INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **59**:827–851
Published online 9 July 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1841

Reduced-order modelling of an adaptive mesh ocean model

F. Fang[1,*,†], C. C. Pain[1], I. M. Navon[2], M. D. Piggott[1], G. J. Gorman[1], P. A. Allison[1] and A. J. H. Goddard[1]

[1]*Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College, Prince Consort Road, London SW7 2BP, U.K.*
[2]*School of Computational Science and Department of Mathematics, Florida State University, Tallahassee, FL 32306-4120, U.S.A.*

SUMMARY

A novel proper orthogonal decomposition (POD) model has been developed for use with an advanced unstructured mesh finite-element ocean model, the Imperial College Ocean Model (ICOM, described in detail below), which includes many recent developments in ocean modelling and numerical analysis. The advantages of the POD model developed here over existing POD approaches are the ability:

1. To increase accuracy when representing geostrophic balance (the balance between the Coriolis terms and the pressure gradient). This is achieved through the use of two sets of geostrophic basis functions where each one is calculated by basis functions for velocities u and v.
2. To speed up the POD simulation. To achieve this a new numerical technique is introduced, whereby a time-dependent matrix in the discretized equation is rapidly constructed from a series of time-independent matrices. This development imparts considerable efficiency gains over the often-used alternative of calculating each finite element over the computational domain at each time level.
3. To use dynamically adaptive meshes in the above POD model.

Received 8 June 2007; Revised 8 April 2008; Accepted 10 April 2008

KEY WORDS: POD; reduced-order modelling; ocean model; finite element; unstructured adaptive mesh

*Correspondence to: F. Fang, Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College, Prince Consort Road, London SW7 2BP, U.K.
†E-mail: f.fang@imperial.ac.uk, http://amcg.ese.imperial.ac.uk, http://people.scs.fsu.edu/~navon/index.html

Contract/grant sponsor: National Environment Research Council; contract/grant number: NE/C52101X/1
Contract/grant sponsor: NSF; contract/grant numbers: ATM-0201808, CCF-0635162, ID 0635162

1. INTRODUCTION

Proper orthogonal decomposition (POD) is a numerical procedure that can be used to extract a basis for a modal decomposition from an ensemble of signals. The technique was originally proposed independently by Kosambi [1], Loève [2] and Karhunen [3] and is alternatively known as the Karhunen–Loève decomposition (KLD) method. Related methodologies have, however, been developed in a variety of disparate disciplines [4]. The procedure is also known as principal components analysis (PCA) [5] in statistics, and empirical orthogonal functions (EOF) in oceanography [6, 7] and meteorology [8].

POD has been widely and successfully applied to numerous fields, including signal analysis and pattern recognition [5], fluid dynamics and coherent structures [9–12] and image reconstruction [13].

An important innovation in the use of POD for large problems in fluid dynamics involves using a series of snapshots, which consist of a set of state solutions evaluated at different time instants and determined from the evolution in time of the full model [14–17]. The snapshots are used to compute the POD basis vectors to yield an optimal representation of the data so that for any given basis vector size, the two-norm of the error between the original and reconstructed snapshots is minimized.

POD methodologies, in combination with the Galerkin projection procedure, have also been shown to provide an efficient means of generating reduced-order models [11, 18, 19]. This technique essentially identifies the most energetic modes in a time-dependent system, thus providing a means of obtaining a low-dimensional description of the system's dynamics. To improve the accuracy of reduced models, the goal-oriented approach has been used to optimize the POD bases [20, 21]. The dual-weighted POD approach seeks to provide an 'enriched' set of basis functions combining information from both model dynamics and the data assimilation system. The practical utility of the POD approach is extended to large-scale complex flow dynamics such as atmospheric and ocean modelling and the four-dimensional variational (4D-Var) data assimilation [18, 22–24], where it is often hard to solve their discrete models due to their large dimension (typically in the range of 10^6–10^8). The POD has been successfully applied for a broad range of fluid dynamic applications; however, the POD approach (1) is sensitive to details of snapshots used and the choice of inner products, (2) depends on how well the data ensemble captures the relevant system behaviour. Therefore, the POD method cannot always guarantee accuracy or stability of the reduced model [25]. The loss of accuracy in the POD-reduced model is compensated by the gain on computational efficiency.

The motivation of the present study is to develop a POD-based reduced model for an unstructured ocean model, i.e. the Imperial College Ocean Model (ICOM), that can simultaneously resolve both small- and large-scale ocean flows while smoothly varying resolution and conforming to complex coastlines and bathymetry [26, 27]. What distinguishes the reduced model developed here from other existing reduced models is the inclusion of an adaptive mesh capability. This represents the main challenge in the implementation of the POD approach. When adaptive meshes are employed, the mesh resolution requirements may be spatially and temporally different, as the meshes are adapted according to the flow features. This unavoidably introduces difficulties in the implementation of a POD-based reduced model for an adaptive model. One of these challenges is that snapshots can be of different lengths at different time levels. To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced models. The solutions from the full

DOI: 10.1002/fld

models are then interpolated from their own mesh onto the same reference fixed mesh at each time level. This allows the same number of base modes at each time level.

One of the important issues in reducing ocean models is to accurately represent the geostrophic balance. In this study, the pressure variable is divided into two parts: non-geostrophic and geostrophic pressures. The basis function for the geostrophic pressure is constructed by two sets, which should satisfy the geostrophic balance and be calculated by the basis functions for the velocity components u and v, respectively.

In order to construct an efficient POD model, a new numerical technique is introduced. A general discretized reduced model at the time level n can be expressed as $A^n\alpha^n = s^n$, where α^n is a vector of the coefficients associated with the basis functions for the variables in the full model. Calculating matrix A^n at each time level is compute-intensive. To accelerate the POD simulation, matrix A^n is constructed by a set of time-independent matrices, which are obtained prior to running the reduced model.

Error estimation is a critical issue in reduced-order modelling. The references related to error estimation can be found in [28–33] to cite but a few. Early work on error in model reduction has been done by Utku *et al.* [28], where the first-order error estimation of the model reduction for nonlinear systems is given at a small number of time steps. More recently, the dual-weighted-residual (DWR) method, which makes use of the solution of an adjoint system, has been incorporated into the error estimation of reduced models [21, 30, 32, 34]. Using this method, it is possible to obtain '*a priori*' error estimate for a certain cost functional of solution. This error estimate can be used to satisfy a given error tolerance. It can also be used to form a very efficient low-dimensional basis especially tailored to the cost functional of interest. For example, Homescu *et al*. [30, 35] employed the DWR method to determine the regions of validity of the reduced models, that is, ranges of perturbations in the original system over which the reduced model is still appropriate. Furthermore, Hinze and Volkwein [29] incorporated both the time derivatives and adjoint information into snapshots in the error estimation for the partial differential equation-constrained optimization and POD inverse model. In this study, the error estimation is carried out using a simple approach described in [18], where, a spectral norm $\|A\|_2$ is defined to estimate the spatial error between the full and reduced models. An error bound is given by $\sqrt{\lambda_{(M+1)}}$ if M POD bases are chosen, where $\lambda_{(M+1)}$ is the $(M+1)$th eigenvalue of AA^{T}.

The remainder of this paper is structured as follows: In Section 2, the ICOM is briefly described. In Section 3, the reduced forward model is then derived, while the geostrophic pressure, mesh adaptivity and acceleration of the POD simulation are discussed in detail in Section 4. In Section 5 the above reduced model is applied to and illustrated by some relevant cases. Summary and conclusions are drawn in the final section.

2. DESCRIPTION OF ICOM

In this study, a POD-based reduced model is developed for ICOM that can simultaneously resolve both small- and large-scale ocean flows while smoothly varying resolution and conforming to complex coastlines and bathymetry [26, 27]. With more appropriate focused numerical resolution (e.g. adaptive and anisotropic resolution of fronts and boundary layers, and optimal representation of vertical structures in the ocean) ocean dynamics may be accurately predicted during future climatic change. To accurately represent local flow around steep topography, the hydrostatic

DOI: 10.1002/fld

assumption is not made in this study. Here, the pressure variable is split into the non-geostrophic and geostrophic parts, which are solved separately. This allows the accurate representation of hydrostatic/geostrophic balance [26]. In principle, coupling of the momentum and continuity equations results in an extremely large system of equations to solve, for which an efficient solution strategy is difficult to devise. Therefore, a technique (e.g. a projection method) is used in which the pressure and velocity variables are solved for independently, thus reducing the total dimension of the systems that must be solved for (for details see [27]).

The underlying model equations consist of the three-dimensional (3D) non-hydrostatic Boussinesq equations,

$$\nabla\cdot\mathbf{u}=0 \tag{1}$$

$$\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}+f\mathbf{k}\times\mathbf{u}=-\nabla p-\rho g\mathbf{k}+\nabla\cdot\boldsymbol{\tau} \tag{2}$$

where $\mathbf{u}\equiv(u,v,w)^{\mathrm{T}}\equiv(u_1,u_2,u_3)^{\mathrm{T}}$ is the velocity vector, $\mathbf{x}\equiv(x,y,z)^{\mathrm{T}}\equiv(x_1,x_2,x_3)^{\mathrm{T}}$ are the orthogonal Cartesian coordinates, p is the perturbation pressure ($p:=p/\rho_0$, ρ_0 is the constant reference density), f represents the Coriolis inertial force, g represents the acceleration due to gravity, ρ is the perturbation density ($\rho:=\rho/\rho_0$) and $\mathbf{k}=(0,0,1)^{\mathrm{T}}$. The stress tensor $\boldsymbol{\tau}$ is used to represent viscous terms and is defined in terms of the deformation rate tensor $\mathbf{S}$ as

$$\tau_{ij}=2\mu_{ij}S_{ij},\quad S_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{1}{3}\sum_{k=1}^{3}\frac{\partial u_k}{\partial x_k},\quad 1\leqslant i,j\leqslant 3$$

with no summation over repeated indices. In this study the horizontal kinematic viscosities (μ_{11},μ_{22}) and vertical kinematic viscosity (μ_{33}) take constant values with the off-diagonal components of $\boldsymbol{\tau}$ defined by $\mu_{ij}=(\mu_{ii}\mu_{jj})^{1/2}$, see also [27, 36]. For barotropic flow (baroclinic flow is incorporated in Section 4), the pressure p consists of hydrostatic $p_{\mathrm{h}}(z)$ and non-hydrostatic $p_{\mathrm{nh}}(x,y,z,t)$ components. The hydrostatic component of pressure balances exactly the constant buoyancy force and both terms are therefore dropped at this stage.

