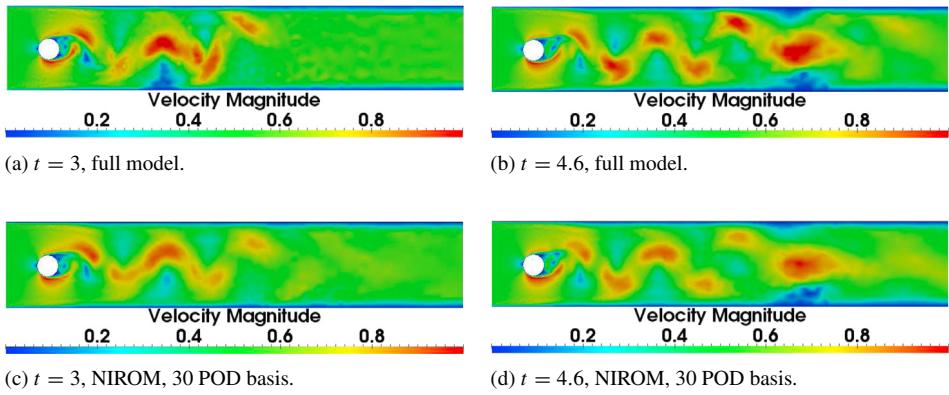


Fig. 10. Case 1b: the figure shows the velocity solutions from high fidelity model and P-NIROMs at an untrained point T1 at time instances 3 and 4.6.

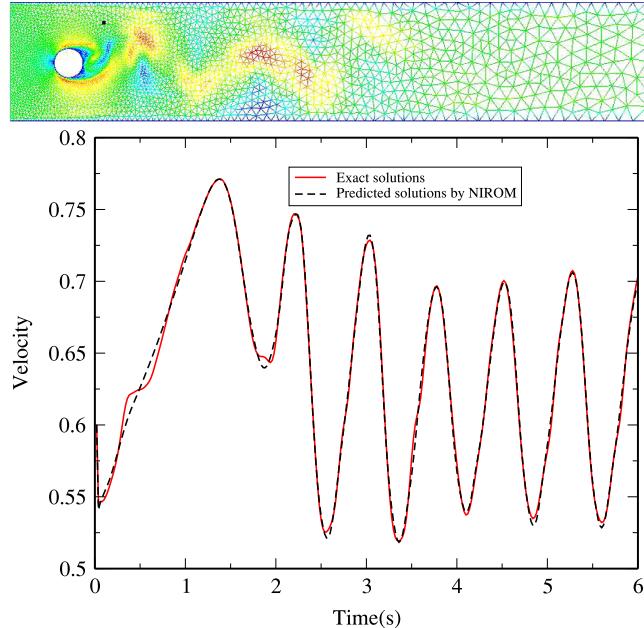


Fig. 11. Case 1b: comparison of the velocity profile at $x = 0.32289$, $y = 0.34007$ at an untrained parameter point T1.

Table 1

A list of combination of training parameters for the flow past a cylinder (parameter one μ_1 : inlet velocity; parameter two μ_2 : viscosity).

Cases	μ_1	μ_2	Reynolds	Cases	μ_1	μ_2	Reynolds
A1	0.5000	0.667×10^{-4}	1500	A8	0.450	0.333×10^{-4}	2700
A2	0.4500	0.667×10^{-4}	1350	A9	0.550	0.333×10^{-4}	3300
A3	0.5500	0.667×10^{-4}	1650	A10	0.450	1.000×10^{-4}	900
A4	0.5000	0.333×10^{-4}	3000	A11	0.550	1.000×10^{-4}	1100
A5	0.5000	1.000×10^{-4}	1000	A12	0.500	0.431×10^{-4}	2320
A6	0.4646	0.667×10^{-4}	1394	A13	0.500	0.902×10^{-4}	1109
A7	0.5354	0.667×10^{-4}	1606				
T1	0.5000	0.833×10^{-4}	1200	T2	0.525	0.5×10^{-4}	2100

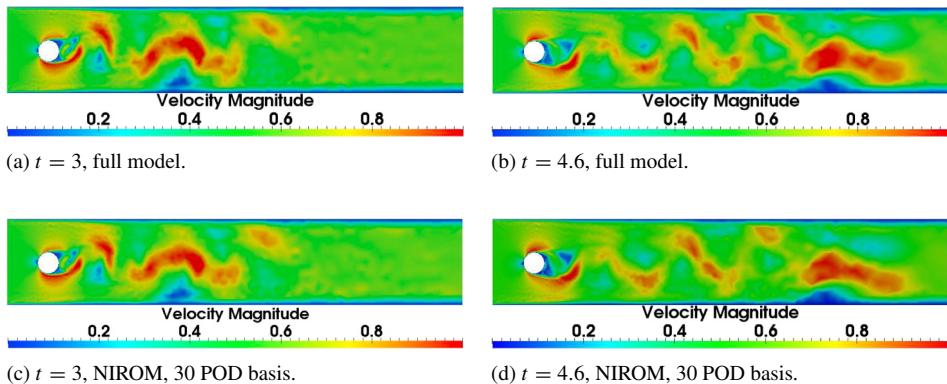


Fig. 12. Case 1b: the velocity solutions from high fidelity model and P-NIROMs at an untrained parameter point T2 at time instances 3 and 4.6.

