Reduced order modelling of an adaptive mesh ocean model

F. Fang [a,*], C.C. Pain [a], I.M. Navon [b], M.D. Piggott [a], G.J. Gorman [a] P. Allison [a] A.J.H. Goddard [a]

[a] *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK.*
URL: `http://amcg.ese.imperial.ac.uk`

[b] *School of Computational Science and Department of Mathematics, Florida State University, Tallahassee, FL, 32306-4120, USA.*

Preprint submitted to ????? *6 June 2007*

Abstract

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 (hereafter, 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:

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;

To speed up the POD simulation. To achieve this a new numerical technique is introduced, whereby a time-dependent matrix in the discretised equation is rapidly constructed independent of time. This development imparts considerable efficiency gains over the oft used alternative of calculating each finite element and node over the computational domain at each time level;

To use dynamically adaptive meshes in the above POD model.

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

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 (1943)[1], Loève (1945)[2] and Karhunen (1946)[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) (Fukunaga, 1990) in statistics, and Empirical Orthogonal Functions (EOF) in oceanography ([5]; [6]) and meteorology ([7]).

Proper Orthogonal Decomposition (POD) has been widely and successfully applied to numerous fields, including signal analysis and pattern recognition ([8]), fluid dynamics and coherent structures (Lumley,1967; [9]; [10]; [11]) and image reconstruction ([12]).

* Corresponding author
Email address: f.fang@imperial.ac.uk (F. Fang).

An important innovation in 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([13],[14],[15] and [16]). 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 snapshot is minimised.

A further advance in POD technology, the gappy POD procedure, was developed to reconstruct full human facial images from partial data-sets ([17]). Gappy POD is a variation of POD proposed by Everson and Sirovich (1995)[17] allowing missing or incomplete data to be reconstructed in a lower dimensional space. In this technique, given a set of POD modes, an incomplete data vector is reconstructed by solving a small linear system. If the snapshots themselves are damaged or incomplete an iterative method is used to derive the POD basis. This methodology has also been applied to fluid dynamic applications ([18]).

POD methodologies, in combination with the Galerkin projection procedure have also been shown to provide an efficient means of generating reduced order models ([10]; [19], [20]). 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 optimise the POD bases ([21], [22]). 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. This practical utility of this approach has been extended to include ocean and climate modelling and the solution of inverse problems ([23], [24], [25] and [19]).

The motivation of the current work 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 whilst smoothly varying resolution and conforming to complex coastlines and bathymetry [26]. What distinguishes the reduced model developed here from other existing reduced models is the inclusion of 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 length at different time levels. To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced models. The solutions from the full 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 work, 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 discretised reduced model at the time level n can be written: $A^n\alpha^n = s^n$, where, α^n is a vector of the coefficients associated with the basis functions for the variables in the full model. The matrix A^n is time dependent and calculated at each element and node over the whole computational domain. Calculating the matrix A^n at each time level is compute-intensive. To accelerate the POD simulation, the 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 ([27]; [28]; [29]; [30]; [31]; [32]) to cite but a few. Early work on error in model reduction has been done by Utku (1985)[27], where the first order error estimation of the model reduction for non-linear systems is given at a small number of time steps. More recently, the Dual- Weighted-Residual method (DWR), which makes use of the solution of an adjoint system, has been incorporated into the error estimation of reduced models ([22],[31]; [29],[33]). 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. (2005, 2007) [29] [34] 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 (2005) [28] incorporated both the time derivatives and adjoint information into snapshots in the error estimation for the PDE constrained optimisation and POD inverse model. In this work, the error estimation is carried out using a simple approach described in [19], 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 $(M+1)th$ eigenvalue for AA^T.

The remainder of this paper is structured as follows: In the second section the Imperial College Ocean Model (ICOM) is briefly described. In the third section the reduced forward model is then derived, whilst the geostrophic pressure, mesh adaptivity, and acceleration of the POD simulation are discussed in detail in section four. In section five 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 work, a POD-based reduced model is developed for ICOM that can simultaneously resolve both small and large scale ocean flows whilst smoothly varying resolution and conforming to complex coastlines and bathymetry [1, 2, 3]. 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 assumption is not made in this work. 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 [2]. 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 [8]).

The underlying model equations consist of the 3-D non-hydrostatic Boussinesq equations,

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

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

where $\mathbf{u} \equiv (u, v, w)^T \equiv (u_1, u_2, u_3)^T$ is the velocity vector, $\mathbf{x} \equiv (x, y, z)^T \equiv (x_1, x_2, x_3)^T$ 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 $\mathbf{k} = (0, 0, 1)^T$. The stress tensor τ is used to represent viscous terms and is defined in terms of the deformation rate tensor $\mathbf{S}$ as

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

with no summation over repeated indices. In this work the horizontal kinematic viscosities (μ_{11}, μ_{22}) and vertical kinematic viscosity (μ_{33}) take constant values with the off-diagonal components of τ defined by $\mu_{ij} = (\mu_{ii}\mu_{jj})^{1/2}$, see also [35, 36]. For barotropic flow (baroclinic flow is incorporated in section 4), the pressure p consists of hydrostatic $p_h(z)$ and non-hydrostatic $p_{nh}(x, y, z, t)$ components. The hydrostatic component of pressure balances exactly the constant buoyancy force and both terms are therefore dropped at this stage.

ICOM utilises dynamic adaptation of a fully unstructured tetrahedral mesh in three- dimensions (3-D), as presented in Pain et al. (2001) [37]. This technique uses a form of h-refinement (or mesh optimisation) to adapt the mesh, changing the size, shape and location of tetrahedral elements to optimise 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–node smoothing, edge and face-edge swapping, and edge splitting and collapsing–to minimise 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. (2001) [37]. An alternate approach of defining an error measure to guide an adaptive meshing algorithm for unstructured tetrahedral finite elements is to utilise an adjoint or goal-based method. This method is based upon 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.

3 Reduced order ocean model

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

3.1 Discretised ocean model

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

$$u_{x,y,z} = \sum_{i=1}^{\mathcal{N}} u_i N_i, \quad v_{x,y,z} = \sum_{i=1}^{\mathcal{N}} v_i N_i, \quad w_{x,y,z} = \sum_{i=1}^{\mathcal{N}} w_i N_i,$$
$$p_{ng} = \sum_{i=1}^{\mathcal{N}} p_{ng,i} N_i, \quad p_g = \sum_{i=1}^{\mathcal{N}} p_{g,i} M_i, \tag{3}$$

where, $\mathcal{N}$ is the number of nodes, $p_n g$ and p_g are the non-geostrophic and geostrophic pressures, respectively.

[Fig. 1 about here.]

