Computers and Fluids 177 (2018) 69–77

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Parameterised non-intrusive reduced order methods for ensemble Kalman filter data assimilation

D. Xiao[a,c,1], J. Du[b,*], F. Fang[a,c,1], C.C. Pain[a,c,1], J. Li[b]

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

[b] International Center for Climate and Environment Sciences, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, 100029, China

[c] Data assimilation lab, Data Science Institute, Imperial College London, UK

ARTICLE INFO

Article history:
Received 6 May 2018
Revised 18 September 2018
Accepted 3 October 2018
Available online 4 October 2018

Keywords:
Parameterised NIROM
RBF
POD
Enkf

ABSTRACT

This paper presents a novel Ensemble Kalman Filter (EnKF) data assimilation method based on a parameterised non-intrusive reduced order model (P-NIROM) which is independent of the original computational code. EnKF techniques involve the expensive calculations of ensembles. In this work, the recently developed P-NIROM Xiao et al. [40] is incorporated into EnKF to speed up the ensemble simulations. A reduced order flow dynamical model is generated from the solution snapshots, which are obtained from a number of the high fidelity full simulations over the specific parametric space R^P. The varying parameter is the background error covariance $\sigma \in R^P$. Using the Smolyak sparse grid method, a set of parameters in the Gaussian probability density function is selected as the training points. The proposed method uses a two-level interpolation method for constructing the P-NIROM using a Radial Basis Function (RBF) interpolation method. The first level interpolation approach is used for generating the solution snapshots and POD basis functions for any given background error covariance while the second level interpolation approach for forming a set of hyper-surfaces representing the reduced system.

The EnKF in combination with P-NIROM (P-NIROM-EnKF) has been implemented within an unstructured mesh finite element ocean model and applied to a three dimensional wind driven circulation gyre case. The numerical results show that the accuracy of ensembles and updated solutions using the P-NIROM-EnKF is maintained while the computational cost is significantly reduced by several orders of magnitude in comparison to the full-EnKF.

1. Introduction

Data assimilation is a procedure that incorporates the observation data into numerical models in an optimal way, thus improving the accuracy of numerical results. The ensemble Kalman filter (EnKF) is one commonly used data assimilation method since it has relatively simpler conception and implementation in comparison to 4DVAR data assimilation methods [2]. It was firstly introduced by Evensen [11], and then became popular in a number of geosciences research fields, such as fluids [27], atmosphere [21], ocean [20], reservoir modelling [12], multiphase flows [7], biogeochemical model [32], wave simulation [23] and subsurface contaminant model [25]. However, EnKF requires a large number of realisations to ensure its convergence. This limits its application to complex problems since the computational cost of each realisation is expensive in high fidelity full simulations. The approach to tackling this issue is to reduce the dimensional size of the original high fidelity model using reduced order modelling (ROM) techniques.

ROM is a rapidly growing discipline, with significant potential advantages in: interactive use, emergency response, ensemble calculations, and data assimilation [1,8,9,17,29]. ROM is expected to play a major role in facilitating real-time predictions. Proper Orthogonal Decomposition (POD) is a widely used method to construct a ROM. Using the POD method, a set of optimal POD bases (orthogonal vectors) is constructed via a truncated singular value decomposition (SVD) method. Then it is used to formulate a reduced dynamical system that describes the main flow features. Due to the optimality of convergence in terms of the kinetic energy of the POD basis functions, the dominant components of a large dimensional process can be captured with only a small num-

* Corresponding author.
E-mail address: dujuan10@mail.iap.ac.cn (J. Du).
[1] http://amcg.ese.imperial.ac.uk

https://doi.org/10.1016/j.compfluid.2018.10.006

ber of POD basis functions, thus reducing the CPU time by several orders of magnitude.

Recently, ROM has been used to perform data assimilation. There are a number of work combining ROM and data assimilation methods. He et al. applied ROMs for improving data assimilation under the framework of an EnKF [19]. In the work of [14], Etros et al. used the balanced truncation based ROM to obtain the state estimators. Lin et al. used the truncated discrete cosine transform and non-linear extension of POD to reduce the spatial parameters and the dynamic states [24]. Tian et al. presented an ensemble 4D-VAR data assimilation method based on POD [31]. Pagani et al. proposed a reduced basis EnKF method for state identification in large-scale dynamical systems [30].

However, those data assimilation methods are combined with intrusive ROM (IROM). The intrusive ROM is dependent on original partial differential equations (PDEs) and source codes. In most cases, modifications are required in order to maintain the source code and extend the applications. These modifications are sometimes impossible in commercial software [18] since the source code of commercial software is not available. In addition, the IROM has instability and non-linear inefficiency issues [26,28,33]. Those instability and non-linear inefficiency issues could affect the accuracy and efficiency of generating ensembles although some efforts have been made to improve the stability and the non-linearity efficiency of ROMs [3,5,6,10,13,15,16,22,34,37,38,41]. In order to tackle these issues in intrusive ROMs, recently, non-intrusive reduced order models (NIROMs) have been introduced [35,36,39]. However, to the best of our knowledge, very little work can be found addressing non-intrusive model reduction for data assimilation, especially the EnKF data assimilation method.

In this paper, we present a EnKF method based on a recently developed parameterised non-intrusive reduced order model (P-NIROM) [40] to tackle the computational intensive ensembles calculation of the high fidelity full model. The P-NIROM is an efficient method for reducing the dimensionality of parameterised partial differential equations (P-PDEs) in a non-intrusive way. The EnKF based on P-NIROM (P-NIROM-EnKF) has been developed within an unstructured mesh finite element ocean model. The performance of the P-NIROM-EnKF has been evaluated by comparing with the EnKF based on the high fidelity full model (full-EnKF) using a three dimensional ocean gyre problem.

The structure of the paper is as follows: Section 2 introduces the ENKF method; Section 3 describes the general parametric non-intrusive model reduction methods (P-NIROM) for efficient EnKF; Section 4 derives the new efficient EnKF method using the P-NIROM; Section 5 illustrates the performance of the P-NIROM-EnKF method by applying it into a gyre problem. Finally in Section 6, summary and conclusions are presented.