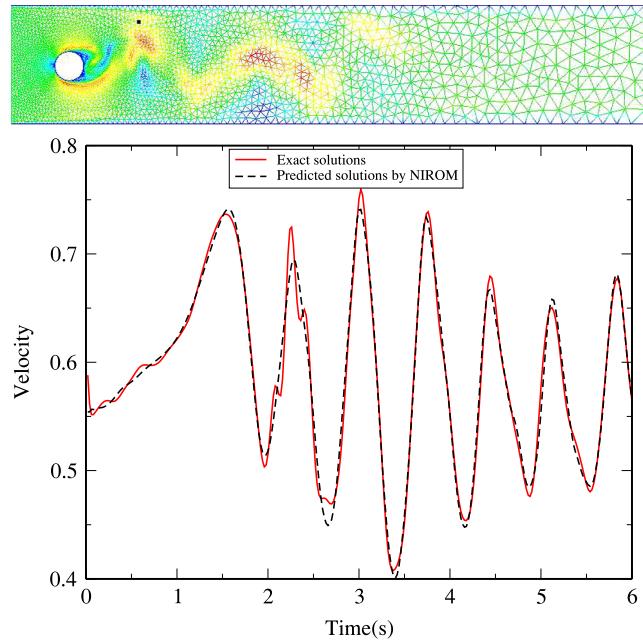


Fig. 13. Case 1b: comparison of the velocity profile at $x = 0.44274$, $y = 0.35188$ at an untrained point T2.

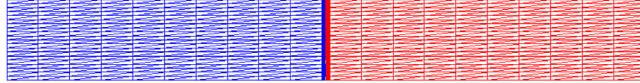


Fig. 14. Lock exchange: the graph shows the computational domain of the 2-D lock exchange problem.

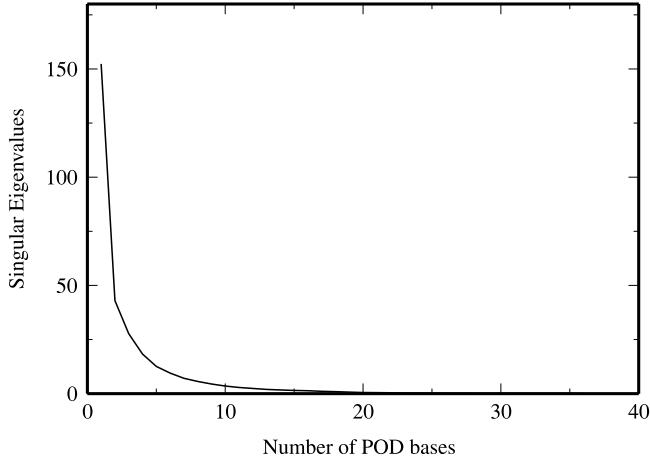


Fig. 15. Lock exchange: the graph shows the singular values of the 2-D lock exchange problem.

0.45 to 0.55. It was set to be $-T_0$ for the cold fluid at the left side of the lock T for the hot fluid at the right side of the lock. Using the Smolyak sparse grid with one approximational level, three training data points were selected: $T_0 = 0.45, 0.5$ and 0.55 . The solution snapshots ($\mathbf{u}(\cdot, n_f, \mu_p)$) over the training points were obtained by running the high fidelity model for each training initial temperature value. An untrained initial temperature ($T_0 = 0.46$) was chosen to show the capability of the P-NIROM, that is, the initial temperature of the hot fluid was set to be -0.46 and 0.46 for the cold fluid. For the given new initial temperature $T_0 = 0.45$, using the RBF interpolation method (the multiquadric basis function was used here), the snapshots were from these solution snapshots over the training temperature points. A set of hyper-surfaces using the RBF was then generated for representing the flow dynamics of the ordinal PDEs.

Fig. 15 presents the singular values of the problem in a decreasing order. In this problem, the reduction in the first 10 singular values is very fast, which means the first 10 leading basis functions corresponding to these 10 singular values capture most of the energy in the original dynamic system. Some of the first 36 basis functions are presented in Fig. 16, which shows the first few basis functions capture the general velocity pattern while the last few basis functions the minor details of velocity structures. In this example 6, 12 and 36 basis functions are chosen to generate the P-NIROM.

Fig. 17 presents the temperature solutions obtained from the high fidelity model and P-NIROM with 6, 12 and 36 basis functions. In comparison to the solution from the high fidelity model, the P-NIROMs appear to be minor visual differences between all the temperature solutions. However, the temperature solutions predicted from P-NIROM with 6 and 12 basis functions are shown to be diffusing a little bit slower than NIROM with 36 basis functions at the time level ($t = 30$). Using a larger number of basis functions results in higher accuracy of the P-NIROM. The P-NIROM with 36 basis functions is almost identical to the high fidelity model. This can be confirmed by inspection of the error Fig. 18, which shows the solution differences between the high fidelity model and P-NIROM with 6, 12 and 36 basis functions. It is found that the difference of solutions between the high fidelity model and P-NIROM with 36 basis functions is too small to notice. The error of P-NIROM is further analysed by RMSE and correlation coefficients, which is presented in Fig. 19. It can be seen that the RMSE and correlation coefficient curves of P-NIROM with 36 basis functions are flat at this scale, and the RMSE of solutions is close to 0 and the correlation coefficient is very close to 1. This means that in this case P-NIROM using 36 basis functions has captured almost 99.99% energy of the original flow dynamical system.

