

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.es.e.imperial.ac.uk>*

^b*School of Computational Science
and Dept of Mathematics, Florida State University
Tallahassee FL, 32306-4120. USA.*

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 Kosami (1943), Loeve (1945) and Karhunen (1946), and is alternatively known as the Karhunen-Loeve decomposition (KLD) method. Related methodologies have, however, been developed in a variety of disparate disciplines (Wu, 2003). The procedure is also known as Principal Components Analysis (PCA) (Fukunaga, 1990) in statistics, and Empirical Orthogonal Functions (EOF) in oceanography (Jolliffe and Majda, 2002; Cormmelin and Majda, 2004) and meteorology (Majda et al, 2003).

Proper Orthogonal Decomposition (POD) has been widely and successfully applied to numerous fields, including signal analysis and pattern recognition (Fukunaga, 1990), fluid dynamics and coherent structures (Lumley, 1967; Aubry et al. 1988; Holmes et al., 1996; Willcox et al, 2002) and image reconstruction (Kirby and Sirovich, 1990).

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

An important innovation in the use of POD for large problems in fluid dynamics involves using a series of "snapshots" to reduce the order of POD eigenvalues (Sirovich, 1987, 1989, 1990 and Holmes, 1990). The "snapshots" are essentially a set of instantaneous flow solutions, obtained from experimental data or a CFD simulation. They are then 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 minimized.

A further advance in POD technology, the gappy POD procedure, was developed to reconstruct full human facial images from partial data-sets (Eversone and Sirovich, 1995). 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 (Thanh et al, 2004)

POD methodologies, in combination with the Galerkin projection procedure have also been shown to provide an efficient means of generating reduced order models (Holmes, 1996; Luo et al, 2006, 2007). 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 (Willcox et al, 2005, Blayo et al, 1998). This practical utility of this approach has been extended to include ocean and climate modelling and the solution of inverse problems (Robert et al., 2005; Hoteit, 2004; Cao et al., 2006; and Luo, 2006).

The motivation of the current work is to develop a POD-based reduced model for an unstructured ocean model, i.e., ICOM, that can simultaneously resolve both small and large scale ocean flows whilst smoothly varying resolution and conforming to complex coastlines and bathymetry (Pain et al, 2005). What distinguishes the reduced model developed here from other existing reduced models is the inclusion of adaptive meshes. 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. The **referenced** mesh can also be obtained by superimposing the resolution at each mesh level associated with a goal-based function.

One of the important issues in reducing ocean models is to accurately represent the geostrophic balance. In this work, the pressure 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 ; v respectively. 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 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. It costs a lot of CPU to calculate the matrix A^n at each time level. To accelerate the POD simulation, the matrix A^n is constructed by a set of time-independent matrices which are obtained before running the reduced model.

Error estimation is a critical issue in reduced modelling. The references related to error estimation can be found in (Utku, 1985; Hinze and Volkwein, 2005; Homescu et al, 2005; Legresley and Alonso, 2003; Meyer and Matthies, 2003; Kunisch and S. Volkwein, 2002). Early work has been done by Utku (1985), where the first order error estimation of the model reduction for non-linear systems were given at a small number of time steps (during which the Jacobian matrix can be considered constant). 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 (Meyer and Matthies, 2003; Homescu et al, 2005). Using this method it is possible to obtain an a 'priori' error estimate for a certain cost functional of the solution. This error estimate can be used for adaptively resizing the number of basis vectors and the length of the time step 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) 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) incorporated both the time derivatives and adjoint information into snapshots in the error estimation for the PDF constrained optimization and POD inverse model. In this work, the error estimation is carried out using a simple approach described in Luo (2007), where, a spectral norm A_2 is defined to estimate the spatial error between the full and reduced models. A error bound is given by the (maximum eigenvalues of $A^T A$) $^{0.5}$.

The remainder of this paper is structured as follows: In the second section 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 details in section four. In section six the above reduced model is applied to some relevant cases. 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 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 pressure and velocity 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,

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

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

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 $\boldsymbol{\tau}$ is used to represent viscous terms and is defined in terms of the deformation rate tensor \mathbf{S} as

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

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 $\boldsymbol{\tau}$ defined by $\mu_{ij} = (\mu_{ii}\mu_{jj})^{1/2}$, see also [? ?]. 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). 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). 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 the quadratic basis function M for the geostrophic pressure (Figure 1). The variables to be solved can be expressed in the finite element form:

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

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

[Fig. 1 about here.]