2. Ensemble Kalman filter

This section provides a brief description of the EnKF. Given a parameterised discrete dynamical model and a $n-$ dimensional vector of the model state $\mathbf{u}$

$$\mathbf{u}_{t_j} = \mathcal{M}_{t_0,t_j}(\mathbf{u}_{t_0}, \mu), \tag{1}$$

where $\mathbf{u}_{t_0}$ denote the background (initial state) at time level t_0, $\mathbf{u}_{t_j}$ is the numerical forecast at time level t_j, $\mu \in \mathcal{R}^p$ (constructing a P dimensional parameter space), and $\mathcal{M}_{t_0,t_j}$ is the model operator from time level t_0 to t_j. Taking into account observational data at time level t_j, the EnKF is used to improve the accuracy of the model forecast at time level t_j. We assume that the model state has the Gaussian probability distribution. By adding the Gaussian noise $\left(\mathcal{G}(0,\sigma_i^2), i \in (1,\ldots,N_e)\right)$ to the background $\mathbf{u}_{t_0}$ and running the simulations from time level t_0 to t_j, the ensemble matrix (containing N_e ensemble members) can be obtained:

$$\mathbf{U}_{t_j} = \left(\mathbf{u}_{t_j,1}, \mathbf{u}_{t_j,2}, \cdots, \mathbf{u}_{t_j,N_e}\right) \in R^{n\times N_e}, \tag{2}$$

where $\mathbf{U}_{t_j}$ is an $n \times N_e$ matrix with N_e ensemble members. The anomaly ensemble matrix is

$$\mathbf{U}'_{t_j} = \frac{1}{\sqrt{(N_e-1)}}\left(\mathbf{U}_{t_j} - \overline{\mathbf{U}}_{t_j}\right), \tag{3}$$

where $\overline{\mathbf{U}}_{t_j}$ is the ensemble mean matrix at time level t_j. The ensemble error covariance matrix is therefore written:

$$\mathbf{R}_{e,t_j} = \mathbf{U}'_{t_j}\mathbf{U}'^{T}_{t_j}, \tag{4}$$

The observation data is $\mathbf{d}_{t_j}$ at time level t_j and has a uncertainty of δ_{t_j}. Then the perturbed observation data $\mathbf{d}_{t_j}$ can be represented by the data vector $\mathbf{d}_{t_j}$ plus a random vector from a normal distribution $\delta_{t_j} = \mathcal{G}_{d,t_j}(0, \mathbf{R}_{d,t_j})$, $\mathbf{R}_{d,t_j} = \delta_{t_j}\delta_{t_j}^T$ is the observation error covariance matrix, and

$$\mathbf{d}_{t_j} = \mathbf{d}_{t_j} + \delta_{t_j}. \tag{5}$$

The analysis ensemble member $\mathbf{u}^a_{i,t_j}$ can be given by:

$$\mathbf{u}^a_{i,t_j} = \mathbf{U}_{t_j} + \mathbf{R}_{e,t_j}\mathbf{H}^T\left(\mathbf{H}\mathbf{R}_{e,t_j}\mathbf{H}^T + \mathbf{R}_{d,t_j}\right)^{-1}\left(\mathbf{d}_{t_j} - \mathbf{H}\mathbf{u}_{i,t_j}\right),\ i \in (1,\ldots,N_e), \tag{6}$$

where $\mathbf{R}_{e,t_j}$ is the background error covariance matrix, $\mathbf{H}$ is an operator which interpolates the model solutions onto the observation space. The Kalman gain is $\mathbf{K}_{t_j} = \mathbf{R}_{e,t_j}\mathbf{H}^T(\mathbf{H}\mathbf{R}_{e,t_j}\mathbf{H}^T + \mathbf{R}_{d,t_j})^{-1}$. The analysis ensemble error covariance matrix $\mathbf{U}_{t_j}$ is obtained:

$$\mathbf{U}^a_{e,t_j} = \left(\mathbf{I} - \mathbf{K}_{t_j}\mathbf{H}\right)\mathbf{R}_{e,t_j}. \tag{7}$$

3. Parameterised non-intrusive modelling techniques for efficient EnKF

This section will introduce a P-NIROM for efficient ensemble calculations in (2). The reduced order modelling methods based on POD are popular and powerful techniques for circumventing the intensive computational burden in large complex numerical simulations. POD is capable of representing large systems using a few number of optimal basis functions. In POD reduced order modelling, the variable state in (1) can be expressed as an expansion of the POD basis functions $\Phi = (\Phi_1, \ldots, \Phi_M)$:

$$\mathbf{u} = \Phi\mathbf{u}^r, \tag{8}$$

where $\mathbf{u}^r = (u^r_1 \ldots, u^r_m, \ldots, u^r_M)^T$ $(1 \le m \le M) \in \mathcal{R}^M$ is the reduced state variable vector (the superscript r indicates the variable associated with the reduced order model) to be determined over the reduced space. Projecting (1) from the $n-$ dimensional space onto the $M-$ dimensional reduced space ($M << n$), yields:

$$\Phi^T\mathbf{u}_{t_j} = \Phi^T\mathcal{M}_{t_0,t_j}\left(\Phi\mathbf{u}^r_{t_0}, \mu\right). \tag{9}$$

The parameterised reduced order model can thus be written as:

$$\mathbf{u}^r_{t_j} = \mathcal{M}^r_{t_0,t_j}\left(\Phi\mathbf{u}^r_{t_0}, \mu\right), \tag{10}$$

where $\mathcal{M}^r_{t_0,t_j}$ is the model operator over the reduced space from time level t_0 to t_j. Eqs. (8) and (10) can be used for efficient ensemble calculations where the CPU time can be reduced by a number orders of magnitude. In this work, the parameter set μ in (10) is the varying Gaussian noise in inputs (initial state) which are used to drive the ensemble simulations. A more recently developed non-intrusive method [40] is proposed to construct the parameterised reduced order model in (10). The P-NIROM [40] is capable of predicting problems with unseen or different parameters (for example, boundary conditions and initial conditions). It is also non-intrusive and independent of the computational source code.

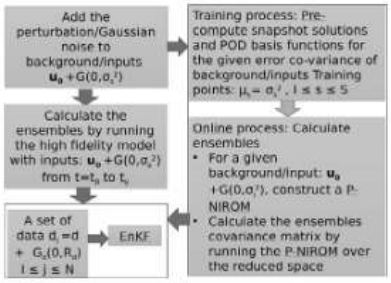


Fig. 1. Traditional EnKF VS P-NIROM-EnKF. The blue represents the same procedure with traditional EnKF; The green indicates the calculation of ensembles using the P-NIROM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

3.1. P-NIROM For calculation of ensemble covariance matrix

Here, we consider the parameter μ in (1) is the state covariance $\mu = \sigma^2$ in the Gaussian noise $\mathcal{G}(0,\sigma^2)$. For a given $\mu_s \in \mathcal{R}^p$ $(1 \le s \le S)$, by adding the Gaussian noise $\mathcal{G}(\mu_i = \sigma_i^2)$ to the initial state/inputs at time level t_0, the ensemble $\mathbf{u}_{t_j,i}$ in (2) can be obtained by running the full order model (1) from time level t_0 to t_j. Our aim is to generate a robust P-NIROM (10), an approximation to the original model (1) with significantly reduced degrees of freedom. The whole procedure of P-NIROM involves the offline (training) and online stages. The offline procedure involves calculation of POD basis functions and snapshots based on the selected state error covariances. The online procedure involves constructing the P-NIROM and using it for calculating the ensemble error covariance matrix for any given background.

Algorithm 1: Offline training process: Constructing the hyper-surface function for calculation snapshot solutions and POD basis functions over the selected range of error covariances of background states/inputs.