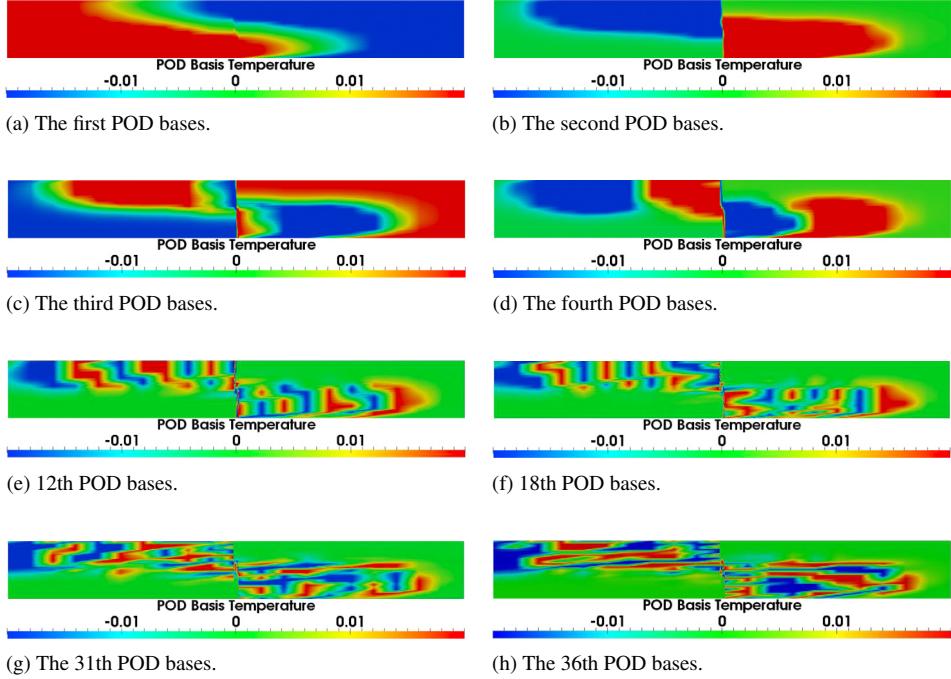


Fig. 16. Lock exchange: the figure shows some of the first 36 basis functions of the problem.

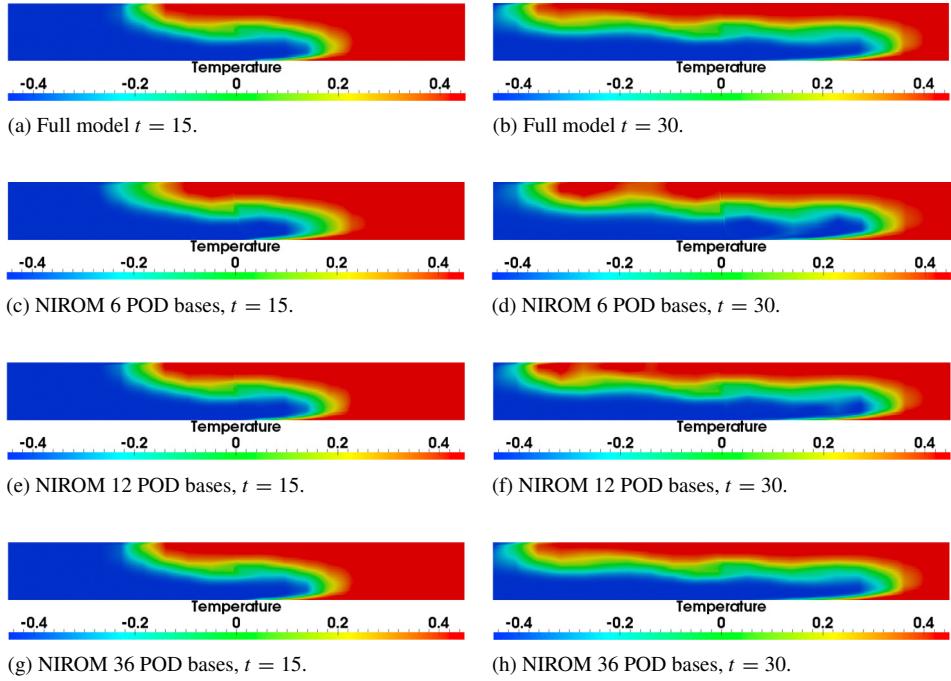


Fig. 17. Lock exchange: the figures displayed above show the temperature from the high fidelity model and the P-NIROM using 6, 12 and 36 POD basis functions at time instances 15 and 30.

5.3. Efficiency of the P-NIROM model

This section compares the online computational cost required by the high fidelity model and NIROM. The specifications of the computer for simulations were: 4 cores with a frequency of 2.00 GHz (*Intel Core™ i7-3537U*

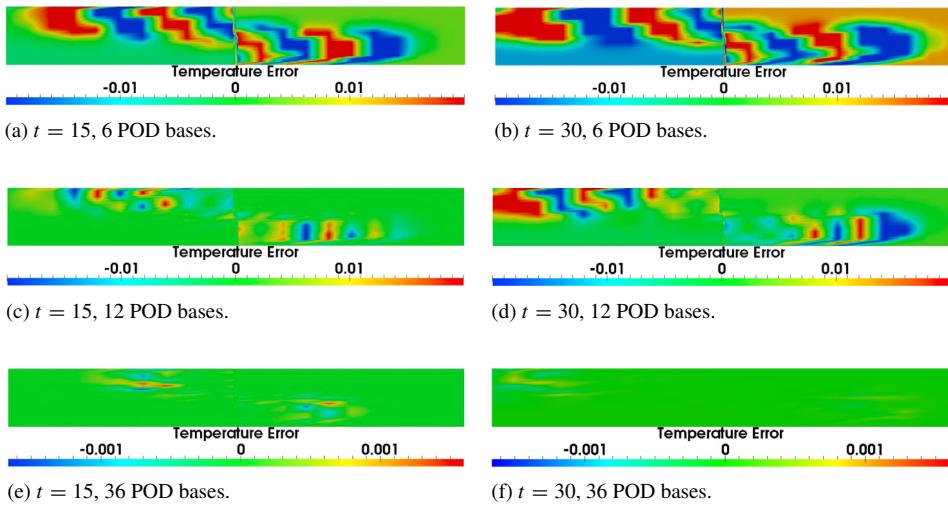


Fig. 18. Lock exchange: the figures show the temperature error between high fidelity model and NIROM with 6, 12 and 36 POD bases at time instances 15 and 30.