ICOM utilizes dynamic adaptation of a fully unstructured tetrahedral mesh in 3D, as presented in Pain *et al.* [37]. This technique uses a form of h-refinement (or mesh optimization) to adapt the mesh, changing the size, shape and location of tetrahedral elements to optimize the mesh according to specific criteria, as defined by an error measure. The algorithm is based on a series of mesh connectivity and node position searches, defining the mesh quality. A Riemannian metric tensor reflecting the error measure is used to calculate the desired element size and, importantly, its shape. A functional is used to gauge the mesh quality; this functional embodies both element size and shape with respect to the metric tensor. A local-based search strategy is adopted to carry out the adaptation operations (including face-to-edge and edge-to-face swapping; edge-to-edge swapping; and local node movement or mesh smoothing) to minimize the functional. The algorithm is robust, produces high-quality anisotropic meshes and has a time complexity, which varies linearly with the number of elements, see Pain *et al.* [37]. An alternate approach of defining an error measure to guide an adaptive meshing algorithm for unstructured tetrahedral finite elements is to utilize an adjoint or goal-based method. This method is based on a functional, encompassing important features of the flow structure. The sensitivity of this functional, with respect to the solution variables, is used as the basis from which an error measure is derived. This error measure acts to predict those areas of the domain where resolution should be changed.

DOI: 10.1002/fld

3. REDUCED-ORDER OCEAN MODEL

A derivation of the 3D reduced forward equations is described in this section. The POD reduction is the most efficient choice among linear decompositions in the sense that it can capture the greatest possible kinetic energy.

3.1. Discretized ocean model

To construct the discretized ocean model, the linear basis function N_i is chosen for the velocity components and non-geostrophic pressure, while quadratic basis function M_i is used for the geostrophic pressure (Figure 1). The variables to be solved can be expressed in the finite-element form:

$$\begin{aligned} u &= \sum_{i=1}^{\mathcal{N}} u_i N_i, \quad v = \sum_{i=1}^{\mathcal{N}} v_i N_i, \quad w = \sum_{i=1}^{\mathcal{N}} w_i N_i \\ p_{\mathrm{ng}} &= \sum_{i=1}^{\mathcal{N}} p_{\mathrm{ng},i} N_i, \quad p_g = \sum_{i=1}^{\mathcal{N}} p_{g,i} M_i \end{aligned} \tag{3}$$

where $\mathcal{N}$ is the number of nodes; u, v and w are the velocity components in x, y and z directions, respectively, and p_{ng} and p_g are the non-geostrophic and geostrophic pressures, respectively.

3.2. Proper orthogonal decomposition

The model variables (e.g. u, v, w, p) are sampled at defined checkpoints during the simulation period $[t_1, \ldots, t_n, \ldots, t_K]$, also referred to as snapshots (K being the number of snapshots). The snapshots can be obtained either from a mathematical (numerical) model of the phenomenon or from experiments/observations. The sampled values of variables at the snapshot i are stored in a vector U_i with $\mathcal{N}$ entries ($\mathcal{N}$ being the number of nodes); here, U can represent one of the variables u, v, w, p. The average of the ensemble of snapshots is defined as

$$\bar{U}_i = \frac{1}{K} \sum_{k=1}^{K} U_{k,i}, \quad 1 \leqslant i \leqslant \mathcal{N} \tag{4}$$

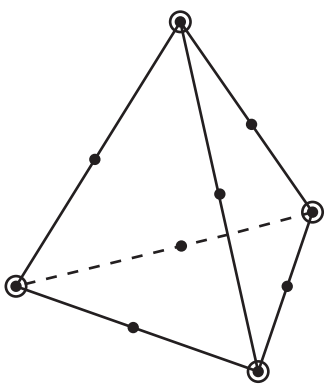


Figure 1. Linear and quadratic mixed finite element. ○: u, v, w, p_{ng}; ●: p_g.

DOI: 10.1002/fld

Taking the deviation from the mean of variables forms

$$V_{k,i} = U_{k,i} - \bar{U}_i, \quad 1 \leqslant i \leqslant \mathcal{N} \tag{5}$$

A collection of all $V_{k,i}$ constructs a rectangular $\mathcal{N} \times K$ matrix A. The goal of the POD is to find a set of orthogonal basis functions $\Phi = \Phi_1, \Phi_2, \ldots, \Phi_K$ such that it maximizes

$$\frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{\mathcal{N}} (V_{k,i} \Phi_k) \tag{6}$$

subject to

$$\sum_{k=1}^{K} \Phi_k^2 = 1 \tag{7}$$

The Singular Value Decomposition (SVD) is used to find the optimal base Φ of the optimization problem (6). From SVD, the matrix $A \in R^{\mathcal{N} \times K}$ can be expressed as

$$A = X \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} Y^{\mathrm{T}} \tag{8}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_d) \in R^{d \times d}$, $X = (\Phi_1, \Phi_2, \ldots, \Phi_{\mathcal{N}}) \in R^{\mathcal{N} \times \mathcal{N}}$ and $Y = (y_1, y_2, \ldots, y_K) \in R^{K \times K}$ are matrices that consist of the orthogonal vectors for AA^{T} and $A^{\mathrm{T}}A$, respectively. The order $\mathcal{N}$ for matrix AA^{T} is far larger than the order K for matrix $A^{\mathrm{T}}A$. Therefore, a $K \times K$ eigenvalue problem is solved

$$A^{\mathrm{T}} A y_k = \lambda_k y_k, \quad 1 \leqslant k \leqslant K \tag{9}$$

This procedure is equivalent to an SVD. The eigenvalues $\lambda_k = \sigma_k^2$ are real and positive and should be sorted in an descending order. The POD basis vectors Φ_k associated with the eigenvalues λ_k are orthogonal and expressed as follows:

$$\Phi_k = A y_k / \sqrt{\lambda_k} \tag{10}$$

The kth eigenvalue is a measure of the kinetic energy transferred within the kth basis mode. If the POD spectrum (energy) decays fast enough, practically all the support of the invariant measure is contained in a compact set. Roughly speaking, all the likely realizations in the ensemble can be found in a relatively small set of bounded extent. By neglecting modes corresponding to the small eigenvalues, the following formula is therefore defined to choose a low-dimensional basis of size M $(M \ll K)$:

$$I(M) = \frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{K} \lambda_i} \tag{11}$$

subject to

$$M = \arg\min\{I(M) : I(M) \geqslant \gamma\} \tag{12}$$

where $0 \leqslant \gamma \leqslant 1$ is the percentage of energy, which is captured by the POD basis $\Phi_1, \ldots, \Phi_m, \ldots, \Phi_M$.

DOI: 10.1002/fld

3.3. POD reduced model for ICOM

The variables in (1) and (2) can be expressed as an expansion of the POD basis functions for u, v, w, p, that is,

$$\begin{aligned} u(t,x,y,z) &= \bar{u} + \sum_{m=1}^{M_u} \alpha_{u,m}(t)\Phi_{u,m}(x,y,z) \\ v(t,x,y,z) &= \bar{v} + \sum_{m=1}^{M_v} \alpha_{v,m}(t)\Phi_{v,m}(x,y,z) \\ w(t,x,y,z) &= \bar{w} + \sum_{m=1}^{M_w} \alpha_{w,m}(t)\Phi_{w,m}(x,y,z) \\ p(t,x,y,z) &= \bar{p} + \sum_{m=1}^{M_p} \alpha_{p,m}(t)\Phi_{p,m}(x,y,z) \end{aligned} \tag{13}$$

where $\bar{u}$, $\bar{v}$, $\bar{w}$ and $\bar{p}$ are the means of the ensemble of snapshots for the velocity components u, v, w and pressure p, respectively; $\Phi_{u,m}$, $\Phi_{v,m}$, $\Phi_{w,m}$ and $\Phi_{p,m}$ are the POD bases for the velocity components u, v, w and pressure p, respectively; M_u, M_v, M_w and M_p are the numbers of POD bases, which are chosen for the velocity components u, v, w and pressure p, respectively; and $\alpha_{u,m}$, $\alpha_{v,m}$, $\alpha_{w,m}$ and $\alpha_{p,m}$ are the coefficients to be determined.

Substituting (13) into (1) and (2) and taking the POD basis function as the test function, then integrating over the computational domain, the POD reduced model is obtained as

$$\begin{aligned} \frac{\partial \alpha_{u,m_u}}{\partial t} &= F_u(t, \alpha_{u,1}, \ldots, \alpha_{u,M_u}, \alpha_{v,1}, \ldots, \alpha_{v,M_v}, \alpha_{w,1}, \ldots, \alpha_{w,M_w}, \alpha_{p,1}, \ldots, \alpha_{p,M_p}) \\ \frac{\partial \alpha_{v,m_v}}{\partial t} &= F_v(t, \alpha_{u,1}, \ldots, \alpha_{u,M_u}, \alpha_{v,1}, \ldots, \alpha_{v,M_v}, \alpha_{w,1}, \ldots, \alpha_{w,M_w}, \alpha_{p,1}, \ldots, \alpha_{p,M_p}) \\ \frac{\partial \alpha_{w,m_w}}{\partial t} &= F_w(t, \alpha_{u,1}, \ldots, \alpha_{u,M_u}, \alpha_{v,1}, \ldots, \alpha_{v,M_v}, \alpha_{w,1}, \ldots, \alpha_{w,M_w}, \alpha_{p,1}, \ldots, \alpha_{p,M_p}) \\ \frac{\partial \alpha_{p,m_p}}{\partial t} &= F_p(t, \alpha_{u,1}, \ldots, \alpha_{u,M_u}, \alpha_{v,1}, \ldots, \alpha_{v,M_v}, \alpha_{w,1}, \ldots, \alpha_{w,M_w}, \alpha_{p,1}, \ldots, \alpha_{p,M_p}) \end{aligned} \tag{14}$$

where $1 \leqslant m_u \leqslant M_u, 1 \leqslant m_v \leqslant M_v, 1 \leqslant m_w \leqslant M_w, 1 \leqslant m_p \leqslant M_p$ and M_u, M_v, M_w, M_p are the numbers of the basis functions for u, v, w, p, respectively. The initial conditions for solving (14) are

$$\begin{aligned} \alpha_{u,m_u}(0) &= ((u(0,x,y,z) - \bar{u}(x,y,z)), \Phi_{u,m_u}) \\ \alpha_{v,m_v}(0) &= ((v(0,x,y,z) - \bar{v}(x,y,z)), \Phi_{v,m_v}) \\ \alpha_{w,m_w}(0) &= ((w(0,x,y,z) - \bar{w}(x,y,z)), \Phi_{w,m_w}) \\ \alpha_{p,m_p}(0) &= ((p(0,x,y,z) - \bar{p}(x,y,z)), \Phi_{p,m_p}) \end{aligned} \tag{15}$$