(1) Select a range of the error covariance of background state variables or inputs R^p;
(2) Choose a set of error covariances $\mu_s = \sigma_s^2$ $s \in (1,\dots,S)$ as the training points;
(3) Run the high fidelity full model for the given training inputs, $\mathbf{u}_{t_0} + \mathcal{G}(0,\mu_s)$, a set of training snapshots can be obtained;
for $j = 1$ *to* N_t **do**
$\mathbf{u}_{\mu_s,t_j} = \mathcal{M}_{t_{j-1},t_j}(\mathbf{u}_{t_{j-1}},\mu_s)$
endfor
(4) Calculate the POD basis functions using SVD:
$\Phi_{\mu_s} = \sum_{j=1}^{N_t} \psi_{\mu_s,j}\mathbf{u}_{\mu_s,j}$

3.1.1. Calculation of POD basis functions and snapshots based on the selected state error covariances

At the training stage, a set of error covariances (training points) of the initial state/inputs, $\mu_s = \sigma_s^2, s \in (1,\dots,S)$, is chosen. By running the high fidelity full model (1) from time level t_0 to t_{N_t}, a set of solution snapshots $(\mathbb{U}(\mu_s))$, depending on the selected error covariance of the initial state or inputs (μ_s), can be obtained:

$$\mathbb{U}(\mu_s) = \left(\mathbf{u}_{\mu_s,t_1},\dots,\mathbf{u}_{\mu_s,t_j},\dots,\mathbf{u}_{\mu_s,t_{N_t}}\right), \quad 1 \le s \le S, \quad 1 \le j \le N_t \tag{11}$$

subject to

$$\mathbf{u}_{\mu_s,t_j} = \mathcal{M}_{t_0,t_j}(\mathbf{u}_{t_0},\mu_s). \tag{12}$$

For any training point (μ_s), using the singular value decomposition for $\mathbb{C}_{\mu_s} = \mathbb{U}(\mu_s)^T\mathbb{U}(\mu_s)$, one obtains the singular vectors $(\psi_{\mu_s,1},\dots,\psi_{\mu_s,n_t})$ with the singular values $(\lambda_{\mu_s,1},\dots,\lambda_{\mu_s,N_t})$. An optimal set of POD basis functions can be written as a linear combinations of the snapshots:

$$\Phi_{\mu_s} = \sum_{j=1}^{N_t} \psi_{\mu_s,j}\mathbf{u}_{\mu_s,j}; \quad 1 \le j \le N_t, \tag{13}$$

where the POD basis functions $\Phi_{\mu_s} = (\Phi_{\mu_s,1},\dots,\Phi_{\mu_s,N_t})$ are optimised to maximise:

$$\frac{1}{N_t}\sum_{j=1}^{N_t} \left| < \mathbf{u}_{\mu_s,j}, \Phi_{\mu_s,j} >_{L^2} \right|^2, \tag{14}$$

subject to

$$\sum_{j=1}^{N_t} | < \Phi_{\mu_s,j}, \Phi_{\mu_s,j} >_{L^2} |^2 = 1, \tag{15}$$

Algorithm 2: Constructing the P-NIROM and calculating the ensemble covariances.

(1) Choose the error covariances μ_s, $1 \le s \le S$;
(2) Generate the initial ensembles by adding the error noise to the basic background $\mathbf{u}_{t_0}$ or inputs $(\mathbf{u}_{t_0} + \mathcal{G}(0,\mu_s))$;
(3) Construct a P-NIROM for the given inputs $(\mathbf{u}_{t_0} + \mathcal{G}(0,\mu_s))$
for $s = 1$ *to* S **do**
(i) Calculate the snapshot solutions $\mathbb{U}_{\mu_s}$ for the given inputs $\mathbf{u}_{t_0} + \mu_s$ using the RBF interpolation (16);
(ii) Calculate the POD basis functions from the snapshots $\mathbb{U}_{\mu_s}$ using SVD;
(iii) Construct the P-NIROM:

$$\mathbf{u}^r_{\mu_s,m,t_j} = f_{\mu_s,m,t_j}\left(\mathbf{u}^r_{\mu_s,1,t_j-1},\dots,\mathbf{u}^r_{\mu_s,m,t_j-1},\dots,\mathbf{u}^r_{\mu_s,M,t_j-1}\right), m \in (1,\dots,M) \tag{23}$$

endfor
(4) Calculate the ensembles $\mathbf{U}_{t_j}$ in (2) using the P-NIROM:
for $j = 1$ *to* N **do**
(a) Initialisation: $\mathbf{u}_{t_0} + \mu_j$
(b) Calculate the ensembles $\mathbf{U}^r_{t_j}$ at time level t_j over the reduced space by running (23) from time level t_0 to t_j:
for $t = t_1$ *to* t_j **do**
for $m = 1$ *to* M **do**
(i) Assign a complete set of the reduced solution $\mathbf{u}^r_{\mu_j,t_j} = (\mathbf{u}^r_{\mu_j,1,t_j-1},\dots,\mathbf{u}^r_{\mu_j,M,t_j-1})$ at previous time level $t_j - 1$ into the hyper-surface $f_{\mu_j,m}$:

$$f_{\mu_j,m,t_j} \leftarrow (\mathbf{u}^r_{\mu_j,1,t_j-1},\dots,\mathbf{u}^r_{\mu_j,m,t_j-1},\dots,\mathbf{u}^r_{\mu_j,M,t_j-1})$$

(ii) Calculate $\mathbf{u}^r_{\mu_j,m,t_j}$t at the current time level t_j using (23):
endfor
endfor
(c) Obtain the approximation of ensembles at the current time level t_j by projecting $\mathbf{u}^r_{\mu_j,t_j}$ onto the full space using:

$$\mathbf{u}_{\mu_j,t_j} = \sum_{m=1}^{M} \mathbf{u}^r_{\mu_j,m,t_j-1}\Phi_{\mu_j,m}$$