Table 2

Online dimensionless CPU cost required for simulating the two test cases using the full model and NIROM during one time step.

Cases	Model	Assembling and solving	Projection	Interpolation	Total
Flow past a cylinder	Full model	0.5891	0	0	0.6002
	NIROM	0	0.0003	0.0001	0.0004
Lock exchange	Full model	0.9489	0	0	0.95003
	NIROM	0	0.0003	0.0001	0.0004

CPU @ 2.00 GHz × 4); a 8 GB memory. One core was used when running the simulations since the test cases were simulated in serial.

Table 2 lists the online CPU cost required for simulating the flow past a cylinder and lock exchange test cases using the full model and NIROM. The offline cost involving constructing the basis functions is not listed in this table.

As shown in the table that the online CPU time required for the NIROM is substantially less than that for high fidelity model. The reduction in CPU time is dependent on the complexity of problem.

6. Conclusion

This article has presented a general P-NIROM technique for model reduction of parameterized time-dependent nonlinear PDEs. It is non-intrusive (independent of equations and numerical discretized schemes/codes) and easy to implement, especially for complex dynamic codes (e.g. unstructured mesh models). The procedure of constructing a P-NIROM can be split into the offline and online procedures. During the offline (training) procedure, the parameter training points are selected using the Smolyak sparse grid. The solution snapshots and POD basis functions can then be obtained by running the high fidelity model for the selected training parameter. During the online computation, for any given (untrained) parameter, the snapshots and corresponding POD basis functions can be computed using an interpolation approach (here the RBF method). Finally we have extended our recently developed NIROM technique (see [41]) to generate the P-NIROM, that is, by using the RBF approach to construct a set of hyper-surfaces that represent fluid dynamics of the system. Over the existing non-intrusive ROM for parameterized PDEs proposed in [17], there is no need to solve an auxiliary parabolic linear PDE which is split from the original nonlinear PDE. The P-NIROM technique has been used for model reduction of the parameter Navier–Stokes equations and applied to an unstructured mesh finite element fluid model.

Two numerical examples were chosen to demonstrate the capabilities of the P-NIROM. In the first numerical example, flow past a cylinder was solved, where the P-NIROM was trained upon a small number of parameter points

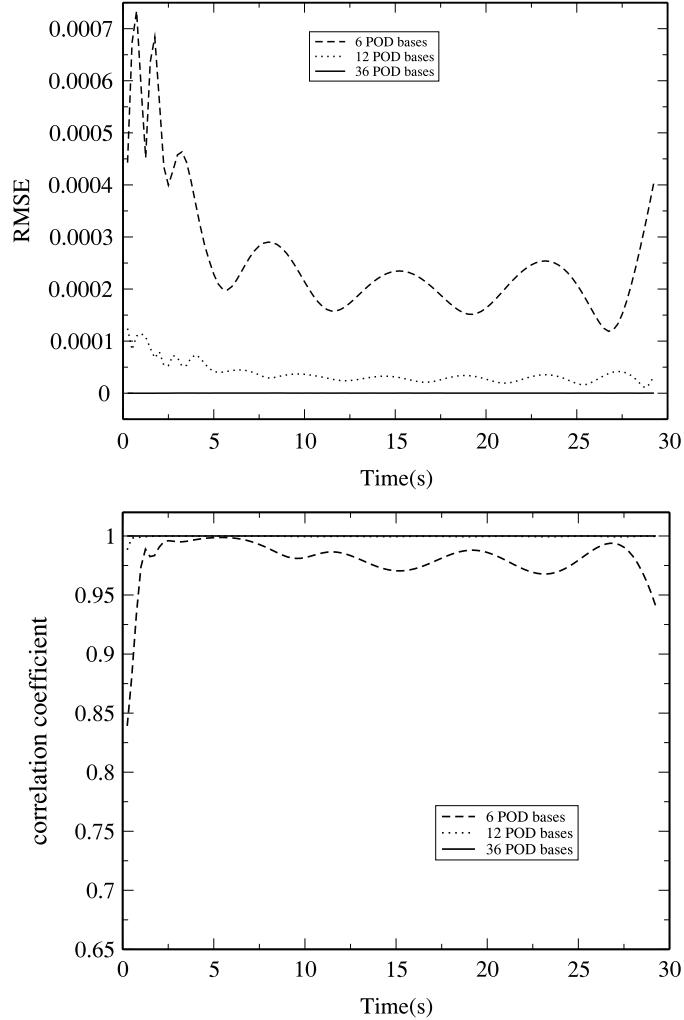


Fig. 19. Lock exchange: the figure shows the root mean squared error (RMSE) and correlation coefficient between the high fidelity model and P-NIROMs with 6, 12 and 36 POD basis functions.

determined by the Smolyak sparse grid and then tested on an untrained parameter point (here, the two dimensional parameter space was constructed by the inlet velocity and viscosity). It is shown that the P-NIROM can retain much of the accuracy of the high fidelity model while the computational time is reduced by three orders of magnitude. In the second example, a lock exchange problem was solved. The prediction capabilities of the NIROM have been evaluated by specifying a new (untrained) initial condition. Again it shows that the problem is well predicted with a significantly reduced computational cost. An error analysis has been undertaken through the RMSE and correlation coefficient of solutions between the high fidelity model and P-NIROM.