3.2 Proper Orthogonal Decomposition

The variables (e.g., u, v, w, p) are sampled at defined checkpoints during the simulation period $[t_1, \dots, t_n, \dots, t_K]$, also referred to as snapshots (K is the number of snapshots). The samples at 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 at a vector U_i with \mathcal{N} entries (\mathcal{N} is the number of nodes), here, U can represent one of variables u, v, w, p . The average of the ensemble of snapshots is defined:

$$\bar{U} = \frac{1}{K} \sum_{i=1}^{\mathcal{N}} U_i, \quad 1 \leq i \leq K, \quad (4)$$

Taking deviation from the mean of variables, forms

$$V_i = U_i - \bar{U}, \quad 1 \leq i \leq K, \quad (5)$$

A collection of all V_i constructs a rectangular N by K matrix Y . The order K for matrix YY^T is far larger than the order K for matrix Y^TY . Therefore a $K \times K$ eigenvalue problem is solved

$$Y^TY y_k = \lambda_k y_k; \quad 1 \leq k \leq K. \quad (6)$$

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:

$$\Phi_k = Y y_k / \sqrt{\lambda_k} \quad (7)$$

It can be shown [6,11], that the k th eigenvalue is a measure of the kinetic energy transferred within k th 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 ($M \ll K$),

$$I(M) = \frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^K \lambda_i} \quad (8)$$

subject to

$$M = \operatorname{argmin}\{I(M) : I(M) \geq \gamma\} \quad (9)$$

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

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,

$$\begin{aligned}
u(t, x, y, z) &= \bar{u} + \sum_{m=1}^{M_u} \alpha_{m,u}(t) \Phi_{m,u}(t, x, y, z) \\
v(t, x, y, z) &= \bar{v} + \sum_{m=1}^{M_v} \alpha_{m,v}(t) \Phi_{m,v}(t, x, y, z) \\
w(t, x, y, z) &= \bar{w} + \sum_{m=1}^{M_w} \alpha_{m,w}(t) \Phi_{m,w}(t, x, y, z) \\
p(t, x, y, z) &= \bar{p} + \sum_{m=1}^{M_p} \alpha_{m,p}(t) \Phi_{m,p}(t, x, y, z)
\end{aligned} \tag{10}$$

Substituting (10) 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:

$$\begin{aligned}
\frac{\partial \alpha_{m_u,u}}{\partial t} &= F_u(t, \alpha_{1,u}, \dots, \alpha_{M_u,u}, \alpha_{1,v}, \dots, \alpha_{M_v,v}, \alpha_{1,w}, \dots, \alpha_{M_w,w}, \alpha_{1,p}, \dots, \alpha_{M_p,p}) \\
\frac{\partial \alpha_{m_v,v}}{\partial t} &= F_v(t, \alpha_{1,u}, \dots, \alpha_{M_u,u}, \alpha_{1,v}, \dots, \alpha_{M_v,v}, \alpha_{1,w}, \dots, \alpha_{M_w,w}, \alpha_{1,p}, \dots, \alpha_{M_p,p}) \\
\frac{\partial \alpha_{m_w,w}}{\partial t} &= F_w(t, \alpha_{1,u}, \dots, \alpha_{M_u,u}, \alpha_{1,v}, \dots, \alpha_{M_v,v}, \alpha_{1,w}, \dots, \alpha_{M_w,w}, \alpha_{1,p}, \dots, \alpha_{M_p,p}) \\
\frac{\partial \alpha_{m_p,p}}{\partial t} &= F_p(t, \alpha_{1,u}, \dots, \alpha_{M_u,u}, \alpha_{1,v}, \dots, \alpha_{M_v,v}, \alpha_{1,w}, \dots, \alpha_{M_w,w}, \alpha_{1,p}, \dots, \alpha_{M_p,p})
\end{aligned} \tag{11}$$

where, $1 \leq m_u \leq M_u, 1 \leq m_v \leq M_v, 1 \leq m_w \leq M_w, 1 \leq m_p \leq M_p$, 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 (11) are

$$\begin{aligned}
\alpha_{m_u,u}(0, x, y, z) &= ((u(0, x, y, z) - \bar{u}(x, y, z)), \Phi_{m_u,u}) \\
\alpha_{m_v,v}(0, x, y, z) &= ((v(0, x, y, z) - \bar{v}(x, y, z)), \Phi_{m_v,v}) \\
\alpha_{m_w,w}(0, x, y, z) &= ((w(0, x, y, z) - \bar{w}(x, y, z)), \Phi_{m_w,w}) \\
\alpha_{m_p,p}(0, x, y, z) &= ((p(0, x, y, z) - \bar{p}(x, y, z)), \Phi_{m_p,p})
\end{aligned} \tag{12}$$