DOI: 10.1002/fld

The errors for the above POD model are bounded by the following expression (details of the derivation can be found in Appendix B and Luo *et al.* [18]):

$$\begin{aligned}
\|u_{\text{full}}-u\|_2 &\leqslant \sqrt{\lambda_{u(M_u+1)}}\\
\|v_{\text{full}}-v\|_2 &\leqslant \sqrt{\lambda_{v(M_v+1)}}\\
\|w_{\text{full}}-w\|_2 &\leqslant \sqrt{\lambda_{w(M_w+1)}}\\
\|p_{\text{full}}-p\|_2 &\leqslant \sqrt{\lambda_{p(M_p+1)}}
\end{aligned} \tag{16}$$

where $\lambda_{u(M_u+1)}$ is the (M_u+1)th eigenvalue for $A_u A_u^{\mathrm{T}}$, $\lambda_{v(M_v+1)}$ is the (M_v+1)th eigenvalue for $A_v A_v^{\mathrm{T}}$, $\lambda_{w(M_w+1)}$ is the (M_w+1)th eigenvalue for $A_w A_w^{\mathrm{T}}$ and $\lambda_{p(M_p+1)}$ is the (M_p+1)th eigenvalue for $A_p A_p^{\mathrm{T}}$; the matrices for each variable A_u, A_v, A_w and A_p can be calculated using Equation (5).

4. GEOSTROPHIC PRESSURE, ADAPTIVE MESHES AND EFFICIENCY IN POD SIMULATION

4.1. Geostrophic pressure

An important issue in ocean modelling is the treatment of the Coriolis term in the momentum equation. To allow the accurate representation of the geostrophic balance (17) or (18) (i.e. to allow for the use of equal-order interpolants for the representation of velocity $f\mathbf{k}\times\mathbf{u}$ and ∇p_g), quadratic pressure and linear velocity are needed [38]. However, the LBB (Ladyzhenskaya–Babuska–Brezzi) stability condition requires precisely the opposite: higher-order polynomials for u, v than p. To solve this contradiction, the pressure in (2) is divided into two parts: $p = p_{\text{ng}} + p_g$. The geostrophic pressure has to satisfy the geostrophic balance:

$$-\nabla p_g = f\mathbf{k}\times\mathbf{u} \tag{17}$$

Taking the divergence of Equation (17), an elliptic equation for geostrophic pressure is obtained as

$$-\nabla^2 p_g = \frac{\partial(-fv)}{\partial x} + \frac{\partial(fu)}{\partial y} \tag{18}$$

To accurately represent geostrophic pressure, its basis functions are split into two sets: Φ_{pgu} and Φ_{pgv}, which are associated with the u- and v-velocity components. The geostrophic pressure that satisfies Equation (18) can be obtained from a quadratic finite-element representation (Figure 1) while linear finite-element representations are used for the velocity components. Furthermore, the geostrophic pressure can be represented by a summation of the two sets of geostrophic basis functions, which are calculated by solving the following elliptic equations using a conjugate

DOI: 10.1002/fld

gradient iterative method:

$$-\nabla^2 \Phi_{pgu,m} = \frac{\partial(f\Phi_{u,m})}{\partial y}$$
$$-\nabla^2 \Phi_{pgv,m} = \frac{\partial(-f\Phi_{v,m})}{\partial x} \tag{19}$$

where $\Phi_{pgu,m}$ and $\Phi_{pgv,m}$ are the basis functions, respectively, for velocity components u and v. $m=(1,\ldots,M)$ indicates a set of basis functions, where M is the total number of basis functions. The geostrophic pressure can therefore be expressed as

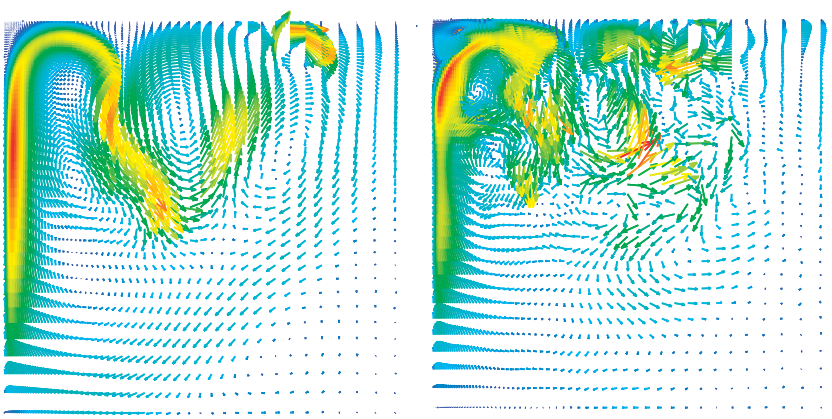
$$p_g = \bar{p}_g + \sum_{m=1}^{M} \alpha_{u,m}\Phi_{u,m} + \sum_{m=1}^{M} \alpha_{v,m}\Phi_{v,m} \tag{20}$$

In addition, the average geostrophic pressure is calculated from

$$-\nabla^2 \bar{p}_g = \frac{\partial(-f\bar{v})}{\partial x} + \frac{\partial(f\bar{u})}{\partial y} \tag{21}$$

where $\bar{u}=(1/K)\sum_{k=1}^{K} u_k$ and $\bar{v}=(1/K)\sum_{k=1}^{K} v_k$. Neumann boundary conditions are employed for $\Phi_{pgu,m}$, $\Phi_{pgv,m}$ and $\bar{p}_g$ in Equations (19) and (21). This only serves to define $\Phi_{pgu,m}$, $\Phi_{pgv,m}$ and $\bar{p}_g$ up to an arbitrary additive constant, $\Phi_{pgu,m}$, $\Phi_{pgv,m}$ and $\bar{p}_g$ are thus fixed by defining them to be zero at an arbitrary node.

It can be seen that the numerical instability in the velocity field (Figure 2(b)) is removed by using the new numerical method described above (Figure 2(a)). This method can also be easily extended to represent buoyancy with temperature and salinity dependence by introducing more basis functions with which to balance these buoyancy terms.



(a) (b)

Figure 2. Effect of geostrophic balance on results from POD simulation: (a) with and (b) without the new geostrophic pressure method.

DOI: 10.1002/fld

4.2. Adaptive meshes in POD

When adaptive meshes are employed in ocean models, the mesh resolution requirements vary spatially and temporally, as the meshes are adapted according to the flow features through the whole simulation. The dimensional size of the variable vectors is different at each time level since the number of nodes varies during the simulation. Snapshots can therefore be of different lengths at different time levels. This unavoidably brings difficulties in the implementation of a POD-based reduced model for an adaptive model. To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced model. The solutions from the original full model are interpolated from their own mesh onto the same reference fixed mesh at each time level and then stored in the snapshots. The information at the snapshots is used to find the optimal POD basis. This allows the same length of base modes to be obtained at each time level. The resolution of the reference mesh and the interpolation errors between the two meshes (the adaptive mesh and the fixed reference mesh) may affect the accuracy of the POD simulation. This will be explained and discussed in detail through the applications presented below. To reduce the interpolation error, a high-order interpolation approach can be adopted.

4.3. Acceleration of the POD simulation

For simplicity, suppose the discrete forward model to be solved at the time level n assumes the form

$$\mathbf{A}^n \alpha^n = s^n \tag{22}$$

where

$$s^n = \mathbf{B}^n + f \tag{23}$$

where $\mathbf{A}^n$ and $\mathbf{B}^n$ are the matrices at the time level n, $\alpha^n = (\alpha_1^n, \ldots, \alpha_m^n, \ldots, \alpha_M^n)$ is the vector variables to be solved at the time level n, here including the coefficients related to the basis functions for the velocity components and the pressure, i.e. $\alpha_m^n = (\alpha_{u,m}^n, \alpha_{v,m}^n, \alpha_{w,m}^n, \alpha_{p,m}^n)$, s^n is a discretized source term at the time level n. Note that it is time-consuming to calculate the time-dependent matrix $\mathbf{A}^n$ at each finite element and node over the computational domain at each time level. To speed up the POD simulation, a new numerical technique is introduced, that is, the time-dependent matrix $\mathbf{A}^n$ is constructed by a set of sub-matrices independent of time. For a nonlinear simulation, matrices $\mathbf{A}^n$ and $\mathbf{B}^n$ can be expressed as

$$\mathbf{A}^n = \hat{\mathbf{A}}_0 + \sum_{m=1}^{M} \alpha_m^{n-1} \hat{\mathbf{A}}_m \tag{24}$$

$$\mathbf{B}^n = \hat{\mathbf{B}}_0 + \sum_{m=1}^{M} \alpha_m^{n-1} \hat{\mathbf{B}}_m \tag{25}$$

where matrices $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$, $\hat{\mathbf{A}}_m$ and $\hat{\mathbf{B}}_m$ are time independent and are derived in Appendix A. Therefore, instead of calculating the time-dependent matrix $\mathbf{A}^n$ at each time level, one needs to calculate those sub-matrices $\hat{\mathbf{A}}^n$ and $\hat{\mathbf{B}}^n$ once prior to the POD simulation. This significantly speeds up the POD simulation.