This P-NIROM is independent of the source code of the full system, therefore, it is easy to extend to complex applications, such as multi-physics problems, model uncertainty analysis, sensitive analysis, model parameter estimation and control (*e.g.* shape optimization control and closed-loop turbulence control [53]). The applications of the P-NIROM are not limited to the fluid flow applications demonstrated in this paper. In the future, we will apply our model to more complicated time-dependent non-linear PDEs and explore the stability of long-term parametric non-linear dynamical systems. The generalized Lyapunov's direct method [28] can be used to guarantee the long-term boundedness if there is a monotonically attracting ‘trapping region’. The concept of long-term boundedness is linked to the stability analysis of parametric nonlinear PDE systems with respect to the parameters *e.g.* initial and boundary values using the energy method. By analysing the spectrum of eigenvalues and Lyapunov exponents, a

sufficient criterion for long-term boundedness of Galerkin systems can be used to exclude infinite blow-ups of the system state solutions in finite or infinite periods of time [28]. In the near future we will explore solution boundedness and methodologies for interpolating the ROM basis functions over parameter ranges.

Acknowledgements

This work was carried out under funding from Janet Watson scholarship at Department of ESE, Imperial College. Authors would like to acknowledge the support of the UK's Natural Environment Research Council projects (NER/A/S/2003/00595, NE/C52101X/1 and NE/C51829X/1), the Engineering and Physical Sciences Research Council (GR/R60898, EP/I00405X/1 and EP/J002011/1), and the Imperial College High Performance Computing Service. Prof. I.M. Navon acknowledges the support of NSF/CMG grant ATM-0931198. Dr. Xiao acknowledges the support of NSFC grant 11502241. Prof. Pain and Fang are grateful for the support provided by BP Exploration. The authors are grateful for the support of the EPSRC grant: Managing Air for Green Inner Cities (MAGIC)(EP/N010221/1) and the EPSRC MEMPHIS multi-phase flow programme grant (EP/K003976/1).

Appendix

A.1. Error analysis for P-NIROM

This section provides a priori error analysis for P-NIROMs. Let $\mathbf{u}(\mu, \mathbf{x}, t)$ and $\mathbf{u}^{P\text{-NIROM}}(\mu, \mathbf{x}, t)$ denote the full and P-NIROM solutions for any given parameter μ respectively. The error norm is chosen to be Euclidean distance. The error between the full and P-NIROM solution can be estimated by:

$$\begin{aligned}\|\mathbf{u} - \mathbf{u}^{P\text{-NIROM}}\| &= \|(\mathbf{u} - \Phi\Phi^T\mathbf{u}) + (\Phi\Phi^T\mathbf{u} - \Phi\Phi^T\tilde{\mathbf{u}}) + (\Phi\Phi^T\tilde{\mathbf{u}} - \mathbf{u}^{P\text{-NIROM}})\| \\ &\leq \|\mathbf{u} - \Phi\Phi^T\mathbf{u}\| + \|\Phi\Phi^T\mathbf{u} - \tilde{\Phi}\tilde{\Phi}^T\tilde{\mathbf{u}}\| + \|\tilde{\Phi}\tilde{\Phi}^T\tilde{\mathbf{u}} - \mathbf{u}^{P\text{-NIROM}}\|\end{aligned}\quad (24)$$

where $\Phi = (\phi_1, \phi_2, \dots, \phi_M)$ consists of the leading POD basis functions which are obtained using POD-SVD and $\tilde{\mathbf{u}}(\mu, \mathbf{x}, t) = \mathcal{I}(\mu)$ is an optimal approximation of \mathbf{u} which can be obtained by interpolating the solutions over the Smolyak grids (for details, see algorithm 1) and the corresponding POD basis functions $\tilde{\Phi}$ are calculated using POD-SVD based on snapshots $\tilde{\mathbf{u}}$. We can see that the error, $\|\mathbf{u} - \mathbf{u}^{P\text{-NIROM}}\|$, is split into three parts: the POD truncation error, the error of solution snapshots and POD basis functions due to the interpolation over the sparse grids, and the error from the calculation of the P-NIROM solution \mathbf{u}^r .

A.1.1. Truncation error

$\Phi\Phi^T\mathbf{u}$ is an optimal approximation of \mathbf{u} and the error is bounded by $\sqrt{\lambda_{M+1}}$ if M leading POD basis functions are chosen [54], that is,

$$\|\mathbf{u} - \Phi\Phi^T\mathbf{u}\| \leq \sqrt{\lambda_{M+1}}, \quad (25)$$

where λ_{M+1} is the $M + 1$ th eigenvalue of $B^T B$ (where, B is defined in (6)). In P-NIROM, the snapshots and POD basis functions for any given parameter μ over the parameter space are calculated by interpolating them over the parameter sparse grids (see algorithm 1), thus resulting in the interpolation error described below.