endfor

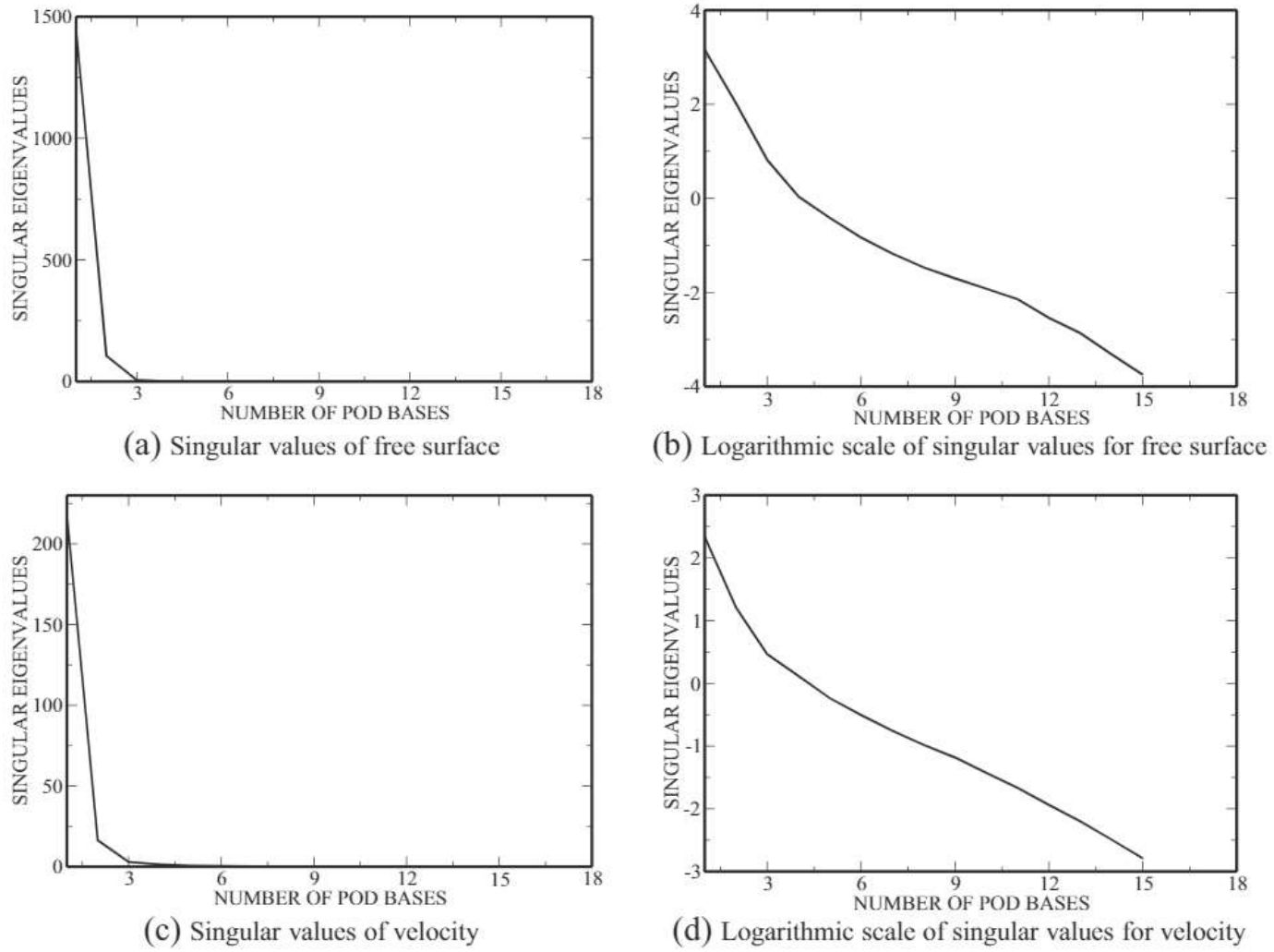


Fig. 2. Singular values and logarithmic scale of singular values for free surface and velocity.

where $< \cdot, \cdot >_{L^2}$ is the canonical inner product in L^2 norm. The j^{th} eigenvalue is a measure of the kinetic energy transferred within the N_t basis functions. The offline training process of snapshot solutions and POD basis functions is summarised in Algorithm (1).

From the training data in (11) and (13), one can now calculate the snapshots and POD basis functions for any given background or inputs $\mathbf{u}_{t_0} + \mathcal{G}(0, \sigma^2)$ using an interpolation function described below.

3.1.2. Construction of P-NIROM and calculation of ensembles and error covariances

After constructing the P-NIROM, one can use it for any new background error covariances, which is treated as the online process. During this online process, the P-NIROM constructs a set of interpolation functions for calculating snapshot solutions and POD basis functions for any new background error covariances.

In this work, the radial basis functions (RBF) interpolation method [4] is used for constructing a set of interpolation functions for the snapshot solutions ($\mathbb{U}(\mu_s)$). In the work of [40], we presented a general P-NIROM. In that work, we construct a set of interpolation functions for the parameter space using the RBF interpolation method. The RBF is a function that its value depends on the distance from some other scatter sample data points or origin. The RBF interpolation method constructs an interpolation function through a number of random sample data points. For any given new background error covariance $\mu \in R^p$, the snapshot solutions $\mathbb{U}(\mu) = (\mathbf{u}_{t_1,\mu}, \ldots, \mathbf{u}_{t_j,\mu}, \ldots, \mathbf{u}_{t_{N_t},\mu})$ can be obtained:

$$\mathbb{U}(\mu) = \mathcal{I}_{RBF}(\mathbb{U}(\mu_1), \ldots, \mathbb{U}(\mu_s), \ldots, \mathbb{U}(\mu_S)), \tag{16}$$

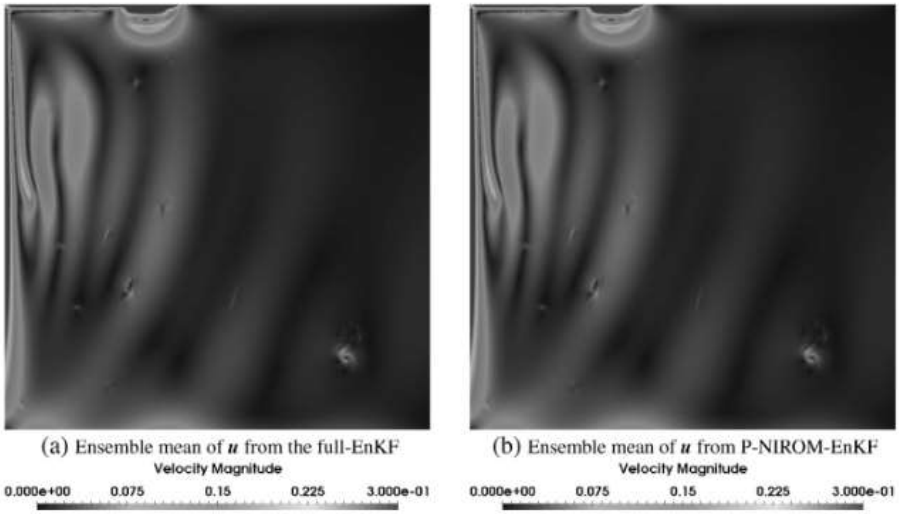
where $\mathcal{I}_{RBF}$ is the RBF interpolation function and has the form of,

$$\mathcal{I}_{RBF} = \sum_{p=1}^{P} w_p \phi(\|\mu - \mu_p\|), \tag{17}$$

where $\mathcal{I}_{RBF}$ can be approximated by a summation of P RBFs ϕ. Each RBF is associated with a different centre point μ_p, and weighted by a coefficient w_p. P denotes the number of sample data points. There are a number of RBFs that can be chosen such as inverse quadratic, multi-quadric, Gaussian, inverse multi-quadric or plate spline. The Gaussian RBF ($\phi(r) = e^{-(r/\sigma)^2}$, r being a radius and σ being a shape parameter) is chosen in this work. The weights $\mathbf{w} = (w_1, \ldots, w_p, \ldots, w_p)^T$ can be calculated by solving Eq. (18),