DOI: 10.1002/fld

5. APPLICATION CASES AND NUMERICAL RESULTS

The utility of the new POD reduced-order model is herein assessed and validated in three two-dimensional (2D), time-dependent test cases: flow past a cylinder, flow past a cylinder on a β plane (whereby the Coriolis parameter is set to vary linearly in space, $f=\beta y$) and a gyre (defined as a spiral system of movement, here, a wind-driven barotropic circulation).

5.1. Case 1: flow past a cylinder

The non-dimensional 2D case is composed of a cylinder with a radius of 1 in the computational domain (1 element deep, 29 long and 10 wide). An inlet boundary with a velocity of 1 (non-dimensional) flows parallel to the domain length towards the right of the domain. The centre of the cylinder is placed 5 from the inlet boundary. The Reynolds number (Re) is 100. The boundary conditions applied to the cylinder and both lateral sides are no normal flow and zero shear (slip) boundary conditions with a spin-up period of 8 and a time step of 0.02 (non-dimensional).

The initial conditions are defined by running the full model from the 'static' state during the spin-up (the process of a model adjusting to its forcing) period $[0, 8]$ (non-dimensional). In this case, an adaptive mesh is adopted in the full model. To guarantee the same length of POD bases at the snapshots, a reference fixed mesh is chosen for the POD simulation (right panel in Figure 3). The information from the full model is interpolated from the adaptive mesh (right panel in Figure 3) onto the reference mesh (right panel). The mesh for the full model adapts every 20 time steps with maximum and minimum mesh sizes of 0.1 and 0.04, respectively. The maximum mesh aspect ratio is 1000.

In this case, 31 snapshots and 20 basis functions are chosen for u, v, w and p, for which 98% of energy is captured. Figure 3 shows the velocity field (vector) obtained from the full (left panel) and POD (right panel) models. It is apparent that the results (especially the details of eddies near the cylinder) from both models are in good agreement. For details, Figure 4 shows the blow-up of the velocity field around the cylinder at the time level $t=14$. It is observed that the results obtained by running the reduced model and the full model provide almost identical details of local flow. By carrying out a comparison between the results from the full and reduced models, the overall error of numerical POD solutions is less than $\max\{\sqrt{\lambda_{u21}}, \sqrt{\lambda_{v21}}\} \leqslant 0.3$ (here, λ_{u21} and λ_{v21} are the 21th eigenvalues, respectively, for the velocity components u and v). The reduced model simulation takes 9 min, while it takes 40 min to run the full forward model.

When adaptive meshes are adopted in the original model, an interpolation error is introduced to snapshots and the POD reduced model. Evaluation of accuracy of the POD model is here carried out through comparison of POD solutions with and without the mesh adaptivity scheme in the POD basis. Figure 5 shows the velocity contour at time levels $t=9.6, 14, 16, 20$, where the heavy-weight, medium-weight and light-weight lines represent the numerical results from the full model, and reduced model with and without the use of adaptive meshes, respectively. The POD results are in good agreement with those from the full model. To further quantify the quality of the POD results with the use of adaptive meshes, the root mean-squared error (RMSE) and correlation coefficient of results between the POD model and the full model are plotted in Figure 6. It is seen that the RMSE of the POD results is less than 0.15 while the correlation is within 97%.

5.2. Case 2: flow past a cylinder on a β plane ($\beta=7.5$)

The POD model developed here is further applied to a flow past a cylinder on a β plane. The schematic of the model domain and the boundary conditions are the same as those in the previous

DOI: 10.1002/fld

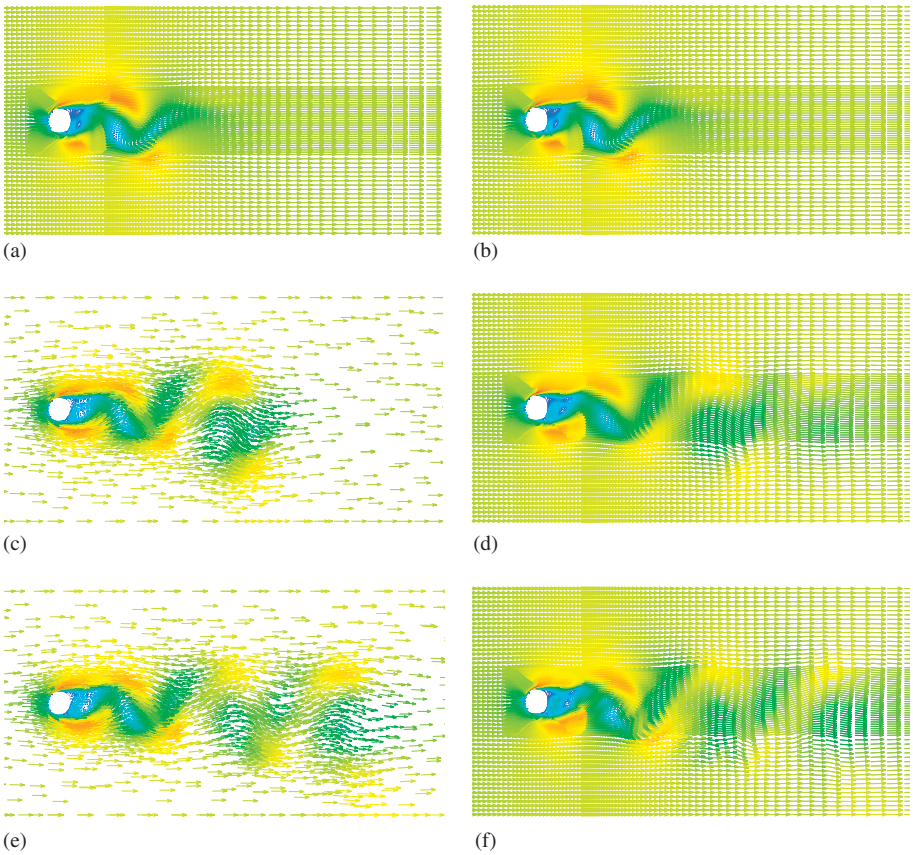


Figure 3. Case 1: comparison of the velocity field between the full and reduced models ($Re=100$) (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t=8$; middle panel: at the time level $t=14$; bottom panel: at the time level $t=20$).

case. The β effect on the flow is considered in this case. The Coriolis parameter is given by $f=\beta y$, here $\beta=7.5$. The novel approach described in Section 4 is used to calculate the geostrophic pressure. The Reynolds number is $Re=200$. The spin-up period is 0.4. The simulation period is 1.6 and the time step is chosen to be 0.002.

Figure 7 shows the velocity field (vector) obtained from the full (left panel) and POD (right panel) models. It can be seen (at time level $t=1.6$) from both the full and reduced simulations that two meandering jets form downstream of the cylinder, with a small separated region in between, which is caused by breaking the Rossby waves [39]. A total of 30 snapshots and 15 basis functions, here, are chosen for u, v, w and p, for which 99.99% of energy is captured. The maximum error of numerical POD solutions is less than $\max\{\sqrt{\lambda_{u16}}, \sqrt{\lambda_{v16}}\} \leqslant 10^{-2}$ (here, λ_{u16} and λ_{v16} are the 16th eigenvalues, respectively, for the velocity components u and v). It takes only 3 h to complete the simulation by running the reduced model, which is considerably less than the CPU time (38–39 h) required to run the full forward model. The computational cost is thus significantly reduced by 92% (i.e. by a factor of 13).

DOI: 10.1002/fld

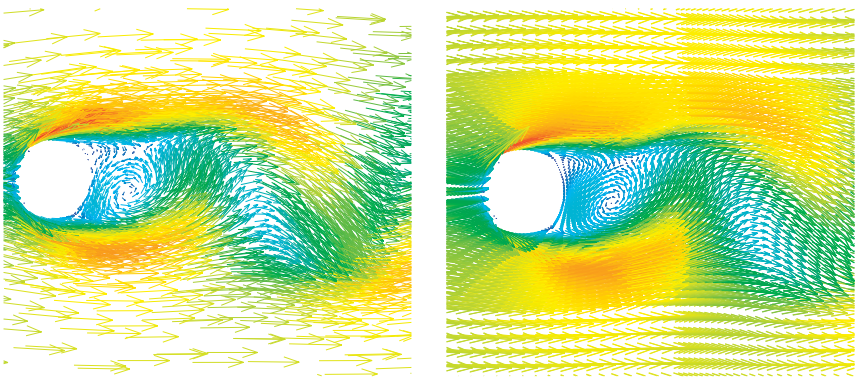


Figure 4. Case 1: blowup of the velocity field in Figure 3 around the cylinder at the time level $t=14$: (a) the full model and (b) the reduced order model.

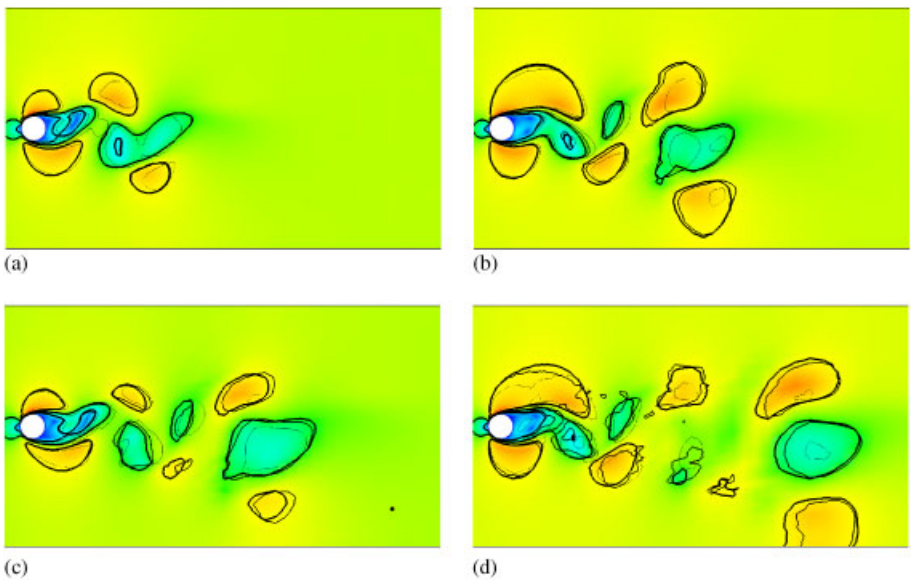


Figure 5. Contour of velocity at time levels: (a) $t=9.6$; (b) $t=14$; (c) $t=16$; (d) $t=20$ (the heavy-weight lines: the full model; the medium-weight lines: the POD model without adaptive meshes; the light-weight lines: the POD model with adaptive meshes).