A.1.2. Error in calculation of snapshots

The second term in (24) is caused by the interpolation error in calculation of snapshots, which is the interpolation method dependent.

$$\begin{aligned}\|\Phi\Phi^T\mathbf{u} - \tilde{\Phi}\tilde{\Phi}^T\tilde{\mathbf{u}}\| &\leq \|\Phi\Phi^T\mathbf{u} - \tilde{\Phi}\tilde{\Phi}^T\mathbf{u}\| + \|\tilde{\Phi}\tilde{\Phi}^T\mathbf{u} - \tilde{\Phi}\tilde{\Phi}^T\tilde{\mathbf{u}}\| \\ &\leq \|\Phi\Phi^T - \tilde{\Phi}\tilde{\Phi}^T\| \|\mathbf{u}\| + \|\tilde{\Phi}\tilde{\Phi}^T\| \|\mathbf{u} - \tilde{\mathbf{u}}\|.\end{aligned}\quad (26)$$

(a) In the Smolyak sparse grid method, these errors in (26) are bounded by [55]:

$$\begin{aligned}\|\Phi\Phi^T - \tilde{\Phi}\tilde{\Phi}^T\| &\leq C_{d,k} m^{-k} (\log(m))^{(k+1)(d-1)} \|\Phi\Phi^T\|, \\ \|\mathbf{u} - \tilde{\mathbf{u}}\| &\leq C_{d,k} m^{-k} (\log(m))^{(k+1)(d-1)} \|\mathbf{u}\|,\end{aligned}\quad (27)$$

where k is the degree of polynomials, d is the dimensional size of the variable space (for parameter space, $d = P$) $m = m(l + d)$, d (l is the approximation level used in the Smolyak grid) is the number of sparse grids used. Due to the orthonormality of POD basis functions, $\Phi\Phi^T = I$ and $\tilde{\Phi}\tilde{\Phi}^T = I$.

- (b) *In the radial basis function method*, for a given positive constant ρ , we define $K_\rho = \{\mu \in R^P \mid \|\mu - \mu_p\| \leq \rho\}$. It is proven the existence of positive constants $k_\phi, k_{\mathbf{u}} \in \mathcal{N}$ and $C_1, C_2 \in \mathcal{R}$, the local errors in (26) are estimated by [56]:

$$\begin{aligned} \|\Phi\Phi^T - \tilde{\Phi}\tilde{\Phi}^T\| &\leq c_\phi \cdot C_1 \cdot h_\rho^{k_\phi}, \\ \|\mathbf{u} - \tilde{\mathbf{u}}\| &\leq c_{\mathbf{u}} \cdot C_2 \cdot h_\rho^{k_{\mathbf{u}}}, \end{aligned} \quad (28)$$

where c_ϕ and $c_{\mathbf{u}}$ are constants dependent of the radial function ϕ , as well as Φ and \mathbf{u} respectively, and $h_\rho := \max_{\mu \in K_\rho} \min_{1 \leq p \leq P} \|\mu - \mu_p\|$.

A.1.3. Error in calculation of P-NIROM solution/coefficient

The third term in (24) is

$$\begin{aligned} \|\tilde{\Phi}\tilde{\Phi}^T \tilde{\mathbf{u}} - \mathbf{u}^{P-NIROM}\| &= \|\tilde{\Phi}\tilde{\mathbf{u}}^{r,*} - \tilde{\Phi}\mathbf{u}^{r,NIROM}\| \\ &\leq \|\tilde{\Phi}\|_{2,2} \|\tilde{\mathbf{u}}^{r,*} - \tilde{\Phi}\mathbf{u}^{r,NIROM}\|, \end{aligned} \quad (29)$$

where, $\mathbf{u}^{r,*} = \Phi^T \tilde{\mathbf{u}}$ is the POD solution by projecting the full model solution onto the reduced space; $\mathbf{u}^{r,NIROM}$ is the solution from the P-NIROM; $\|\tilde{\Phi}\|_{2,2}$ is defined below:

$$\|\tilde{\Phi}\|_{2,2} = \sup_x \|\tilde{\Phi}x\| / \|x\|. \quad (30)$$

Taking into account $\Phi\Phi^T = I$ (where I is an identity matrix), thus $\|\tilde{\Phi}\|_{2,2} = 1$.

- (a) *In the Smolyak sparse grid method*, the error between $\mathbf{u}^{r,*}$ and $\mathbf{u}^{r,NIROM}$ is bounded by [55]:

$$\|\tilde{\mathbf{u}}^{r,*} - \tilde{\Phi}\mathbf{u}^{r,NIROM}\| \leq C_{d,k}^r m^{r,-k^r} (\log(m^r))^{(k^r+1)(d^r-1)} \|\tilde{\mathbf{u}}^{r,*}\|, \quad (31)$$

where k^r is the degree of polynomials, d^r is the dimensional size of variable space, the $m^r = m^r(l^r + d^r, d^r)$ (l^r is the level used in the Smolyak grid) is the number of sparse grid used.

- (b) *In the radial basis function method*, for a given positive constant $\rho_{\mathbf{u}^r}$, we define $K_{\rho_{\mathbf{u}^r}} = \{\mathbf{u}^{r,NIROM} \in R^{N_T} \mid \|\mathbf{u}^{r,NIROM} - \mathbf{u}_{N_t}^{r,NIROM}\| \leq \rho_{\mathbf{u}^r}\}$ (here $\{\mathbf{u}_{N_t}\}$ are training points, N_T is the number of the training points). It is proven the existence of positive constants $k_{\mathbf{u}^r} \in \mathcal{N}$ and $C_{\mathbf{u}^r} \in \mathcal{R}$, the local errors in (29) are estimated by

$$\|\tilde{\mathbf{u}}^{r,*} - \tilde{\Phi}\mathbf{u}^{r,NIROM}\| \leq c_{\mathbf{u}^r} \cdot C_{\mathbf{u}^r} \cdot h_{\rho_{\mathbf{u}^r}}^{k_{\mathbf{u}^r}}, \quad (32)$$

where $c_{\mathbf{u}^r}$ is constant dependent on the radial function ϕ and \mathbf{u} , and $h_{\rho_{\mathbf{u}^r}} := \max_{\mu \in K_{\rho_{\mathbf{u}^r}}} \min_{1 \leq p \leq P} \|\mu - \mu_p\|$.