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 variables u, v, w, p. The average of the ensemble of snapshots is defined as:

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

Taking the deviation from the mean of variables,forms

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

A collection of all $V_{k,i}$ constructs a rectangular $\mathcal{N}$ by K matrix A. The goal of Proper Orthogonal Decomposition (POD) is to find a set of orthogonal basis functions $\Phi = \Phi_1, \Phi_2, \ldots, \Phi_K$ such that it maximise

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

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 optimisation problem (6). From SVD, the matrix $A \in R^{\mathcal{N} \times K}$ can be expressed

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

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

$$A^T A y_k = \lambda_k y_k; \quad 1 \leq k \leq K. \tag{9}$$

This procedure is equivalent to a Singular Value Decomposition (SVD). The eigenvalues λ_k 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:

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

It can be shown [6,11], that the kth eigenvalue is a measure of the kinetic energy transferred within the kth basis mode (strictly speaking this is applied, when the field under consideration is the velocity field, but can be generalised to others fields as well). 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 realisations 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 ($M << K$),

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

subject to

$$M = argmin\{I(M) : I(M) \geq \gamma\}, \tag{12}$$

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

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,

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

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:

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

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

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

The errors for the above POD model are bounded by the following expression (details of the derivation can be found in Luo et al. (2007) [19]):

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

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

4 Geostrophic pressure, adaptive meshes and Efficiency in POD simulation

4.1 Geostrophic pressure

One important issue in ocean modelling is the treatment of Coriolis term in the momentum equation. To allow the accurate representation of the geostrophic pressure, the pressure in (2) is divided into two parts: $p = p_{ng} + p_g$. The geostrophic pressure has to satisfy the geostrophic balance:

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

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

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

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) whilst 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 gradient iterative method:

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

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, M is the total number of basis functions. The geostrophic pressure can therefore be expressed as:

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

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, $\bar{u} = \sum_{k=1}^{K} u_K$and $\bar{v} = \sum_{k=1}^{K} v_k$. It is shown in Figure 2 that the numerical results are significantly improved by using the new numerical method described above. This method can

also be easily extended to represent buoyancy with temperature and salinity dependence by introducing more basis functions for which balance these buoyancy terms.

[Fig. 2 about here.]

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 length 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 obtaining the same length of base modes 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, $\alpha_m^n = (\alpha_{m,u}^n, \alpha_{m,v}^n, \alpha_{m,w}^n, \alpha_{m,p}^n)$, s^n is a discretised 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, the matrices $\mathbf{A}^n$ and $\mathbf{B}^n$ can be written 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, the matrices $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$ $\hat{\mathbf{A}}_m$ and $\hat{\mathbf{B}}_m$ are time independent, and derived in the Appendix A. Equations (24) and (25) can be rewritten as:

$$\mathbf{A}^n = \hat{\mathbf{A}}_0 + \hat{\mathbf{A}}^n \alpha^{n-1}, \tag{26}$$

$$\mathbf{B}^n = \hat{\mathbf{B}}_0 + \hat{\mathbf{B}}^n \alpha^{n-1}, \tag{27}$$

where, $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_m, \ldots, \hat{\mathbf{A}}_M)$, $\hat{\mathbf{B}} = (\hat{\mathbf{B}}_1, \ldots, \hat{\mathbf{B}}_m, \ldots, \hat{\mathbf{B}}_M)$, which are independent of time. 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.

5 Application cases and numerical results

The utility of the new POD reduced order model is herein assessed and validated in three 2D, time-dependent test case; - flow past a cylinder, flow past a cylinder on a β plane and a gyre.

5.1 Case 1: flow past a cylinder

The 2D case is composed of a cylinder with a radius of 1 in the computational domain i 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 and the boundary condition applied to the cylinder and both lateral sides is set to no-slip 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 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, 20 snapshots and 10 basis functions are chosen for u, v, w and p, for which 95 percent 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 nearby 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 $t = 10$. 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 $max\{\sqrt{\lambda_{u11}}, \sqrt{\lambda_{v11}}\} \leq 0.8$ (here, λ_{u11} and λ_{v11} are the 11th eigenvalues respectively for the velocity components u and v. The whole simulation of running the reduced model is completed within 9 minutes, whilst it takes 40 minutes of CPU time for running the full forward model.

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 that in the previous 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 Reynolds number is $Re = 200$. The spin-up period is 0.4. The simulation period is 2 and the time step is chosen to be 0.002.

Figure 3 shows the velocity field (vector) obtained from the full (left panel) and POD (right panel) models. It can be seen from both the full and reduced simulations that two separated jets form downstream of the cylinder, as described in reference **??**. 40 snapshots and 30 basis functions, here, are chosen for u, v, w and p, in which 99.9999 percent of energy is captured. The maximum error of numerical POD solutions is less than $max\{\sqrt{\lambda_{u31}}, \sqrt{\lambda_{v31}}\} \leq 10^{-3}$ (here, λ_{u31} and λ_{v31} are the 31th eigenvalues respectively for the velocity components u and v. It takes only 3 hrs to complete the simulation by running the reduced model which is considerably less than the CPU time (38-39 hrs) required to run the full forward model. The computational cost is thus significantly reduced by 92% (i.e. by a factor of 13).

[Fig. 3 about here.]

[Fig. 4 about here.]

[Fig. 5 about here.]

5.3 Case 3: Gyre

To simulate a wind driven barotropic circulation, the computational domain is taken to be a square box of $1000km$ with a depth of 500 m. A maximum zonal wind stress of $\tau_0 = 0.1Nm^{-1}$ is applied in a cosine of latitude profile. The Reynolds number is $Re = 250$, $\beta = 1.8 \times 10^{-11}$ and the reference density $\rho_0 = 1000\ kgm^{-1}$. The problem is non-dimensionalised, so that the domain is a box of 1, and a depth of 0.0005 with one element in the vertical. Incorporating the $\beta-$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 6 hrs. No-slip boundary conditions are applied to the lateral boundaries. The spin-up period is 0.3024 (200 $days$). The simulation period is $[200, 400]$ $days$.

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 work, to accurately represent the geostrophic balance, the geostrophic pressure is calculated using the novel approach described in section 4. Comparison of velocity field between the full and POD models is provided in figure 6. The velocity fields at the different time levels exhibit an overall good agreement with those from the full model.

[Fig. 6 about here.]

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 1. 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 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 2. It usually takes less than 3 hrs to complete the simulation by running the reduced model. For all the setting, compared with the CPU (30 hrs) required for running the full model, the CPU for running the reduced model decreases by a factor of 10 whilst about 98.5% of the energy is captured. It is noted that that 99% of percent of CPU required for the reduced model is used to setup the POD bases and calculate the time-independent sub-matrices (details in the section 4.3) for preparation of running the simulation. The actual CPU time required to running the reduced model during the simulation period is less than 1 minute.

[Table 1 about here.]

[Table 2 about here.]