5.3. Case 3: gyre

To simulate a wind-driven barotropic circulation, the computational domain is taken to be a square box of 1000 km with a depth of 500 m. The wind forcing on the free surface is given as

$$\tau_y = \tau_0 \cos(\pi y/L), \quad \tau_x = 0.0 \tag{26}$$

DOI: 10.1002/fld

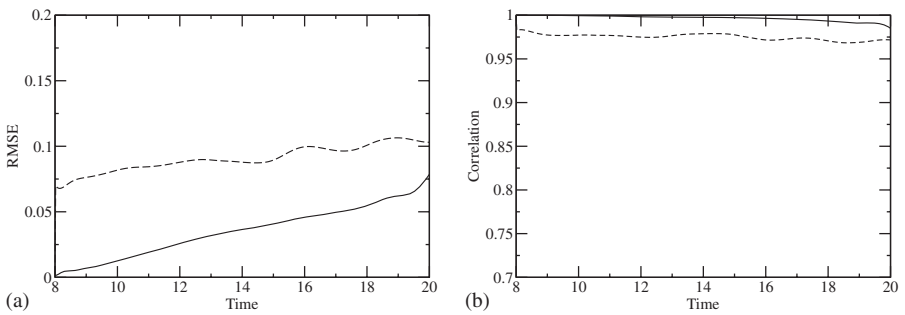


Figure 6. RMSE and correlation of velocity results between the POD model and the full model (a) RMS and (b) correlation (the solid line: without the use of adaptive mesh technique; the dashed line: with the use of adaptive mesh technique).

where τ_x and τ_y are the wind stresses on the free surface along the x and y directions, respectively, and $L=1000\,\text{km}$. A maximum zonal wind stress of $\tau_0=0.1\,\text{N}\,\text{m}^{-1}$ is applied in the latitude (y) direction. The Reynolds number $Re=250$, $\beta=1.8\times10^{-11}$ and the reference density $\rho_0=1000\,\text{kg}\,\text{m}^{-1}$.

The problem is non-dimensionalized, so that the domain is a box of horizontal dimension 1, and a depth of 0.0005 with one element in the vertical. Incorporating the β-plane approximately gives a non-dimensional rotation vector of $\Omega=(0, 257.143, 0)$ and non-dimensional wind stress of $\tau_0=163.2653$. The time step is 3.78×10^{-4}, equivalent to 3 h. No-slip boundary conditions are applied to the lateral boundaries. The spin-up period is 0.3024 (100 days). The simulation period is [0.3024, 0.6048] (i.e. [100, 200] days).

As discussed above, the POD simulation could end up having results far from the true values (Figure 2) if the geographic pressure is represented improperly. In this study, to accurately represent the geostrophic balance, the geostrophic pressure is calculated using the novel approach described in Section 4. A comparison of the velocity field between the full and POD models is provided in Figure 8. The velocity fields at the different time levels exhibit an overall good agreement with those from the full model.

To judge the quality of the POD model developed here, an error estimate is provided. The percentage of energy represented by the POD bases is listed in Table I. About 99% of the energy can be captured when 60 POD bases are chosen with 81 snapshots, and 97% of the energy is captured when 30 POD bases and 41 snapshots are used. About 91% of energy is captured if half the number of leading POD bases is chosen. In general, the more the POD bases and snapshots are chosen, the better the energy is represented. A list of CPU times required for running the reduced model is provided in Table II. It usually takes less than 3 h to complete the simulation by running the reduced model. For all the settings, compared with the CPU time (30 h) required for running the full model, the CPU time for running the reduced model decreases by a factor of 10 while about 98.5% of the energy is captured. It is noted that 99% of CPU time required for the reduced model is used to set up the POD bases and calculate the time-independent sub-matrices (details in Section 4.3) for preparation of running the simulation. The actual CPU time required for running the reduced model during the simulation period is less than 1 min.

The error of numerical results obtained by a different number of POD bases can be calculated as in Equation (16). Figure 9 displays the eigenvalues and the error associated with a corresponding

DOI: 10.1002/fld

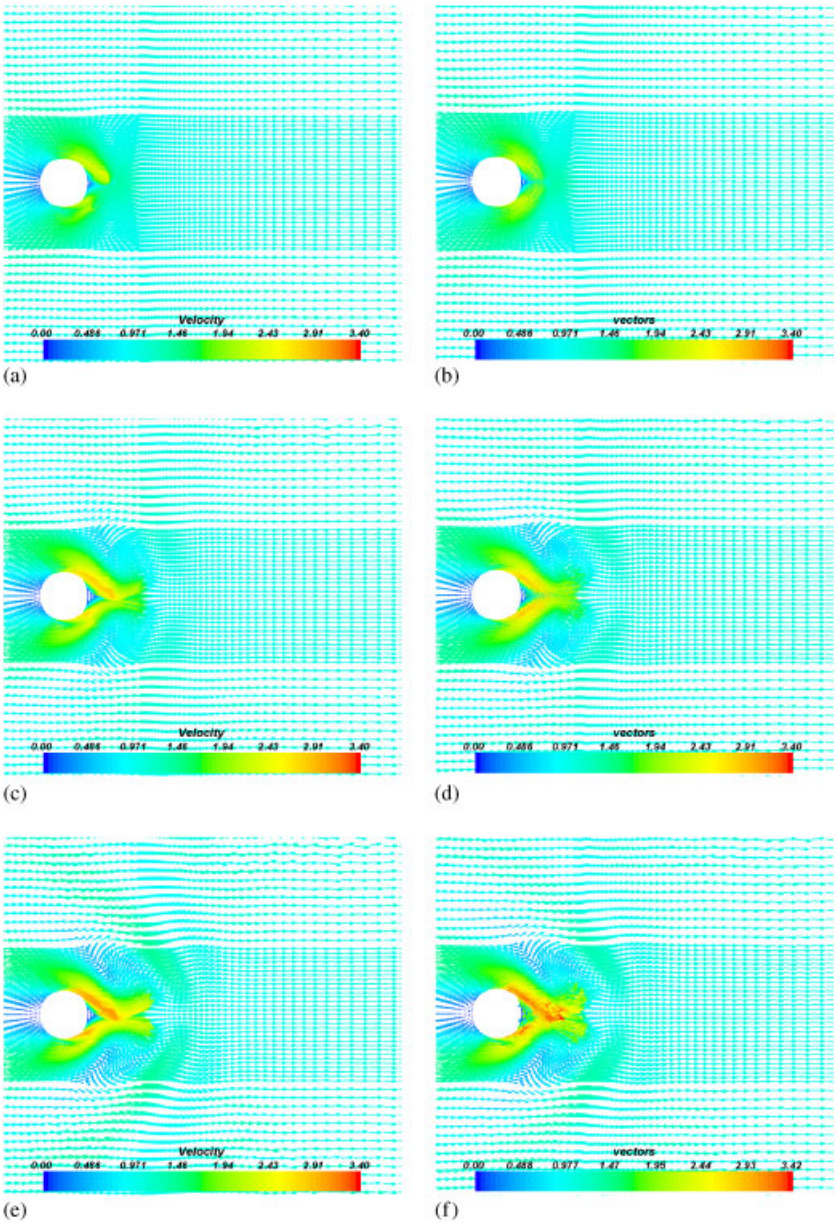


Figure 7. Case 2—cylinder on a β plane ($\beta = 7.5$, $Re = 200$): comparison of the velocity field between the full and reduced models (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t = 0.4$; middle panel: at the time level $t = 1.0$; bottom panel: at the time level $t = 1.6$).

DOI: 10.1002/fld

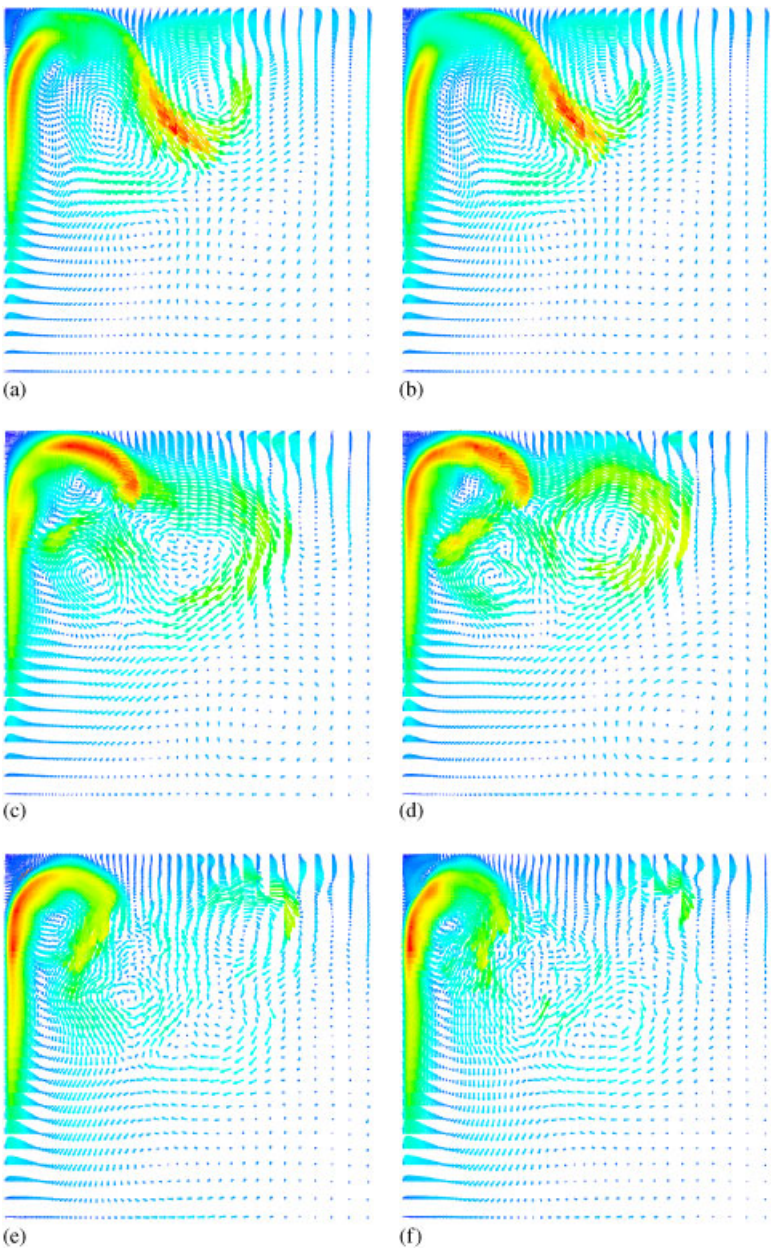


Figure 8. Comparison of the velocity field between full model (left column) and reduced model (right column) in test case 3 for $Re=250$. Models are virtually indistinguishable. After POD basis and time-independent component of the reduced model has been generated from the forward model, reduced model took approximately 1 min of the CPU time. In contrast, the full forward model took 30 h to run: (a) and (b) $t=100$ days; (c) and (d) $t=150$ days; and (e) and (f) $t=200$ days.