References

- [1] M. Alotaibi, V.M. Calo, Y. Efendiev, J. Galvis, M. Ghommem, Global local nonlinear model reduction for flows in heterogeneous porous media, *Comput. Methods Appl. Mech. Engrg.* 292 (2015) 122–137. Special Issue on Advances in Simulations of Subsurface Flow and Transport (Honoring Professor Mary F. Wheeler).
- [2] A. Corigliano, M. Dossi, S. Mariani, Model order reduction and domain decomposition strategies for the solution of the dynamic elastic plastic structural problem, *Comput. Methods Appl. Mech. Engrg.* 290 (2015) 127–155.
- [3] F. Fang, T. Zhang, D. Pavlidis, C. Pain, A. Buchan, I.M. Navon, Reduced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons, *Atmos. Environ.* 96 (2014) 96–106.
- [4] F. Fang, C. Pain, I.M. Navon, A. Elsheikh, J. Du, D. Xiao, Non-Linear Petrov-Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods, *J. Comput. Phys.* 234 (2013) 540–559.
- [5] R. Stefanescu, I.M. Navon, POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model, *J. Comput. Phys.* 237 (2013) 95–114.
- [6] R. Stefanescu, A. Sandu, I.M. Navon, Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations, *Int. J. Numer. Methods Fluids* 76 (8) (2014) 497–521.

- [7] K. Hoang, Y. Fu, J. Song, An hp-proper orthogonal decomposition moving least squares approach for molecular dynamics simulation, *Comput. Methods Appl. Mech. Engrg.* 298 (2016) 548–575.
- [8] D. Daescu, I.M. Navon, A dual-weighted approach to order reduction in 4D-Var data assimilation, *Mon. Weather Rev.* 136 (3) (2008) 1026–1041.
- [9] Y. Cao, I.M. Navon, Z. Luo, A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition, *Int. J. Numer. Methods Fluids* 53 (2007) 1571–1583.
- [10] S. Giere, T. Iliescu, V. John, D. Wells, [SUPG] reduced order models for convection-dominated convection diffusion reaction equations, *Comput. Methods Appl. Mech. Eng.* 289 (2015) 454–474.
- [11] R. Ștefănescu, A. Sandu, I.M. Navon, POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation, *J. Comput. Phys.* 295 (2015) 569–595.
- [12] Y. Cao, J. Zhu, I.M. Navon, Z. Luo, A reduced order approach to four dimensional variational data assimilation using proper orthogonal decomposition, *Int. J. Numer. Methods Fluids* 53 (2007) 1571–1583.
- [13] A. Buchan, A. Calloo, M. Goffin, S. Dargaville, F. Fang, C. Pain, I.M. Navon, A POD reduced order model for resolving angular direction in neutron/photon transport problems, *J. Comput. Phys.* 296 (2015) 138–157.
- [14] N. Ardjmandpour, C. Pain, F. Fang, A. Buchan, J. Singer, M. Player, X. Xu, I.M. Navon, J. Carter, Reduced order borehole induction modelling, *Int. J. Comput. Fluid Dyn.* 28 (3–4) (2014) 140–157.
- [15] M. Diez, E.F. Campana, F. Stern, Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion, *Comput. Methods Appl. Mech. Eng.* 283 (2015) 1525–1544.
- [16] F. Fang, C. Pain, I.M. Navon, D. Xiao, An efficient goal based reduced order model approach for targeted adaptive observations, *Int. J. Numer. Methods Fluids* (2016).
- [17] C. Audouze, F.D. Vuyst, P.B. Nair, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, *Numer. Methods Partial Differ. Equ.* 29 (5) (2013) 1587–1628.
- [18] P. Benner, S. Gugercin, K. Willcox, A survey of model reduction methods for parametric systems, preprint MPIMD/13-14, Max Planck Institute Magdeburg, Aug. 2013. Available from <http://www.mpi-magdeburg.mpg.de/preprints/>.
- [19] B. Haasdonk, M. Dihlmann, M. Ohlberger, A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space, *Math. Comput. Model. Dyn. Syst.* 17 (4) (2011) 423–442.
- [20] B. Haasdonk, M. Ohlberger, Space-adaptive reduced basis simulation for time-dependent problems, in: Proceedings of the Vienna International Conference on Mathematical Modelling, 2009.
- [21] A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92, Springer, 2015.
- [22] M. Ohlberger, G. Rozza, A reduced basis method for evolution schemes with parameter-dependent explicit operators, *Electron. Trans. Numer. Anal.* 32 (2008) 145–161.
- [23] M. Drohmann, B. Haasdonk, M. Ohlberger, Reduced basis model reduction of parametrized two phase flow in porous media, *IFAC Proceedings Volumes* 45 (2) (2012) 722–727.
- [24] B. Peherstorfer, S. Zimmer, H.-J. Bungartz, Model reduction with the reduced basis method and sparse grids, in: *Sparse Grids and Applications*, Springer, 2012, pp. 223–242.
- [25] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Arch. Comput. Methods Eng.* 15 (3) (2008) 229–275.
- [26] C. Han, Blackbox Stencil Interpolation Method for Model Reduction (Master's thesis), Massachusetts Institute of Technology, 2012.
- [27] 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.
- [28] M. Schlegel, B.R. Noack, On long-term boundedness of Galerkin models, *J. Fluid Mech.* 765 (2015) 325–352.
- [29] J. Osth, B.R. Noack, S. Krajnovi, D. Barros, J. Bore, On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body, *J. Fluid Mech.* 747 (2014) 518–544.
- [30] C.B.-M.K. Carlberg, C. Farhat, Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations, *Internat. J. Numer. Methods Engrg.* 86 (2011) 155–181.
- [31] M.S.Y. Chu, J. Hahn, State-preserving nonlinear model reduction procedure, *Chem. Eng. Sci.* 66 (2011) 3907–3913.
- [32] K. Willcox, A. Megretski, Model reduction for large-scale linear applications, in: Proc. of 13th IFAC Symposium on System Identification, Rotterdam, Netherlands, 2003, pp. 1431–1436.
- [33] A.J. Feriedoun Sabethghadam, α regularization of the POD-Galerkin dynamical systems of the Kuramoto-Sivashinsky equation, *Appl. Math. Comput.* 218 (2012) 6012–6026.
- [34] D. Xiao, F. Fang, J. Du, C. Pain, I.M. Navon, A.G. Buchan, A. ElSheikh, G. Hu, Non-Linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair, *Comput. Methods Appl. Mech. Engrg.* 255 (2013) 147–157.
- [35] S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.* 32 (2010) 2737–2764.
- [36] M. Barrault, Y. Maday, N. Nguyen, A. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Acad. Sci., Paris* 339 (2004) 667–672.
- [37] J. Du, F. Fang, C.C. Pain, I.M. Navon, J. Zhu, D.A. Ham, POD reduced-order unstructured mesh modeling applied to 2D and 3D fluid flow, *Comput. Math. Appl.* 65 (3) (2013) 362–379.
- [38] F. Fang, C. Pain, I.M. Navon, M. Piggott, G. Gorman, P. Allison, A. Goddard, Reduced-order modelling of an adaptive mesh ocean model, *Int. J. Numer. Methods Fluids* 59 (8) (2009) 827–851.
- [39] D. Xiao, F. Fang, A.G. Buchan, C. Pain, I.M. Navon*, J. Du, G. Hu, Non-linear model reduction for the Navier–Stokes equations using residual DEIM method, *J. Comput. Phys.* 263 (2014) 1–18.
- [40] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, *J. Comput. Phys.* 242 (2013) 623–647.