The error of numerical results obtained by a different number of POD bases can be calculated as in equation (16). Figure 7 illustrates the eigenvalues and the error associated with a corresponding number of POD bases for the velocity components u and v. It is indicated that the first 25% of the leading POD bases has a significant impact on POD results. The error of POD results decreases by $70\% - 80\%$ of its original values whilst the energy captured can achieve above 76% if the first 25% of the leading POD bases are chosen.

The Root Mean Squared Error (RMSE) and correlation coefficient of results between the full and POD models at different time levels are provided in figures 9 and 8. It can be seen that as the simulated time accrues, the POD error increases, whilst 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 ms^{-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 10).

[Fig. 7 about here.]

[Fig. 8 about here.]

[Fig. 9 about here.]

[Fig. 10 about here.]

6 Summary and conclusions

A POD reduced model has been developed for a finite element adaptive mesh refinement ocean model (here, ICOM). To the best of our knowledge, this is the first attempt to apply the POD approach 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. 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, a H^1 norm is suggested to redefine the norms involved in the POD definition so that derivatives of the snapshots as well as that of the basis functions are included in the POD average (Iollo et al. (2000)) [38].

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 by 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 $80-99\%$, whilst the RMSE results is less than 2.5 ms^{-1} and 97% of energy can be captured if a suitable number of POD bases is chosen (say, if half of the leading POD bases is chosen with 81 snapshots). The error for leading POD results decreases by $70\%-80\%$ of its original values if 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 discretised POD equations prior to running the reduced model. It is proved that the computer cost required for running the reduced model can decreased by a factor of 30.

Further research will address the following issues of a goal oriented POD to optimise 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 controlls.

Appendix A

The second-order Crank-Nicolson time stepping algorithm is used, and the sub-matrices are: (1) the A_0 matrices in u momentum equation;

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

(2) the A_0 matrices in v momentum equation;

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

(3) the A_0 matrices in w momentum equation;

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

(4) the A_0 matrices in continuity equation;

$$
\begin{aligned}
\hat{A}_{0,3\mathcal{M}+i,j} = & \sum_{k=1}^{N} \frac{\partial \Phi_{u,j,k}}{\partial x} \\
\hat{A}_{0,3\mathcal{M}+i,\mathcal{M}+j} = & \sum_{k=1}^{\mathcal{N}} \frac{\partial \Phi_{v,j,k}}{\partial y} \\
\hat{A}_{0,3\mathcal{M}+i,2\mathcal{M}+j} = & \sum_{k=1}^{\mathcal{N}} \frac{\partial \Phi_{w,j,k}}{\partial z}
\end{aligned} \tag{A-4}
$$

(5) the B_0 matrices in u momentum equation;

$$
\begin{aligned}
\hat{B}_{0,i,j} = & \sum_{k=1}^{N} u^{n-1} - \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \left[\bar{u}_k \frac{\partial \Phi_{u,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{u,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{u,j,k}}{\partial z} + \right. \\
& \left. \frac{\partial^2 \Phi_{u,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{u,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{u,j,k}}{\partial z^2} + \frac{\partial \bar{u}_k}{\partial x} \Phi_{u,j,k} \right], \\
\hat{B}_{0,i,\mathcal{M}+j} = & \sum_{k=1}^{\mathcal{N}} \left[- f \Phi_{v,j,k} + \frac{\partial \bar{u}_k}{\partial y} \Phi_{v,j,k} \right] \\
\hat{B}_{0,i,2\mathcal{M}+j} = & \frac{\partial \bar{u}_k}{\partial z} \Phi_{w,j,k} \\
\hat{B}_{0,i,3\mathcal{M}+j} = & \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \frac{\partial p_k}{\partial x},
\end{aligned} \tag{A-5}
$$

(6) the B_0 matrices in v momentum equation;

$$
\begin{aligned}
\hat{B}_{0,\mathcal{M}+i,j} = & \sum_{k=1}^{N} \left[f \Phi_{u,j,k}) + \frac{\partial \bar{v}_k}{\partial x} \Phi_{u,j,k} \right] \\
\hat{B}_{0,\mathcal{M}+i,\mathcal{M}+j} = & \sum_{k=1}^{\mathcal{N}} v^{n-1} - \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \left[\bar{u}_k \frac{\partial \Phi_{v,j,k}}{\partial x} + \bar{v}_k \frac{\partial \Phi_{v,j,k}}{\partial y} + \bar{w}_k \frac{\partial \Phi_{v,j,k}}{\partial z} + \right. \\
& \left. \frac{\partial^2 \Phi_{v,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{v,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{v,j,k}}{\partial z^2} + \frac{\partial \bar{v}_k}{\partial y} \Phi_{v,j,k} \right], \\
\hat{B}_{0,\mathcal{M}+i,2\mathcal{M}+j} = & \frac{\partial \bar{v}_k}{\partial z} \Phi_{w,j,k} \\
\hat{B}_{0,\mathcal{M}+i,3\mathcal{M}+j} = & \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \frac{\partial p_k}{\partial y},
\end{aligned} \tag{A-6}
$$

(7) the B_0 matrices in w momentum equation;

$$
\begin{aligned}
\hat{B}_{0,2\mathcal{M}+i,j} &= \frac{\partial \bar{w}_k}{\partial x}\Phi_{u,j,k} \\
\hat{B}_{0,2\mathcal{M}+i,\mathcal{M}+j} &= \frac{\partial \bar{w}_k}{\partial y}\Phi_{v,j,k} \\
\hat{B}_{0,2\mathcal{M}+i,2\mathcal{M}+j} &= \sum_{k=1}^{N} w^{n-1} - \frac{1}{2}\Delta t \sum_{k=1}^{N}\Bigg[\bar{u}_k\frac{\partial \Phi_{w,j,k}}{\partial x} + \bar{v}_k\frac{\partial \Phi_{w,j,k}}{\partial y} + \bar{w}_k\frac{\partial \Phi_{w,j,k}}{\partial z} + \\
&\qquad \frac{\partial^2 \Phi_{w,j,k}}{\partial x^2} + \frac{\partial^2 \Phi_{w,j,k}}{\partial y^2} + \frac{\partial^2 \Phi_{w,j,k}}{\partial z^2} + \frac{\partial \bar{w}_k}{\partial z}\Phi_{w,j,k}\Bigg], \\
\hat{B}_{0,2\mathcal{M}+i,3\mathcal{M}+j} &= \frac{1}{2}\Delta t \sum_{k=1}^{\mathcal{N}} \frac{\partial p_k}{\partial z},
\end{aligned}
\tag{A-7}
$$

(8) the A_m matrices;

$$
\begin{aligned}
\hat{A}_{i,j} &= \frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\left[\Phi_{u,l,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,l,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{A}_{\mathcal{M}+i,\mathcal{M}+j} &= \frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\left[\Phi_{u,l,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,l,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{A}_{2\mathcal{M}+i,2\mathcal{M}+j} &= \frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\left[\Phi_{u,l,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,l,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right]
\end{aligned}
\tag{A-8}
$$

(9) the B_m matrices;

$$
\begin{aligned}
\hat{B}_{i,j} &= -\frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\left[\Phi_{u,l,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,l,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{B}_{\mathcal{M}+i,\mathcal{M}+j} &= -\frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\left[\Phi_{u,l,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,l,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right] \\
\hat{B}_{2\mathcal{M}+i,2\mathcal{M}+j} &= -\frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\left[\Phi_{u,l,k}\frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k}\frac{\partial \Phi_{u,j,k}}{\partial y}\Phi_{w,l,k}\frac{\partial \Phi_{u,j,k}}{\partial z}\right]
\end{aligned}
\tag{A-9}
$$

where, $\mathcal{M}$ represents the number of basis functions, $\mathcal{N}$ is the number of nodes, $1 \leq i, j \leq M$.