DOI: 10.1002/fld

Table I. Energy percentage captured by the POD bases for velocity components, u, v and pressure p.

Energy (%) captured (41 snapshots)	Energy (%) captured (81 snapshots)
77.373 (for u, 10 bases)	88.614 (for u, 20 bases)
76.003 (for v, 10 bases)	89.723 (for v, 20 bases)
81.103 (for p, 10 bases)	92.880 (for p, 20 bases)
91.448 (for u, 20 bases)	97.025 (for u, 40 bases)
91.693 (for v, 20 bases)	97.738 (for v, 40 bases)
94.343 (for p, 20 bases)	98.614 (for p, 40 bases)
97.386 (for u, 30 bases)	99.458 (for u, 60 bases)
97.624 (for v, 30 bases)	99.600 (for v, 60 bases)
98.584 (for p, 30 bases)	99.766 (for p, 60 bases)

Table II. A list of CPU times required for running the reduced model and the reduced percent of CPU compared with that (30 h) required for running the full model.

CPU time (h) (41 snapshots)	CPU time (h) (81 snapshots)
0.77 (reduced by 97% of the CPU time compared with the full model while 10 bases are chosen)	1.4 (reduced by 95% of the CPU time compared with the full model while 20 bases are chosen)
1.30 (reduced by 95% of the CPU time compared with the full model while 20 bases are chosen)	2.47 (reduced by 92% of the CPU time compared with the full model while 40 bases are chosen)
2.00 (reduced by 93% of the CPU time compared with the full model while 30 bases are chosen)	11.0 (reduced by 63% of the CPU time compared with the full model while 60 bases are chosen)

Note that the actual CPU time required for running the reduced model during the simulation period is less than 1 min after the POD bases and the time-independent sub-matrices (Section 4.3) are calculated.

number of POD bases for the velocity components u and v. It is noted that the first 25% of the leading POD bases have a significant impact on POD results. The error of the POD results decreases by 70–80% of its original values while about 76% of the energy can be captured if the first 25% of the leading POD bases are chosen.

The RMSE and correlation coefficient of results between the full and POD models at different time levels are provided in Figures 10 and 11. It can be seen that as the simulated time accrues, the POD error increases, while the correlation decreases. During the first half of the simulation period, the POD results are in general satisfactory since the correlations are mostly larger than 0.8 and the RMSE is less than $2.5\,\mathrm{m\,s^{-1}}$. It is also shown that an increase in the number of snapshots and POD bases leads to an improvement in the correlation coefficient and reduction in the RMSE (Figure 12).

6. SUMMARY AND CONCLUSIONS

A POD reduced model has been developed for a finite-element adaptive mesh refinement ocean model (ICOM). To the best of our knowledge, this is the first attempt to apply the POD approach

DOI: 10.1002/fld

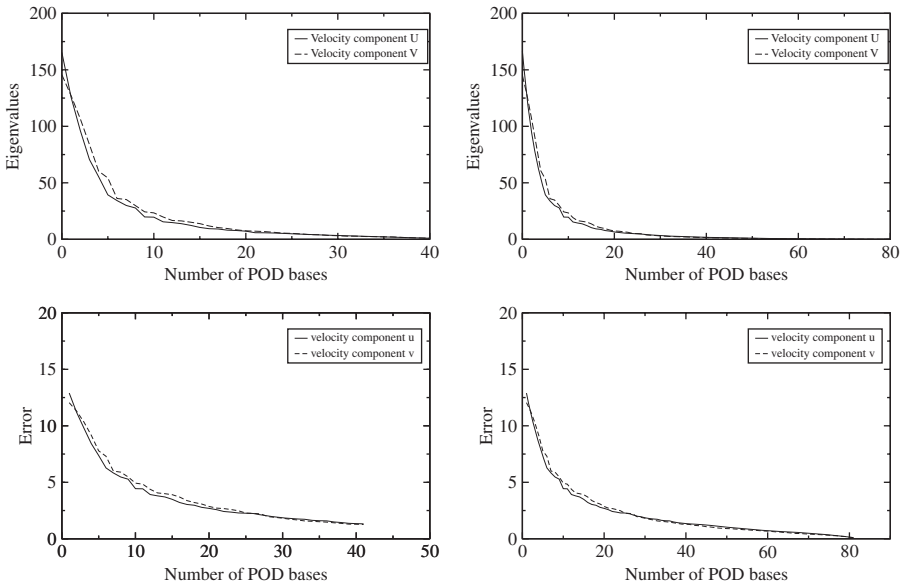


Figure 9. Case 3: eigenvalues and errors for velocity components u and v (top panel: eigenvalues; bottom panel: error left panel: 41 snapshots; right panel: 81 snapshots).

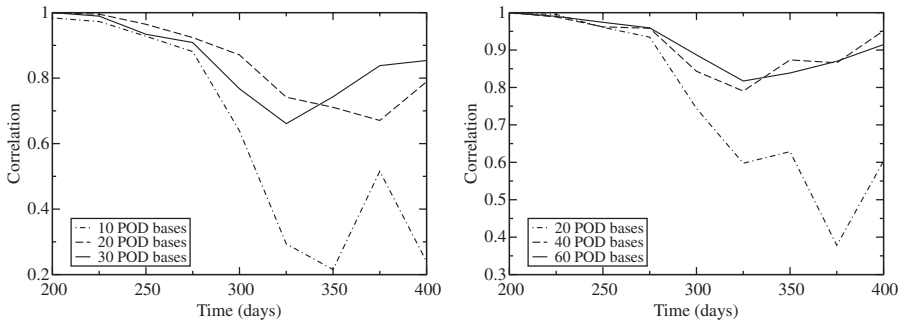


Figure 10. Case 3: correlation of velocity results between the full and POD models *versus* time (left panel: 41 snapshots; right panel: 81 snapshots).

to an adaptive finite-element ocean model. To be able to obtain the same length of POD snapshots at each time level, a reference fixed mesh is chosen for the POD reduced model. The results from the full model are interpolated from the adaptive mesh onto the reference mesh for each of the snapshots and stored to find the optimal POD bases. The interpolation errors between the two meshes (the adaptive mesh and the fixed reference mesh) may affect the accuracy of the POD simulation. Evaluation of accuracy of the POD model is carried out through comparison of POD solutions with and without the mesh adaptivity scheme in the POD basis. It is noted in our case that the results from the POD model become oscillatory and unstable as the Reynolds number increases beyond 1000. To improve the stability of the POD model, an H^1 norm is suggested to

DOI: 10.1002/fld

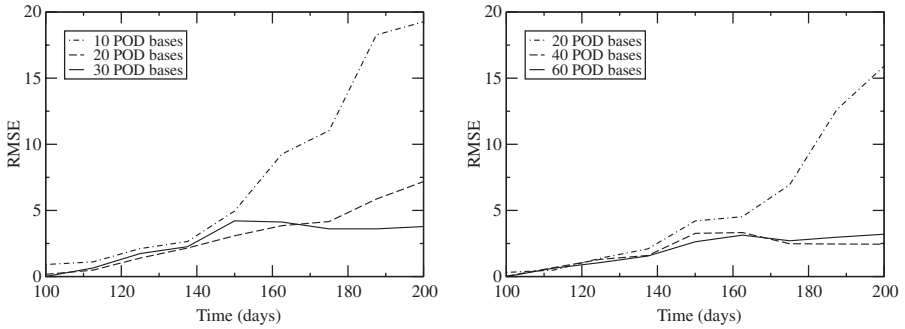


Figure 11. Case 3: RMS at time levels (left panel: 41 snapshots; right panel: 81 snapshots).

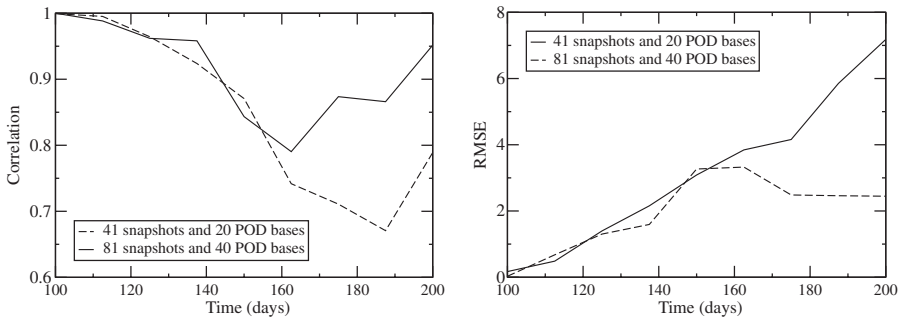


Figure 12. Case 3: comparison of correlation and RMS of velocity results with different snapshot numbers (left panel: correlation of velocity results between the full and POD models; right panel: RMS).

redefine the norms involved in the POD definition so that derivatives of the snapshots as well as those of the basis functions are included in the POD average [40].