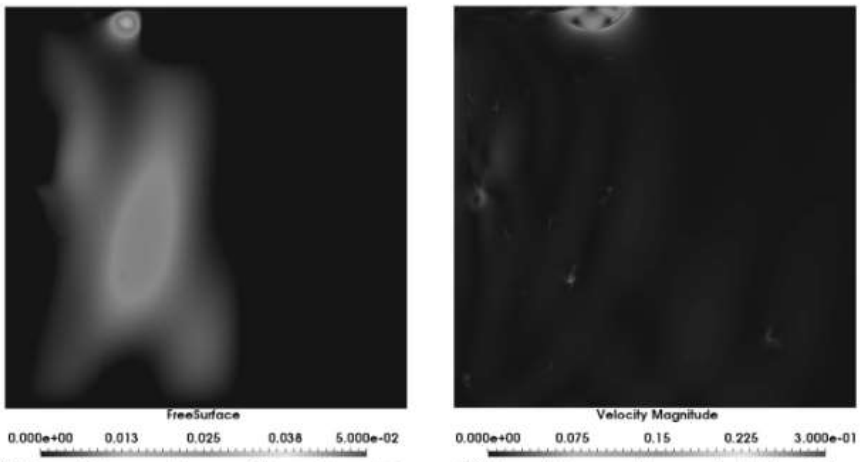
$$A\mathbf{w} = \mathbf{b}, \tag{18}$$

where $\mathbf{b}$ is a vector containing target functional values (snapshots values here) associated with the parameter μ, and matrix A has a



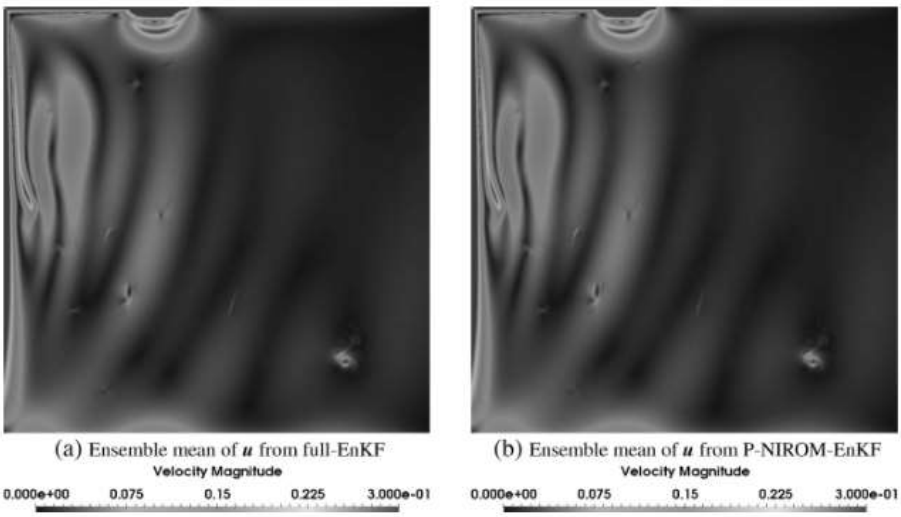
(a) Ensemble mean of $\boldsymbol{u}$ from the full-EnKF (b) Ensemble mean of $\boldsymbol{u}$ from P-NIROM-EnKF

Fig. 3. The ensemble mean of velocity solutions using the (a) full-EnKF and (b) P-NIROM-EnKF at day 10.



(a) Error of the ensemble mean of free surface solutions (b) Error of the ensemble mean of velocity solutions

Fig. 4. The error of the ensemble mean of free surface and velocity solutions (at day 10) using P-NIROM-EnKF.

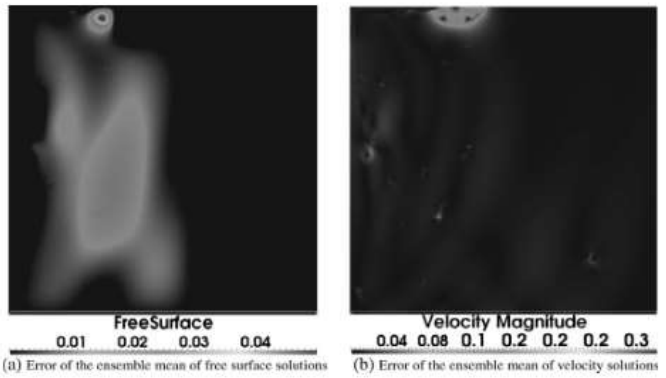


(a) Ensemble mean of $\boldsymbol{u}$ from full-EnKF (b) Ensemble mean of $\boldsymbol{u}$ from P-NIROM-EnKF

Fig. 5. The ensemble mean of velocity solutions using the (a) full-EnKF and (b) P-NIROM-EnKF at day 20.

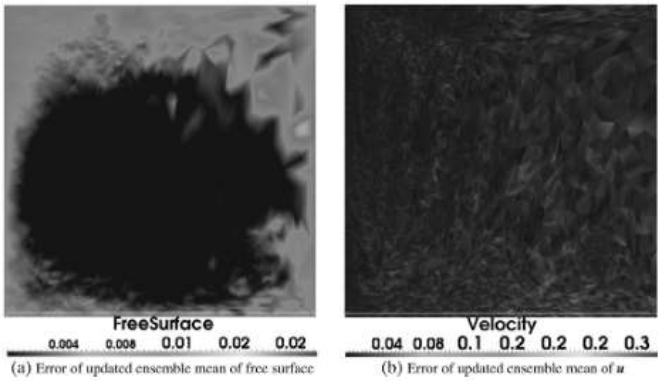
Table 1
CPU cost comparison between full-EnKF and P-NIROM-EnKF required for one time level.

Cases	Model	Assembling and solving	Projection	Interpolation	Total
Ocean gyre	Full model	2.90120	0	0	2.90120
	P-NIROM	0	0.0003	0.0001	0.00040



(a) Error of the ensemble mean of free surface solutions (b) Error of the ensemble mean of velocity solutions

Fig. 6. The error of the ensemble mean of the free surface and velocity solutions (at day 20) using P-NIROM-EnKF.



(a) Error of updated ensemble mean of free surface (b) Error of updated ensemble mean of $\boldsymbol{u}$

Fig. 8. The difference between the full EnKF ensemble mean and the P-NIROM ensemble mean of the free surface(a) and velocity(b) solutions respectively after performing data assimilation.

form of,

$$A = \begin{bmatrix} \phi(\|\mu_1 - \mu_1\|) & \phi(\|\mu_1 - \mu_2\|) & \cdots \phi(\|\mu_1 - \mu_P\|) \\ \phi(\|\mu_2 - \mu_1\|) & \phi(\|\mu_2 - \mu_2\|) & \cdots \phi(\|\mu_2 - \mu_P\|) \\ \vdots & \vdots & \vdots \\ \phi(\|\mu_P - \mu_1\|) & \phi(\|\mu_P - \mu_2\|) & \cdots \phi(\|\mu_P - \mu_P\|) \end{bmatrix}. \tag{19}$$

More details can be founded in the work of [40].