Acknowledgements

This work was carried out under funding from ??. Prof. I. M. Navon would like to acknowledgement support of NSF grant ATM-0201808.

References

[1] D. D. Kosambi. Statistics in function space. *J. Indian Math. Soc.*, 7:76–88, 1943.
[2] M. Loève. Fonctions aleatoires de second ordre. *Compte Rend. Acad.Sci.*, 220, 1945.
[3] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae*, 37, 1946.
[4] C. G. Wu, Y. C. Liang, W. Z. Lin, H. P. Lee, and S. P. Lim. A note on equivalence of proper orthogonal decomposition methods. *Journal of Sound and Vibration*, 265:1103–1110, 2003.
[5] I. T. Jolliffe. *Principal Component Analysis*. Springer: Berlin, 2002.
[6] D. T. Crommelin and A. J. Majda. Strategies for model reduction: Comparing different optimal bases. *Journal of the Atmos. Sci.*, 61:2206–2217, 2004.
[7] A. J. Majda, I. Timofeyev, and E. Vanden-Eijnden. Systematic strategies for stochastic mode reduction in climate. *Journal of the Atmospheric Sciences*.
[8] K. Fukunaga. *Introduction to Statistical Recognition Second Edition (Computer Science and Scientific Computing Series)*. Boston: Academic Press, 1990.
[9] N. Aubry, P. Holmes, and J. L. Lumley. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Dynamics*, 192:115–173, 1988.
[10] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge, UK: Cambridge University Press, 1998.
[11] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.
[12] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.
[13] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J Opt Soc Amer*, A4(3):519–524, 1987.
[14] L. Sirovich. Turbulence and the dynamics of coherent structures, part ii: Symmetries and transformations. *Quart Appl. Math.*, 45:573–582, 1987.
[15] L. Sirovich. Turbulence and the dynamics of coherent structures, part iii: Dynamics and scaling. *Quart Appl. Math.*, 45:583–590, 1987.
[16] P. Holmes. Nonlinear dynamics, chaos and mechanics. *Applied Mechanics Reviews*, 43, 5(2):S23–S39, 996.
[17] R. Everson and L. Sirovich. Karhunen-Loève procedure for gappy data. *J. Opt. Soc. Am.*, A 12:1657– 1664, 1995.
[18] T. Bui-Thanh, M. Damodaran, and K. Willcox. Reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.

[19] Z. Luo, J. Chen, J. Zhu, R. Wang, and I. M. Navon. An optimizing reduced order FDS for the tropical Pacific ocean reduced gravity model. *International Journal for Numerical Methods in Fluids*, DOI: 10.1002/fld.1452, 2007.

[20] Z. Luo, J. Zhu, R. Wang, and I. M. Navon. 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*, in press, 2007.

[21] K. Willcox, O. Ghattas, B. van Bloemen Waanders, and B.Bader. An optimization framework for goal-oriented, model-based reduction of large-scale systems. *44th IEEE Conference on Decision and Control and European Control Conference*, 2005.

[22] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *J. of Computational Physics*, 224(2):880–896, 2007.

[23] C. Robert, S. Durbiano, E. Blayo, J. Verron, J. Blum, and F. X. le Dimet. A reduced-order strategy for 4D-VAR data assimilation. *Journal of Marine Systems*, 57(1-2):70–82, 2005.

[24] I. Hoteit and A. Kohl. Efficiency of reduced-order, time-dependent adjoint data assimilation approaches. *J. of Oceanography*, 62(4):539–550, 2006.

[25] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo. A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 53(10):1571–1583, 2006.

[26] C. C. Pain, M. D. Piggott, A. J. H. Goddard, F. Fang, G. J. Gorman, D. P. Marshall, M. D. Eaton, P. W. Power, and C. R. E. de Oliveira. Three-dimensional unstructured mesh ocean modelling. *Ocean Modell.*, 10(1–2):5–33, 2005.

[27] S. Utku, J. Clemente, and M. Salama. Errors in reduction methods. *Computers and Structures*, 21(6):1153–1157, 1985.

[28] M. Hinze and S. Volkwein. *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, P. Benner, V. Mehrmann, D. Sorensen (Eds), 2005.

[29] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *Siam Journal on Numerical Analysis*, 43:1693–1714, 2005.

[30] P. LeGresley and J. Alonso. Dynamic domain decomposition and error correction for reduced order models. In *AIAA 41st Aerospace Sciences Meeting*, volume 2003-0250. 2003.

[31] M. Meyer and H. Matthies. Efficient model reduction in non-linear dynamics using the Karhunen-loève expansion and dual-weighted-residual methods. *Comput. Mech.*, 31:179–191, 2003.

[32] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40:492–515, 2002.

[33] D. N. Daescu and I. M. Navon. A dual-weighted approach to order reduction in 4d-var data assimilation. *Sub-judice in MWR*, 2007.

[34] C. Homescu, L.R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *Siam Review*, 49(2):277–299, 2007.

[35] H. Tennekes and J. L. Lumley. *A First Course in Turbulence.* MIT Press, Cambridge, MA, 1972.

[36] R. Ford, C. C. Pain, M. D. Piggott, A. J. H. Goddard, C. R. E. de Oliveira, and A. P.

Umbleby. A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. Part I: Model formulation. *Mon. Weath. Rev.*, 132(12):2816–2831, 2004.

[37] C. C. Pain, A. P. Umpleby, C. R. E. de Oliveira, and A. J. H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Comput. Methods Appl. Mech. Engrg.*, 190:3771–3796, 2001.

[38] A. Iollo, S. Lanteri, and J. A. Desideri. Stability properties of POD-Galerkin approximations for the compressible Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*, 13:377–396, 2000.