The POD reduced model is applied to 2D time-dependent ocean cases. The Coriolis effect is taken into account in the POD model. An accurate representation of the geostrophic balance can be achieved by two sets of basis functions for the geostrophic pressure, which are obtained from the basis functions for the velocity components u and v, respectively.

An error analysis has also been carried out for the validation and accuracy of the adaptive POD model. It is shown that the results from the reduced model coincide with those from the full model. The correlation of results between the reduced and full models can achieve 80–99%, while the RMSE results are less than $2.5\,\mathrm{m\,s^{-1}}$ and 97% of the energy can be captured if a suitable number of POD bases are chosen (say, if half of the leading POD bases are chosen with 81 snapshots). The error for leading POD results decreases by 70–80% of its original values if only the first 25% of leading POD bases are chosen. An increase in the number of snapshots and POD bases leads to an improvement in the correlation coefficient and reduction in the RMSE.

To increase the efficiency of the POD simulation, a new numerical technique is introduced, that is, one can generate *a priori* time-independent decomposition of the matrix in the discretized POD equations prior to running the reduced model. It is proved that the computer cost required for running the reduced model can decrease by a factor of 30.

DOI: 10.1002/fld

Further research will address the following issues of a goal-oriented POD to optimize the weights of POD bases, Gappy POD, which allows the consideration of incomplete data sets, and adjoint POD for data assimilation with adaptivity both in mesh as well as in the updating of the reduced-order model controllers.

APPENDIX A

The second-order Crank–Nicolson time stepping algorithm is used here. Taking into account Equation (13), i.e.

$$\begin{aligned} u(t,x,y,z) &= \bar{u} + \sum_{m=1}^{M_u} \alpha_{u,m}(t)\Phi_{u,m}(t,x,y,z) \\ v(t,x,y,z) &= \bar{v} + \sum_{m=1}^{M_v} \alpha_{v,m}(t)\Phi_{v,m}(t,x,y,z) \\ w(t,x,y,z) &= \bar{w} + \sum_{m=1}^{M_w} \alpha_{w,m}(t)\Phi_{w,m}(t,x,y,z) \\ p(t,x,y,z) &= \bar{p} + \sum_{m=1}^{M_p} \alpha_{p,m}(t)\Phi_{p,m}(t,x,y,z) \end{aligned} \tag{A1}$$

where $\bar{u}$, $\bar{v}$, $\bar{w}$ and $\bar{p}$ are the means of the ensemble of snapshots for the velocity components u, v, w and pressure p, respectively; $\Phi_{u,m}$, $\Phi_{v,m}$, $\Phi_{w,m}$ and $\Phi_{p,m}$ are the POD bases for the velocity components u, v, w and pressure p, respectively; M_u, M_v, M_w and M_p are the number of POD bases that are chosen for the velocity components u, v, w and pressure p, respectively (here, $M_u = M_v = M_w = M_p$); $\alpha_{u,m}$, $\alpha_{v,m}$, $\alpha_{w,m}$, $\alpha_{p,m}$ are the coefficients to be determined. The sub-matrices A_0 and B_0 associated with each term in the momentum equations are expressed as follows:

(1) The A_0 matrices in u momentum equation;

$$\begin{aligned} \hat{A}_{0,i,j} &= \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\left[\bar{u}_k\frac{\partial\Phi_{u,j,k}}{\partial x} + \bar{v}_k\frac{\partial\Phi_{u,j,k}}{\partial y} + \bar{w}_k\frac{\partial\Phi_{u,j,k}}{\partial z}\right. \\ &\quad \left. +\mu\left(\frac{\partial^2\Phi_{u,j,k}}{\partial x^2} + \frac{\partial^2\Phi_{u,j,k}}{\partial y^2} + \frac{\partial^2\Phi_{u,j,k}}{\partial z^2}v\right)\right] \\ \hat{A}_{0,i,M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}(-f\Phi_{v,j,k}) \\ \hat{A}_{0,i,2M+j} &= 0 \\ \hat{A}_{0,i,3M+j} &= \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\frac{\partial\Phi_{p,j,k}}{\partial x} \end{aligned} \tag{A2}$$

DOI: 10.1002/fld

(2) The A_0 matrices in v momentum equation;

$$\hat{A}_{0,M+i,j} = \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}(f\Phi_{u,j,k})$$

$$\hat{A}_{0,M+i,M+j} = \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\left[\bar{u}_k \frac{\partial \Phi_{v,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{v,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{v,j,k}}{\partial z} + \mu\left(\frac{\partial^2 \Phi_{v,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{v,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{v,j,k}}{\partial z^2}\mu\right)\right] \tag{A3}$$

$$\hat{A}_{0,M+i,2M+j} = 0$$

$$\hat{A}_{0,M+i,3M+j} = \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k} \frac{\partial \Phi_{p,j,k}}{\partial y}$$

(3) The A_0 matrices in w momentum equation;

$$\hat{A}_{0,2M+i,j} = 0$$

$$\hat{A}_{0,2M+i,M+j} = 0$$

$$\hat{A}_{0,2M+i,2M+j} = \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\left[\bar{u}_k \frac{\partial \Phi_{w,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{w,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{w,j,k}}{\partial z} + \mu\left(\frac{\partial^2 \Phi_{w,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{w,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{w,j,k}}{\partial z^2}\mu\right)\right] \tag{A4}$$

$$\hat{A}_{0,2M+i,3M+j} = \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k} \frac{\partial \Phi_{p,j,k}}{\partial z}$$

(4) The A_0 matrices in continuity equation;

$$\hat{A}_{0,3M+i,j} = \sum_{k=1}^{\mathcal{N}} \Phi_{p,i,k} \frac{\partial \Phi_{u,j,k}}{\partial x}$$

$$\hat{A}_{0,3M+i,M+j} = \sum_{k=1}^{\mathcal{N}} \Phi_{p,i,k} \frac{\partial \Phi_{v,j,k}}{\partial y} \tag{A5}$$

$$\hat{A}_{0,3M+i,2M+j} = \sum_{k=1}^{\mathcal{N}} \Phi_{p,i,k} \frac{\partial \Phi_{w,j,k}}{\partial z}$$

DOI: 10.1002/fld

(5) The B_0 matrices in u momentum equation;

$$\begin{aligned}
\hat{B}_{0,i,j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\Phi_{u,j,k} - \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\left[\bar{u}_k \frac{\partial \Phi_{u,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{u,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{u,j,k}}{\partial z}\right. \\
&\quad \left. + \mu\left(\frac{\partial^2 \Phi_{u,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{u,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{u,j,k}}{\partial z^2}\mu\right) + \frac{\partial \bar{u}_k}{\partial x}\Phi_{u,j,k}\right] \\
\hat{B}_{0,i,M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\left[-f\Phi_{v,j,k} + \frac{\partial \bar{u}_k}{\partial y}\Phi_{v,j,k}\right] \\
\hat{B}_{0,i,2M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\frac{\partial \bar{u}_k}{\partial z}\Phi_{w,j,k} \\
\hat{B}_{0,i,3M+j} &= -\frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\frac{\partial \Phi_{p,j,k}}{\partial x}
\end{aligned} \tag{A6}$$

(6) The B_0 matrices in v momentum equation;

$$\begin{aligned}
\hat{B}_{0,M+i,j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\left[f\Phi_{u,j,k} + \frac{\partial \bar{v}_k}{\partial x}\Phi_{u,j,k}\right] \\
\hat{B}_{0,M+i,M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\Phi_{v,j,k} - \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\left[\bar{u}_k \frac{\partial \Phi_{v,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{v,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{v,j,k}}{\partial z}\right. \\
&\quad \left. + \mu\left(\frac{\partial^2 \Phi_{v,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{v,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{v,j,k}}{\partial z^2}\mu\right) + \frac{\partial \bar{v}_k}{\partial y}\Phi_{v,j,k}\right] \\
\hat{B}_{0,M+i,2M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\frac{\partial \bar{v}_k}{\partial z}\Phi_{w,j,k} \\
\hat{B}_{0,M+i,3M+j} &= -\frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\frac{\partial \Phi_{p,j,k}}{\partial y}
\end{aligned} \tag{A7}$$

(7) The B_0 matrices in w momentum equation;

$$\begin{aligned}
\hat{B}_{0,2M+i,j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\frac{\partial \bar{w}_k}{\partial x}\Phi_{u,j,k} \\
\hat{B}_{0,2M+i,M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\frac{\partial \bar{w}_k}{\partial y}\Phi_{v,j,k} \\
\hat{B}_{0,2M+i,2M+j} &= \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\Phi_{w,j,k} - \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{w,j,k}\left[\bar{u}_k \frac{\partial \Phi_{w,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{w,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{w,j,k}}{\partial z}\right. \\
&\quad \left. + \mu\left(\frac{\partial^2 \Phi_{w,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{w,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{w,j,k}}{\partial z^2}\mu\right) + \frac{\partial \bar{w}_k}{\partial z}\Phi_{w,j,k}\right] \\
\hat{B}_{0,2M+i,3M+j} &= -\frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\frac{\partial \Phi_{p,j,k}}{\partial z}
\end{aligned} \tag{A8}$$

DOI: 10.1002/fld

The sub-matrices A_m and B_m associated with the nonlinear terms in the momentum equations are written as follows:

(8) The A_m matrices;

$$\begin{aligned}
\hat{A}_{m,i,j} &= \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\left[\Phi_{u,m,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,m,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,m,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{A}_{m,M+i,2M+j} &= \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\left[\Phi_{u,m,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,m,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,m,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{A}_{m,2M+i,3M+j} &= \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\left[\Phi_{u,m,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,m,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,m,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right]
\end{aligned} \tag{A9}$$

(9) The B_m matrices;

$$\begin{aligned}
\hat{B}_{m,i,j} &= -\frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{u,i,k}\left[\Phi_{u,m,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,m,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,m,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{B}_{m,M+i,2M+j} &= -\frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{v,i,k}\left[\Phi_{u,m,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,m,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,m,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{B}_{m,2M+i,3M+j} &= -\frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \Phi_{w,i,k}\left[\Phi_{u,m,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,m,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,m,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right]
\end{aligned} \tag{A10}$$

where μ is the kinematic viscosity, M represents the number of basis functions and $\mathcal{N}$ is the number of nodes, $1 \leqslant i, j \leqslant M$.