A set of POD basis functions $\Phi_\mu = (\Phi_{\mu,1}, \ldots, \Phi_{\mu,m}, \ldots, \Phi_{\mu,M})$ (M is the number of POD basis function to be chosen) can be obtained by SVD . By projecting the high fidelity model (1) onto the reduced space, one can obtain the P-NIROM (10). In general, for any given background error covariance $\mu \in R^P$, the ROM in (10) at one time level $t_{j-1} \to t_j$ can be re-written:

$$\mathbf{u}^r_{\mu,t_j} = f_\mu\left(\mathbf{u}^r_{\mu,t_{j-1}}\right), \quad j \in (1, \ldots, N_t). \tag{20}$$

In the work of [39], a new ROM that uses a set of hyper-surfaces to replace the traditional ROM's formulation is presented: Non-intrusive reduced order model (NIROM) based on RBF. In the new NIROM presented in [39], the reduced dynamic system f_μ in Eq. (20) has been replaced by a new formulation,

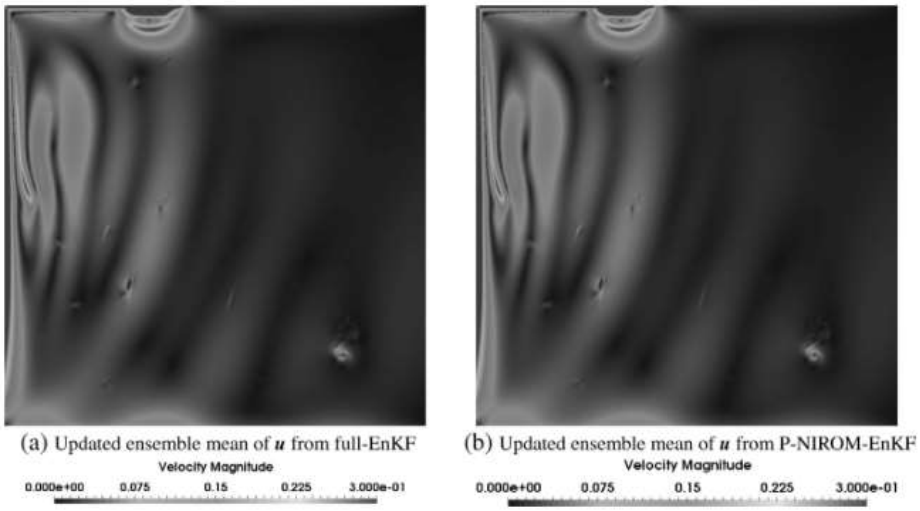
$$f_{\mu,m,t_j}\left(\mathbf{u}^r_\mu\right) = \sum_{s=1}^{S} w_s \phi\left(\left\|\mathbf{u}^r_\mu - \mathbf{u}^{r,s}_{\mu,t_{j-1}}\right\|\right), \quad s \in (1, \ldots, S), \tag{21}$$

where w_s denote the weights, and $\mathbf{u}^r_{\mu,t_j} = \left(\mathbf{u}^r_{\mu,1,t_{j-1}}, \ldots, \mathbf{u}^r_{\mu,M,t_{j-1}}\right)$ are the reduced numerical solution for any given background error covariance $\mu = \sigma^2 \in \mathcal{R}^P$ at time level $t_j - 1$. The RBF ϕ is chosen to be Gaussian RBF, $\phi(r) = e^{-(r/\sigma)^2}$ in this formulation. For more details, see the work of [39].

After obtaining the hyper-surfaces, one can use them to calculate the solutions of the ROM at current time level just by inputting the solutions at previous time level,

$$\mathbf{u}^r_{\mu,m,t_j} = f_{\mu,m,t_j}\left(\mathbf{u}^r_{\mu,1,t_{j-1}}, \ldots, \mathbf{u}^r_{\mu,m,t_{j-1}}, \ldots, \mathbf{u}^r_{\mu,M,t_{j-1}}\right), \quad m \in (1, \ldots, M), \tag{22}$$

where $f_{\mu,m,t_j} \in \mathcal{R}^{M+1}$ is a $M+1$ dimensional surface.



(a) Updated ensemble mean of $\boldsymbol{u}$ from full-EnKF (b) Updated ensemble mean of $\boldsymbol{u}$ from P-NIROM-EnKF

Fig. 7. The updated velocity solutions at day 20 after performing data assimilation using the (a) full-EnKF and (b) P-NIROM-EnKF.

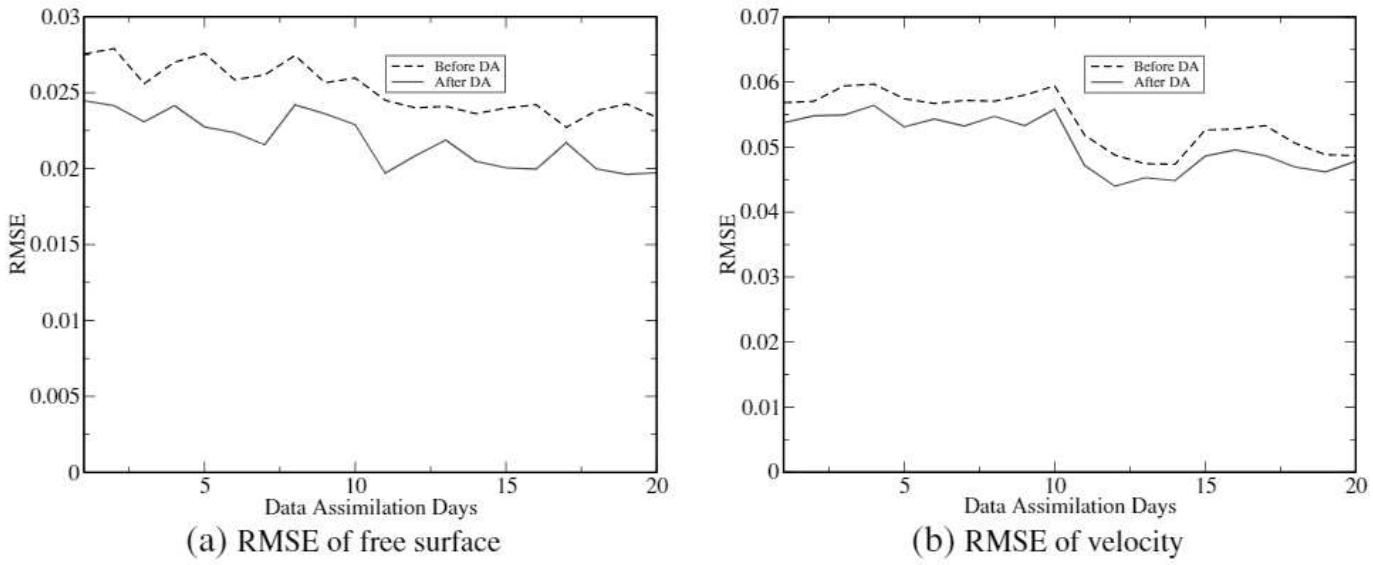


Fig. 9. The RMSE of free surface and velocity solutions before and after performing P-NIROM-EnKF.

3.2. Summary of using P-NIROM for calculating the ensemble covariance matrix

The procedure of constructing the P-NIROM and using it for obtaining the ensemble covariance matrix can be summarised in Algorithm (2). In Algorithm (2), it includes two procedures: online and offline procedures. The offline involves generating a number of different simulations with different inputs $\mathbf{u}_{t_0} + \mu_s$ and constructing the P-NIROM, see steps 1–3 in Algorithm (2). The online procedure involves using the P-NIROM to generate ensembles $\mathbf{U}_{t_j}$, see step 4. When using the P-NIROM, one only need to assign reduced solutions at previous time level into the hyper-surface.