List of Figures

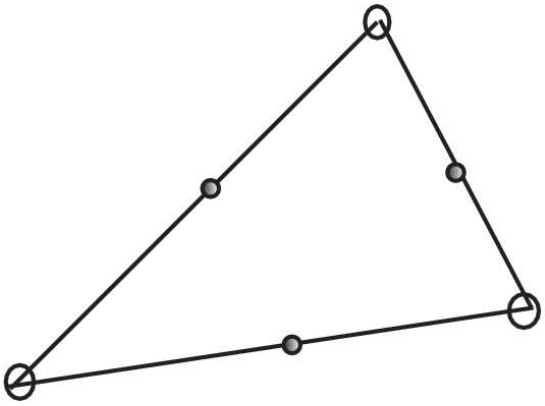

Fig. 1. Linear and quadratic mixed finite element. •: u, v, w, p_{ng}; ∘: p_g

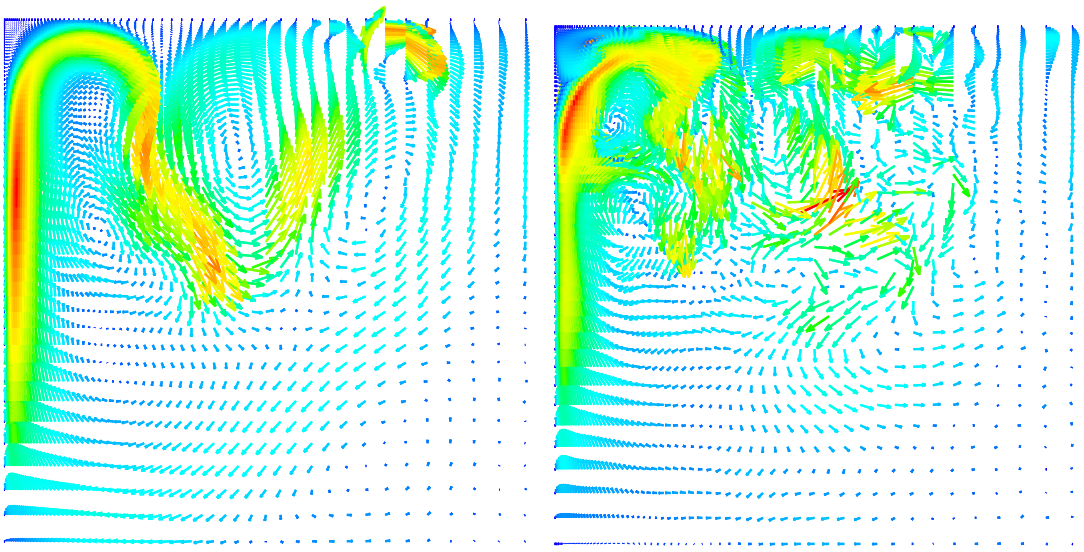

Fig. 2. Effect of geostrophic balance on results from POD simulation (left panel: with the new geostrophic pressure method; right panel: without the new geostrophic pressure method).

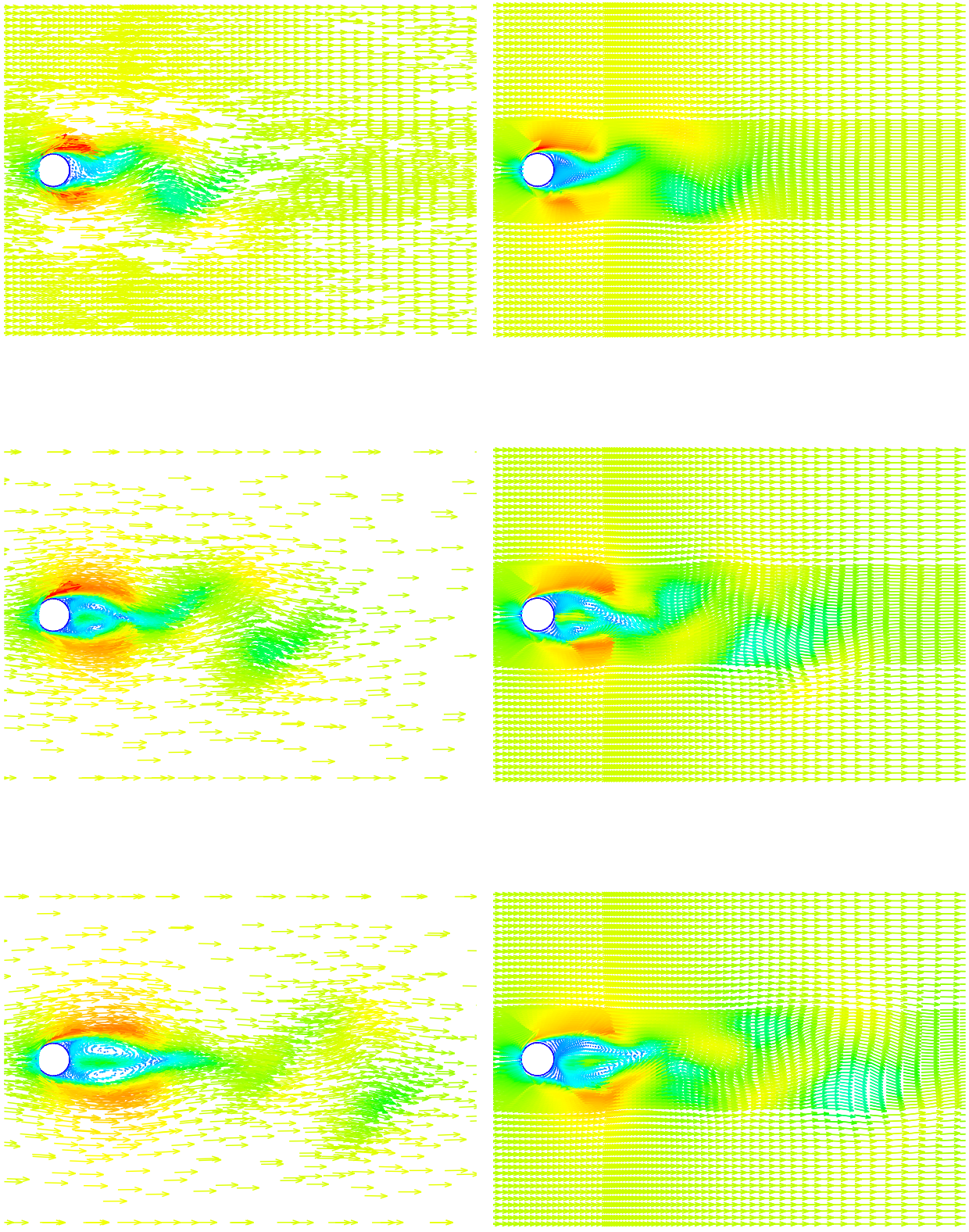

Fig. 3. Case1: comparison of velocity field between the full and reduced models ($Re = 100$) (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t = 8$; middle panel: at the time level $t = 10$; bottom panel: at the time level $t = 12$).