APPENDIX B

As stated in Section 3.1, from SVD, matrix A_u ($\mathcal{N} \times K$, a collection of snapshots where the sampled values of variables, e.g. the velocity component u, are stored) can be expressed as

$$A_u = X \begin{pmatrix} \Lambda_u & 0 \\ 0 & 0 \end{pmatrix} Y^{\mathrm{T}} \tag{B1}$$

where $\mathbf{\Lambda}_u = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_d) \in R^{d \times d}$, $X = (\Phi_1, \Phi_2, \ldots, \Phi_{\mathcal{N}}) \in R^{\mathcal{N} \times \mathcal{N}}$ and $Y = (y_1, y_2, \ldots, y_K) \in R^{K \times K}$ are matrices that consist of the orthogonal vectors for $A_u A_u^{\mathrm{T}}$ and $A_u^{\mathrm{T}} A_u$, respectively. A_u $K \times K$ eigenvalue problem is therefore solved

$$A_u^{\mathrm{T}} A_u y_k = \lambda_{u,k} y_k, \quad 1 \leqslant k \leqslant K \tag{B2}$$

The eigenvalues $\lambda_{u,k} = \sigma_k^2$ $(1 \leqslant k \leqslant K)$ are real and positive and should be sorted in an descending order. The POD basis vectors Φ_k associated with the eigenvalues $\lambda_{u,k}$ are orthogonal and expressed as follows:

$$\Phi_k = A_u y_k / \sqrt{\lambda_{u,k}} \tag{B3}$$

DOI: 10.1002/fld

Using the relationship properties of spectral radius and $\|\cdot\|_{2,2}$ for matrices, if $M_u \leqslant r = \text{rank}(A_u)$ $(r \leqslant K)$, the following equation is then obtained:

$$\sigma_{M_u+1} = \min_{\text{rank}(B)\leqslant M_u} \|A_u - B\|_{2,2} = \|A_u - A_{M_u}\|_{2,2} \tag{B4}$$

where $A_{M_u} = \sum_{m=1}^{M} \sigma_m \Phi_m y_m^{\mathrm{T}}$. The minimum distance between matrices A_u and B is obtained by using matrix A_{M_u}, i.e. A_{M_u} should be the optimal representation of A_u.

Denote the K column vectors of matrices A_u by u^k and ε^k $(k = 1, 2, \ldots, K)$ by unit column vectors. Using the compatibility of the norm for matrices and vectors yields

$$\|u^k - P_{M_u}(u^k)\|_2 = \|(A_u - A_{M_u})\varepsilon\|_2 \leqslant \|A_u - A_{M_u}\|_{2,2}\|\varepsilon\|_2 = \sqrt{\lambda_{u(M_u+1)}} \tag{B5}$$

where $P_{M_u}(a^l) = \sum_{m=1}^{M_u} (\Phi_m, a^m)\Phi_m$, $(\Phi_m, u^m)\Phi_m$ is the canonical inner product for vectors Φ_m and u^m. Therefore, $P_{M_u}(u^k)$ is the optimal approximation to u^m, whose error is less than $\sqrt{\lambda_{u(M_u+1)}}$.

ACKNOWLEDGEMENTS

This work was carried out under funding from the National Environment Research Council NE/C52101X/1. Prof. I. M. Navon would like to acknowledge the support of NSF grants ATM-0201808 and CCF-0635162 managed by Dr Eun K. Park, and NSF Award ID 0635162. Finally, the authors would like to thank the anonymous reviewer who assisted in substantially improving this paper.

REFERENCES

1. Kosambi DD. Statistics in function space. *Journal of the Indian Mathematical Society* 1943; **7**:76–88.
2. Loève M. Sur les fonctions aleatoires stationaires du second ordre. *Revue Scientifique* 1945; **83**:297–303.
3. Karhunen K. Zur spektraltheorie stochastischer prozesse. *Annales Academiæ Scientiarum Fennicæ* 1946; **34**:1–7.
4. Wu CG, Liang YC, Lin WZ, Lee HP, Lim SP. A note on equivalence of proper orthogonal decomposition methods. *Journal of Sound and Vibration* 2003; **265**:1103–1110.
5. Fukunaga K. *Introduction to Statistical Recognition* (2nd edn). Computer Science and Scientific Computing Series. Academic Press: Boston, MA, 1990.
6. Jolliffe IT. *Principal Component Analysis*. Springer: Berlin, 2002.
7. Crommelin DT, Majda AJ. Strategies for model reduction: comparing different optimal bases. *Journal of the Atmospheric Sciences* 2004; **61**:2206–2217.
8. Majda AJ, Timofeyev I, Vanden-Eijnden E. Systematic strategies for stochastic mode reduction in climate. *Journal of the Atmospheric Sciences* 2003; **60**:1705–1723.
9. Lumley JL. The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation* 1967; 166–178.
10. Aubry N, Holmes P, Lumley JL. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Dynamics* 1988; **192**:115–173.
11. Holmes P, Lumley JL, Berkooz G. *Turbulence*, *Coherent Structures*, *Dynamical Systems and Symmetry*. Cambridge University Press: Cambridge, U.K., 1998.
12. Willcox K, Peraire J. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal* 2002; **40**(11):2323–2330.
13. Kirby M, Sirovich L. Application of the Karhunen–Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1990; **12**(1):103–108.
14. Sirovich L, Kirby M. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America* 1987; **A4**(3):519–524.
15. Sirovich L. Turbulence and the dynamics of coherent structures. Part II: symmetries and transformations. *Quarterly of Applied Mathematics* 1987; **45**:573–582.

DOI: 10.1002/fld

16. Sirovich L. Turbulence and the dynamics of coherent structures. Part III: dynamics and scaling. *Quarterly of Applied Mathematics* 1987; **45**:583–590.
17. Holmes P. Nonlinear dynamics, chaos and mechanics. *Applied Mechanics Reviews* 1990; **43**(5, pt 2):S23–S39.
18. Luo Z, Chen J, Zhu J, Wang R, Navon IM. An optimising reduced order FDS for the tropical Pacific ocean reduced gravity model. *International Journal for Numerical Methods in Fluids* 2007; **55**:143–161.
19. Luo Z, Zhu J, Wang R, Navon IM. Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical Pacific ocean reduced gravity model. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(41–44):4184–4195.
20. Bader B, Ghattas O, van Bloemen Waanders B, Willcox K. An optimization framework for goal-oriented, model-based reduction of large-scale systems. *44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.
21. Bui-Thanh T, Willcox K, Ghattas O, van Bloemen Waanders B. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics* 2007; **224**(2):880–896.
22. Robert C, Durbiano S, Blayo E, Verron J, Blum J, le Dimet FX. A reduced-order strategy for 4D-VAR data assimilation. *Journal of Marine Systems* 2005; **57**(1–2):70–82.
23. Hoteit I, Kohl A. Efficiency of reduced-order, time-dependent adjoint data assimilation approaches. *Journal of Oceanography* 2006; **62**(4):539–550.
24. Cao Y, Zhu J, Navon IM, Luo Z. A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids* 2007; **53**(10):1571–1583.
25. Willcox K, Megretski A. Fourier series for accurate, stable, reduced-order models in large-scale applications. *SIAM Journal on Scientific Computing* 2005; **26**(3):944–962.
26. Pain CC, Piggott MD, Goddard AJH, Fang F, Gorman GJ, Marshall DP, Eaton MD, Power PW, de Oliveira CRE. Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling* 2005; **10**(1–2):5–33.
27. Ford R, Pain CC, Piggott MD, Goddard AJH, de Oliveira CRE, Umbleby AP. A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. Part I: model formulation. *Monthly Weather Review* 2004; **132**(12):2816–2831.
28. Utku S, Clemente J, Salama M. Errors in reduction methods. *Computers and Structures* 1985; **21**(6):1153–1157.
29. Hinze M, Volkwein S. In *Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems*: *Error Estimates and Suboptimal Control in Dimension Reduction of Large-scale Systems*. Lecture Notes in Computational and Applied Mathematics, Benner P, Mehrmann V, Sorensen D (eds). Springer: Berlin, 2005.
30. Homescu C, Petzold LR, Serban R. Error estimation for reduced-order models of dynamical systems. *SIAM Journal on Numerical Analysis* 2005; **43**:1693–1714.
31. LeGresley P, Alonso J. Dynamic domain decomposition and error correction for reduced order models. *41st AIAA Aerospace Sciences Meeting & Exhibit*, *AIAA Paper 2003-0250*, Reno, NV, 6–9 January 2003.
32. Meyer M, Matthies H. Efficient model reduction in non-linear dynamics using the Karhunen–Loève expansion and dual-weighted-residual methods. *Computational Mechanics* 2003; **31**:179–191.
33. Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis* 2002; **40**:492–515.
34. Daescu DN, Navon IM. A dual-weighted approach to order reduction in 4D-Var data assimilation. *Monthly Weather Review* 2008; **136**(3):1026–1041.
35. Homescu C, Petzold LR, Serban R. Error estimation for reduced-order models of dynamical systems. *SIAM Review* 2007; **49**(2):277–299.
36. Tennekes H, Lumley JL. *A First Course in Turbulence*. MIT Press: Cambridge, MA, 1972.
37. Pain CC, Umpleby AP, de Oliveira CRE, Goddard AJH. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3771–3796.
38. Piggott MD, Gorman GJ, Pain CC, Allison PA, Candy AS, Martin BT, Wells MR. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. *International Journal for Numerical Methods in Fluids* 2008; **56**:1003–1015. DOI: 10.1002/fld.1663.
39. Tansley CE, Marshall D. Flow past a cylinder on a β plane, with application to Gulf stream separation and the Antarctic circumpolar current. *Journal of Physical Oceanography* 2001; **31**:3274–3283.
40. Iollo A, Lanteri S, Desideri JA. Stability properties of POD-Galerkin approximations for the compressible Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics* 2000; **13**:377–396.