4. EnKF framework based on P-NIROM

The general framework of P-NIROM-EnKF is described in Fig. 1. In this figure, the procedures in blue represent the same procedure with the traditional EnKF method. The procedures in green indicate the calculation of ensembles using the P-NIROM.

(1) Given a set of background error covariances R_{e,t_0}, generate the initial ensemble matrix $\mathbb{U}_{t_0}$;
(2) Using the P-NIROM, propagate each initial ensemble member from time level t_0 to t_j over the reduced space, one obtains:

$$\mathbb{U}^r_{\mu_i,t_j} = \left(\mathbf{u}^r_{\mu_1,t_j}, \ldots, \mathbf{u}^r_{\mu_s,t_j}, \ldots, \mathbf{u}^r_{\mu_S,t_j}\right). \tag{23}$$

(3) Calculate the anomaly ensemble matrix over the reduced space $\mathbf{U}'^r_{t_j} = \frac{1}{\sqrt{(N_e-1)}}\left(\mathbf{U}^r_{t_j} - \overline{\mathbf{U}}^r_{t_j}\right)$;
(4) Calculate the ensemble error covariance matrix:

$$\mathbf{R}_{e,t_j} = \Phi \mathbf{U}'^r_{t_j} \mathbf{U}'^{rT}_{t_j} \Phi^T, \tag{24}$$

(5) Update the ensemble matrix and its error covariance matrix at time level t_j using (6) and (7) respectively:

5. Numerical examples

The non-intrusive reduced order ensemble Kalman filter method is tested by comparing it with the high fidelity full EnKF using a three dimensional wind driven circulation gyre case. The computational domain of the gyre case is 1500 km × 1500 km in the vertical direction. The wind forcing boundary condition is set to be the ocean surface. The beta-plane approximation considers the Coriolis terms. The wind stress uses the cosine function in the horizontal x direction. The velocity and ocean surface height solutions are used to construct the ensemble matrix. The geometry of the gyre case is a 2D unstructured triangle elements with an extrusion in the vertical direction. The resolution of the model has a horizontal length of five kilo meters and biased wind stress. There are 41 observation locations designed as the intersections of several tracks over the whole domain. The twin experiment scheme is employed in the data assimilation framework. Pseudo-observational data (surface height) is obtained from the high fidelity simulation. The EnKF data assimilation occurs every 10 days and the simulation period is 800 days. The mesh in the simulation has 187,884 nodes. The varying input parameter is the background error covariance $\sigma \in [0, 1]$. Given the training background state $(\mathbf{u}_{t_0} + \mathcal{G}(0, \mu_s))$, the snapshots and corresponding POD basis functions are generated at the training stage. The P-NIROM is then constructed and used to calculate the ensemble covariance matrix for any given $\sigma \in [0, 1]$. For details of P-NIROM-EnKF, see Fig. 1.

Fig. 2 shows the singular values and logarithmic scale of singular values resulting from the SVD of the solution snapshots of free surface and velocity. The singular values decrease drastically by nearly three orders of magnitude within the first 3 POD basis functions. In this work, 12 POD basis functions are chosen for the free surface height and all the components of velocity (u, v, w), which can capture above 99.5% of the dynamical energy of solution snapshots, thus ensuring the accuracy of predicted solutions using the P-NIROM.

A comparison of solutions between the full-EnKF and P-NIROM-EnKF has been carried out. Figs. 3 and 5 illustrate the ensemble mean of velocity solutions using the full-EnKF and P-NIROM-EnKF at days 10 and 20 respectively. As shown in the figure, the velocity solutions with the P-NIROM-EnKF are visually close to those of the full-EnKF. Furthermore, the error of the ensemble mean w.r.t. the observations from the P-NIROM-EnKF is given in Figs. 4 and 6. As we can see, the errors of the ensemble mean of solutions in P-NIROM-EnKF are small, which suggests that the accuracy of ensembles obtained from the P-NIROM is maintained whilst the CPU cost is reduced by several orders of magnitude in comparison to the high fidelity model.

Fig. 7 shows the updated velocity solutions after performing data assimilation with the full-EnKF and P-NIROM-EnKF. The up-

dated velocity solutions using the P-NIROM-EnKF exhibit in agreement with those using the full-EnKF. The errors of updated free surface and velocity solutions between the full-EnKF and P-NIROM-EnKF are given in Fig. 8. The figure again shows that the solutions using the P-NIROM-EnKF have roughly the same accuracy of those using the full-EnKF. Fig. 9 shows the root-mean-square error (RMSE) of free surface and velocity before and after ENKF method respectively. It is evident that using P-NIROM-EnKF, the results of free surface and velocity solutions have been improved.

Table 1 shows a comparison of the CPU time required for running the high fidelity full model and P-NIROM. The simulations were performed on 12 cores machine of an Intel®Xeon®X5680 processor with 3.3GHz and 48GB RAM. The test cases were run in serial, therefore, only one core was used when running the test case. It can be seen that the CPU time required for the P-NIROM is considerably less than that for the high fidelity full model and is reduced by a factor of 7000. It is worth noting that the P-NIROM improves the computational efficiency drastically when the number of ensembles is large. In addition, the computational cost of the high fidelity full model is dependent on the number of nodes in the mesh, which means the computation time increases when finer mesh is used.

6. Conclusion

In this paper, a new P-NIROM-EnKF data assimilation method is presented for efficient calculation of ensembles. The derivation of P-NIROM-EnKF is given in details. In the P-NIROM-EnKF, the Smolyak sparse grid method is used to select the distributions of the varying parameters (here, the background error covariance). In addition, the RBF interpolation method to construct a set of response functions (hyper-surfaces or surfaces) to represent the parameter space and reduced fluid dynamics space. Compared to existing EnKF approaches based on ROM, this P-NIROM-EnKF presented here does not need to modify the source code. In addition, it is easy to extend into more complicated applications.

The P-NIROM-EnKF has been implemented under the framework of an unstructured mesh finite element advanced ocean model (Fluidity). The performance of the P-NIROM-EnKF has been evaluated using a 3D wind driven circulation gyre case and compared with full-EnKF. An accuracy assessment is carried out by RMSE. It is demonstrated that accuracy of ensembles from the P-NIROM-EnKF is maintained whilst the size of the ensemble error covariance matrix is significantly decreased, thus reducing the CPU cost by several orders of magnitude. Future work will apply this method to more complicated cases such as city's air pollution or urban flows realistic cases.

Acknowledgments

Dr. Du acknowledges National Key R&D Program of China (Grant No. 2016YFC1401704), National Natural Science Foundation of China (Grant No. 41776041) and the program Grant No. 315030401. The authors are grateful for the support of the EPSRC grant: Managing Air for Green Inner Cities (MAGIC)(EP/N010221/1), NSFC grant 11502241 and the Inovate UK Smart-GeoWells consortium (EP/R005761/1).