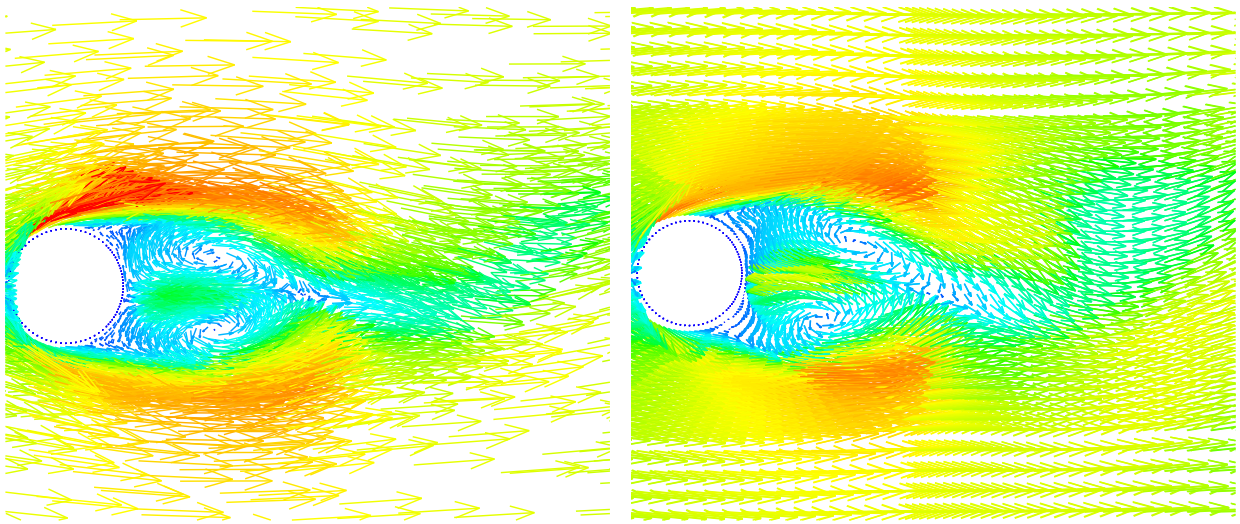

Fig. 4. Case1: Blowup of the velocity field in figure 3 around the cylinder at the time level $t = 10$. (left panel: the full model; right panel: the reduced model)

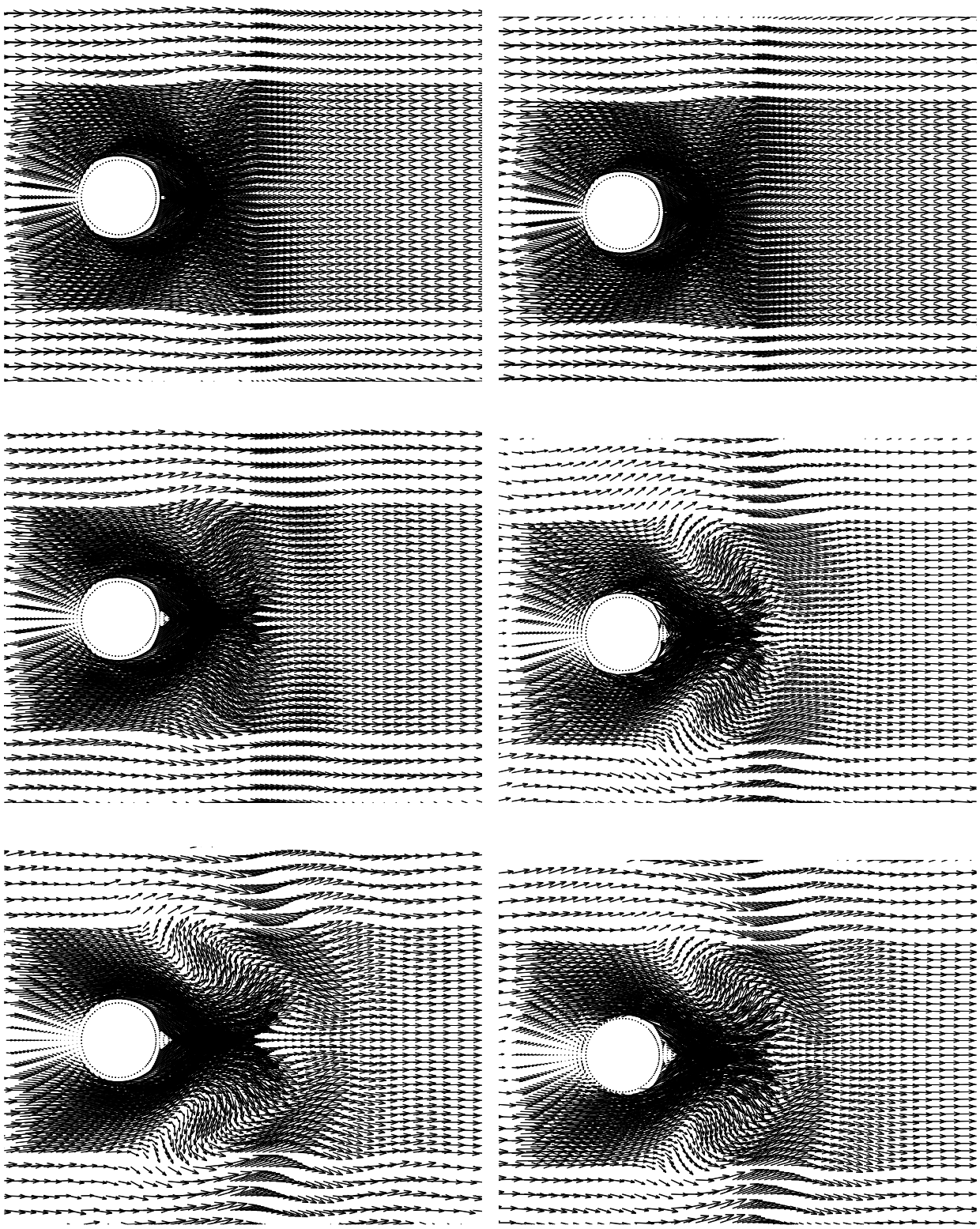

Fig. 5. Case2–cylinder on a β plane ($\beta = 7.5$, $Re = 100$): comparison of velocity field between the full and reduced models (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t = 0.4$; middle panel: at the time level $t = 0.8$; bottom panel: at the time level $t = 2$).