- [41] D. Xiao, F. Fang, A. Buchan, C. Pain, I.M. Navon, A. Muggeridge, Non-intrusive reduced order modelling of the Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 293 (2015) 552–541.
- [42] 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.
- [43] R. Noori, A. Karbassi, H. Mehdizadeh, M. Vesali-Naseh, M. Sabahi, A framework development for predicting the longitudinal dispersion coefficient in natural streams using an artificial neural network, *Environ. Prog. Sustainable Energy* 30 (3) (2011) 439–449.
- [44] B.R. Noack, M. Morzynski, G. Tadmor, *Reduced-Order Modelling for Flow Control*, Vol. 528, Springer, 2011.
- [45] C. Audouze, F. De Vuyst, P. Nair, Reduced-order modeling of parameterized pdes using time–space-parameter principal component analysis, *Internat. J. Numer. Methods Engrg.* 80 (8) (2009) 1025–1057.
- [46] D. Xiao, P. Yang, F. Fang, J. Xiang, C. Pain, I.M. Navon, Non-intrusive reduced order modeling of fluid–structure interactions, *Comput. Methods Appl. Mech. Engrg.* 303 (2016) 35–54.
- [47] D. Xiao, Z. Lin, F. Fang, C. Pain, I.M. Navon, P. Salinas, A. Muggeridge, Non-intrusive reduced order modeling for multiphase porous media flows using smolyak sparse grids, *Int. J. Numer. Methods Fluids* (2016).
- [48] S.A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, in: *Dokl. Akad. Nauk SSSR*, 4 (240–243) (1963) 123.
- [49] K.L. Judd, L. Maliar, S. Maliar, R. Valero, Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain, *J. Econ. Dyn. Control* 44 (2014) 92–123.
- [50] T. Gerstner, M. Griebel, Numerical integration using sparse grids, *Numer. Algorithms* 18 (3–4) (1998) 209–232.
- [51] C. Pain, M. Piggott, A. Goddard, F. Fang, G. Gorman, D. Marshall, M. Eaton, P. Power, C. De Oliveira, Three-dimensional unstructured mesh ocean modelling, *Ocean Modelling* 10 (1) (2005) 5–33.
- [52] T. Benjamin, Gravity currents and related phenomena, *J. Fluid Mech.* 31 (1968) 209–248.
- [53] S.L. Brunton, B.R. Noack, Closed-loop turbulence control: Progress and challenges, *Appl. Mech. Rev.* 67 (5) (2015) 050801.
- [54] Z. Luo, J. Zhu, R. Wang, I.M. Navon, Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical pacific ocean reduced gravity model, *Comput. Methods Appl. Mech. Engrg.* 196 (2007) 4184–4195.
- [55] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, *Adv. Comput. Math.* 12 (4) (2000) 273–288.
- [56] Z.-m. Wu, R. Schaback, Local error estimates for radial basis function interpolation of scattered data, *IMA J. Numer. Anal.* 13 (1) (1993) 13–27.