References

[1] Altaf M, El Gharamti M, Heemink AW, Hoteit I. A reduced adjoint approach to variational data assimilation. Comput Methods Appl Mech Eng 2013;254:1–13.
[2] Ballard SP, Li Z, Simonin D, Caron J-F. Performance of 4d-var nwp-based nowcasting of precipitation at the met office for summer 2012. Q J R Meteorolog Soc 2016;142(694):472–87.
[3] Barrault M, Maday Y, Nguyen N, Patera A. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. C R Acad Sci Paris, Ser 2004;339:667–72.
[4] Buhmann MD. Radial basis functions: theory and implementations, 12. Cambridge university press; 2003.
[5] Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. J Comput Phys 2013;242:623–47.
[6] Chaturantabut S, Sorensen D. Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 2010;32:2737–64.
[7] Chen Y, Oliver DS. Cross-covariances and localization for enkf in multiphase flow data assimilation. Comput Geosci 2010;14(4):579–601.
[8] Dimitriu G, Apreutesei N. Comparative study with data assimilation experiments using proper orthogonal decomposition method. In: International Conference on Large-Scale Scientific Computing. Springer; 2007. p. 393–400.
[9] Dimitriu G, Apreutesei NC, Stefanescu R. Numerical Simulations with Data Assimilation Using an Adaptive POD Procedure. In: LSSC. Springer; 2009. p. 165–72.
[10] Du J, Fang F, Pain CC, Navon I, Zhu J, Ham DA. POD Reduced-order unstructured mesh modeling applied to 2d and 3d fluid flow. Computers and Mathematics with Applications 2013;65(3):362–79.
[11] Evensen G. Data assimilation: the ensemble kalman filter. Springer Science & Business Media; 2009.
[12] Evensen G, Hove J, Meisingset H, Reiso E, Seim KS, Espelid Ø, et al. Using the enkf for assisted history matching of a north sea reservoir model SPE reservoir simulation symposium. Society of Petroleum Engineers; 2007.
[13] Fang F, Pain C, Navon I, Piggott M, Gorman G, Allison P, et al. Reduced-order modelling of an adaptive mesh ocean model. Int J Numer Methods Fluids 2009;59(8):827–51.
[14] Farrell BF, Ioannou PJ. State estimation using a reduced-order kalman filter. J Atmos Sci 2001;58(23):3666–80.
[15] Feriedoun Sabetghadam AJ. α Regularization of the POD-galerkin dynamical systems of the kuramoto-sivashinsky equation. Appl Math Comput 2012;218:6012–26.
[16] Fang F, Pain C, Navon I, Elsheikh A, Du J, Xiao D. Non-linear petrov-galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods. J Comput Phys 2013;234:540–59.
[17] Gunzburger M, Jiang N, Schneier M. An ensemble-proper orthogonal decomposition method for the nonstationary navier–stokes equations. SIAM J Numer Anal 2017;55(1):286–304.
[18] Han C. Blackbox stencil interpolation method for model reduction. Massachusetts Institute of Technology; 2012.
[19] He J, Sarma P, Durlofsky LJ, et al. Use of reduced-order models for improved data assimilation within an enkf context SPE Reservoir Simulation Symposium. Society of Petroleum Engineers; 2011.
[20] Hoteit I, Pham D-T, Blum J, et al. A simplified reduced order kalman filtering and application to altimetric data assimilation in tropical pacific. J Mar Syst 2002;36(1–2).
[21] Houtekamer PL, Mitchell HL, Pellerin G, Buehner M, Charron M, Spacek L, et al. Atmospheric data assimilation with an ensemble kalman filter: results with real observations. Mon Weather Rev 2005;133(3):604–20.
[22] K Carlberg CB-M, Farhat C. Efficient non-linear model reduction via a least--squares petrov-galerkin projection and compressive tensor approximations. Intern J Numer Methods Eng 2011;86:155–81.
[23] Langodan S, Viswanadhapalli Y, Hoteit I. The impact of atmospheric data assimilation on wave simulations in the red sea. Ocean Eng 2016;116:200–15.
[24] Lin B, McLaughlin D. Efficient characterization of uncertain model parameters with a reduced-order ensemble kalman filter. SIAM J Sci Comput 2014;36(2):B198–224.
[25] Liu B, Gharamti M, Hoteit I. Assessing clustering strategies for gaussian mixture filtering a subsurface contaminant model. J Hydrol (Amst) 2016;535:1–21.
[26] Osth J, Noack BR, Krajnovi S, Barros D, Bore J. 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 2014;747:518–44. doi:10.1017/jfm.2014.168.
[27] Rosenthal WS, Venkataramani S, Mariano AJ, Restrepo JM. Displacement data assimilation. J Comput Phys 2017;330:594–614.
[28] Schlegel M, Noack BR. On long-term boundedness of galerkin models. J Fluid Mech 2015;765:325–52. doi:10.1017/jfm.2014.736.
[29] Ştefănescu R, Sandu A, Navon IM. POD/DEIM Reduced-order strategies for efficient four dimensional variational data assimilation. J Comput Phys 2015;295:569–95.
[30] Stefano P, Andrea M, Alfio Q. A reduced basis ensemble kalman filter for state/parameter identification in large-scale nonlinear dynamical systems; 2016.
[31] Tian X, Xie Z, Sun Q. A pod-based ensemble four-dimensional variational assimilation method. Tellus A 2011;63(4):805–16.
[32] Tsiaras KP, Hoteit I, Kalaroni S, Petihakis G, Triantafyllou G. A hybrid ensemble-oi kalman filter for efficient data assimilation into a 3-d biogeochemical model of the mediterranean. Ocean Dyn 2017:1–18.
[33] Walton S, Hassan O, Morgan K. Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. Appl Math Model 2013;37(20):8930–45.
[34] Willcox K, Megretski A. Model reduction for large-scale linear applications. In: Proc. of 13th IFAC Symposium on System Identification, Rotterdam, Netherlands; 2003. p. 1431–6.
[35] Xiao D. Non-intrusive reduced order models and their applications. Imperial College London; 2016.

[36] Xiao D, Fang F, Buchan A, Pain C, Navon I, Muggeridge A. Non-intrusive reduced order modelling of the navier–stokes equations. Comput Methods Appl Mech Eng 2015;293. 552–541.
[37] Xiao D, Fang F, Buchan AG, Pain C, Navon* I, Du J, et al. Non-linear model reduction for the navier-stokes equations using residual DEIM method. J Comput Phys 2014;263:1–18. doi:10.1016/j.jcp.2014.01.011.
[38] Xiao D, Fang F, Du J, Pain C, Navon I, Buchan AG, et al. Non-linear petrov-galerkin methods for reduced order modelling of the navier-stokes equations using a mixed finite element pair. Comput Methods Appl Mech Eng 2013;255:147–57. doi:10.1016/j.bbr.2011.03.031.
[39] Xiao D, Fang F, Pain C, Hu G. Non-intrusive reduced order modelling of the navier-stokes equations based on RBF interpolation. Int J Numer Methods Fluids 2015;79(11):580–95. doi:10.1002/fld.4066.
[40] Xiao D, Fang F, Pain C, Navon I. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. Comput Methods Appl Mech Eng 2017;317:868–89. doi:10.1016/j.cma.2016.12.033.
[41] Chu MSY, Hahn J. State-preserving nonlinear model reduction procedure. Chemical Engineering Science 2011;66:3907–13.