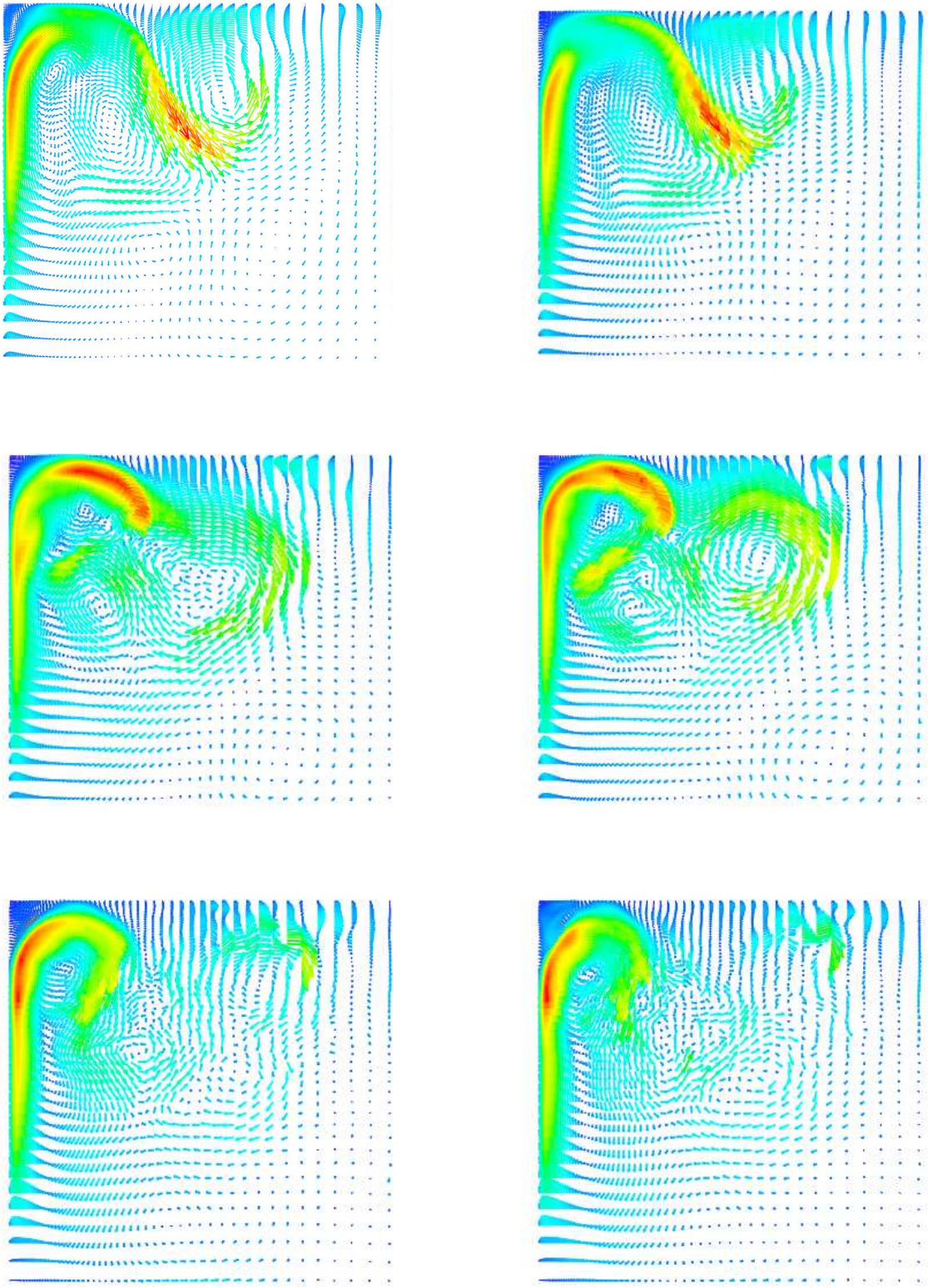

Fig. 6. Case3: comparison of velocity field between the full and reduced models ($Re = 250$) (left panel: the full model; right panel: the reduced model; top panel: at the time level $t = 200$ $days$; middle panel: at the time level $t = 300$ $days$; bottom panel: at the time level $t = 400$ $days$.

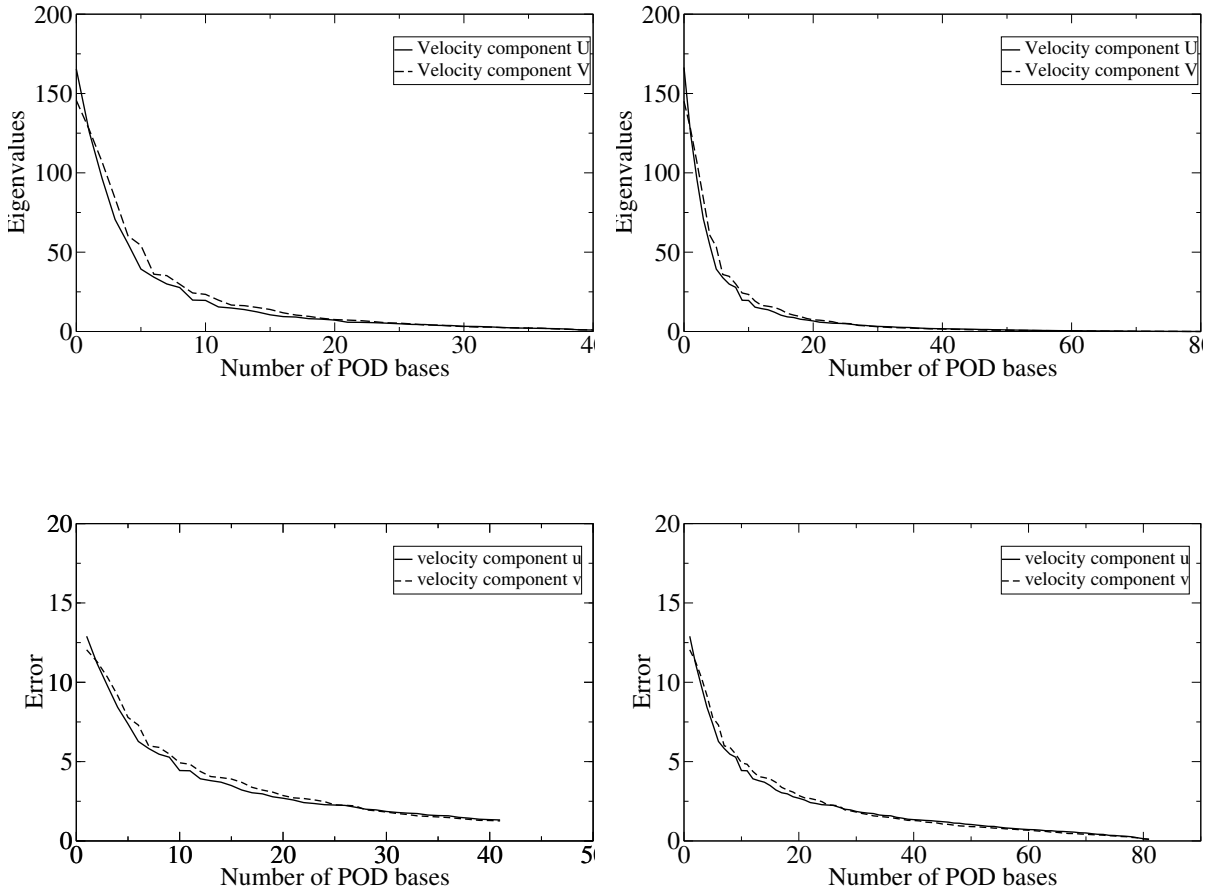

Fig. 7. Case3: Eigenvalues and errors for velocity components u and v (top panel: eigenvalues; bottom panel: error left panel: 41 snapshots; right panel: 81 snapshots).

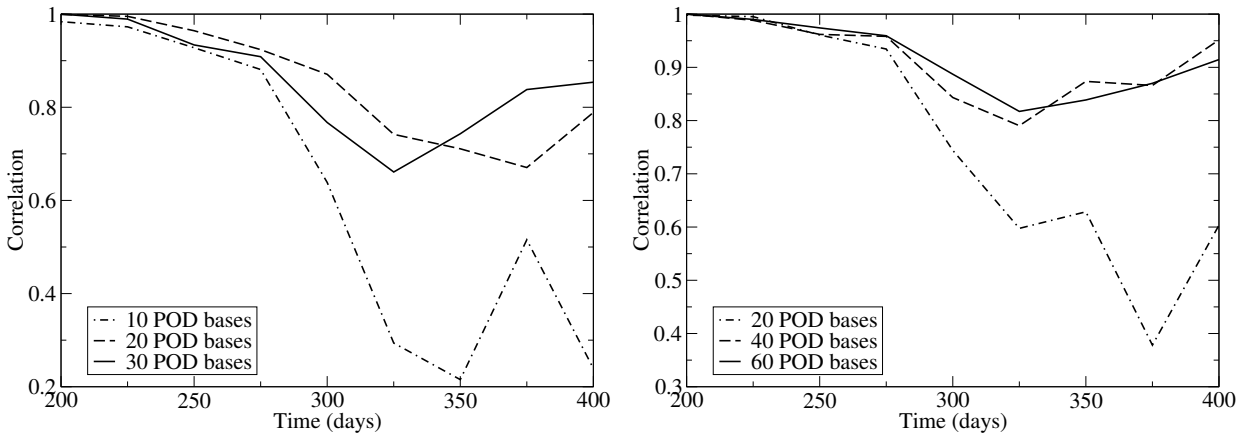

Fig. 8. Case3: Correlation at time levels (left panel: 41 snapshots; right panel: 81 snapshots).

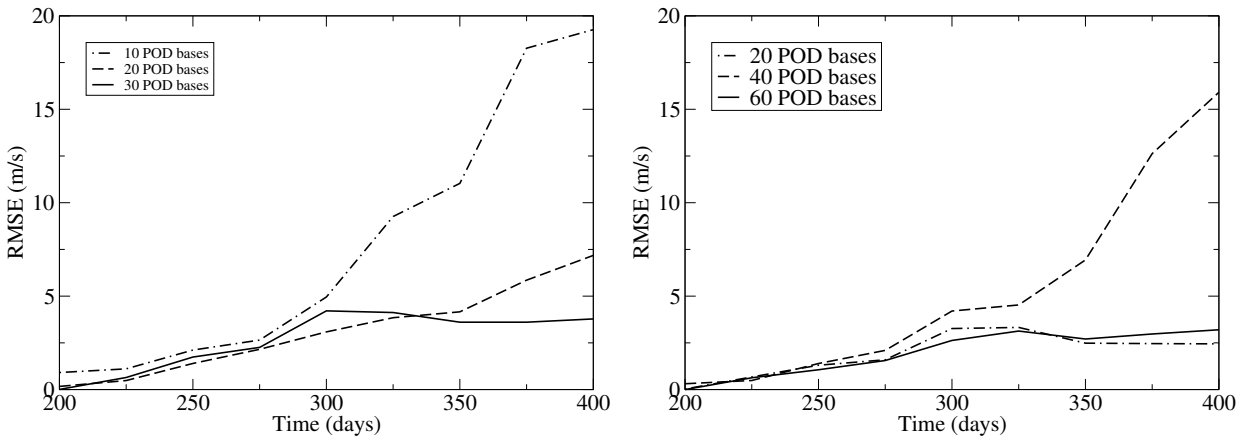

Fig. 9. Case3: RMS at time levels (left panel: 41 snapshots; right panel: 81 snapshots).

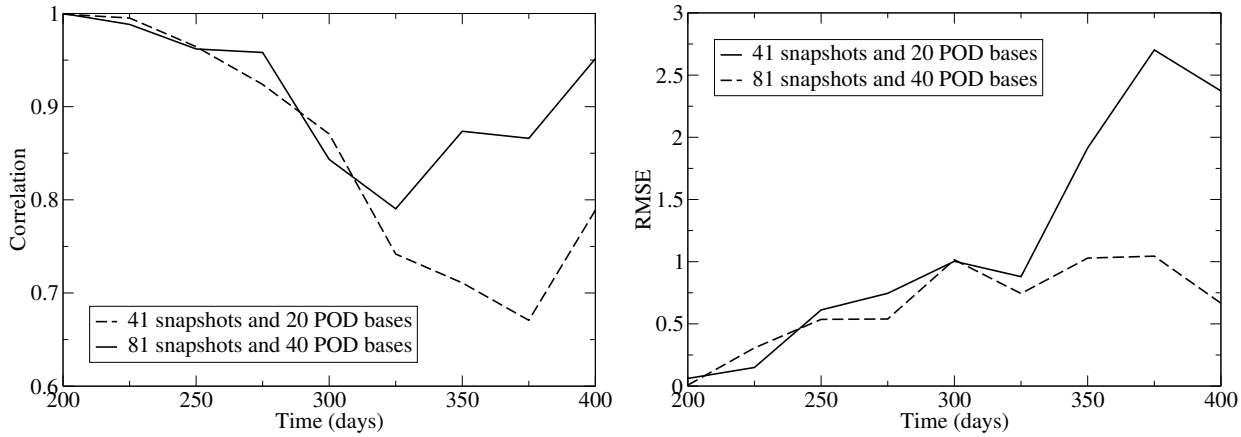

Fig. 10. Case3: Comparison of correlation and RMS with different snapshot numbers (left panel: Correlation; right panel: RMS).

List of Tables

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

Table 1: Energy captured by the POD bases for velocity components, u, v and pressure p.

number of POD bases	CPU (hrs) (41 snapshots)	CPU (hrs) (81 snapshots)
10 for 41 bases 20 for 81 bases	0.77 (reduced by 97%)	1.4 (reduced by 95%)
20 for 41 bases 40 for 81 bases	1.30 (reduced by 95%)	2.47 (reduced by 92%)
30 for 41 bases 60 for 81 bases	2.00 (reduced by 93%)	11.0 (reduced by 63%)

Table 2: a list of CPU times required for running the reduced model and the reduced percent of CPU compared with that (30 hrs) required for running the full model.Note the actual CPU time required to running the reduced model during the simulation period is less than 1 Minuit after the POD bases and the time-independent sub-matrices (section 4.3) are calculated.