The errors for the above POD model can be obtained by (details in Luo (2007)):

$$\begin{aligned}
\|u_{full} - u\|_2 &\leq \sqrt{\lambda_{M_u+1}} \\
\|v_{full} - v\|_2 &\leq \sqrt{\lambda_{M_v+1}} \\
\|w_{full} - w\|_2 &\leq \sqrt{\lambda_{M_w+1}} \\
\|p_{full} - p\|_2 &\leq \sqrt{\lambda_{M_p+1}}
\end{aligned} \tag{13}$$

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:

$$-\nabla p_g = f \mathbf{k} \nabla \mathbf{u} \tag{14}$$

Taking the divergence of equation (14), an elliptic equation for geostrophic pressure is obtained

$$-\nabla^2 p_g = \frac{\partial(-fv)}{\partial x} + \frac{\partial(fu)}{\partial y} \tag{15}$$

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 (15) can be obtained from a quadratic finite element representation (figure 1) whilst linear finite element representations for the velocity components. Furthermore since the velocity components are represented by a summation of POD basis functions can be represented by a summation of the two sets of geostrophic basis functions. Therefore these basis functions are calculated by solving the elliptic equations:

$$\begin{aligned}
-\nabla^2 \Phi_{pgu,m} &= \frac{\partial(f\Phi_{m,u})}{\partial y} \\
-\nabla^2 \Phi_{pgv,m} &= \frac{\partial(-f\Phi_{m,v})}{\partial x}
\end{aligned} \tag{16}$$

where, $\Phi_{pgu,m}$ and $\Phi_{pgv,m}$ are the basis functions respectively for velocity components u and v . $m = (1, \dots, M)$ indicates a set of basis functions, M is the total number of basis functions.

The geostrophic pressure can therefore be expressed as:

$$p_g = \bar{p}_g + \sum_{m=1}^M \alpha_{m,u} \Phi_{m,u} + \sum_{m=1}^M \alpha_{m,v} \Phi_{m,v} \quad (17)$$

In addition the average geostrophic pressure is calculated from:

$$-\nabla^2 \bar{p}_g = \frac{\partial(-f\bar{v})}{\partial x} + \frac{\partial(f\bar{u})}{\partial y} \quad (18)$$

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 extended easily 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 to 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 models. The solutions from the original full models are interpolated from their own mesh onto the same reference fixed mesh at each time level, and then stored at the snapshots. The information at the snapshots is used to find the optimal POD basis. This allows the same length of base modes at each time level. The resolution of the referenced mesh and the interpolation errors between the two meshes may affect the accuracy of the POD simulation. This will be exploited and discussed in detail through applications below. The reference mesh can be obtained by superimposing the resolution at each mesh level associated with a goal-based function. To reduce the interpolation error, the 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 :

$$\mathbf{A}^n \alpha^n = s^n \quad (19)$$

where,

$$s^n = \mathbf{B}^n + f \quad (20)$$

where, \mathbf{A}^n and \mathbf{B}^n are the matrices at the time level n , $\alpha^n = (\alpha_1^n, \dots, \alpha_m^n, \dots, \alpha_M^n)$ 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:

$$\mathbf{A}^n = \hat{\mathbf{A}}_0 + \sum_{m=1}^M \alpha_m^{n-1} \hat{\mathbf{A}}_m \quad (21)$$

$$\mathbf{B}^n = \hat{\mathbf{B}}_0 + \sum_{m=1}^M \alpha_m^{n-1} \hat{\mathbf{B}}_m \quad (22)$$

where, the matrices $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are time independent, and derived in the appendix. Equations (21) and (22) can be rewritten as:

$$\mathbf{A}^n = \hat{\mathbf{A}}_0 + \hat{\mathbf{A}}^n \alpha^{n-1} \quad (23)$$

$$\mathbf{B}^n = \hat{\mathbf{B}}_0 + \hat{\mathbf{B}}^n \alpha^{n-1} \quad (24)$$

where, $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_1, \dots, \hat{\mathbf{A}}_m, \dots, \hat{\mathbf{A}}_M)$, $\hat{\mathbf{B}} = (\hat{\mathbf{B}}_1, \dots, \hat{\mathbf{B}}_m, \dots, \hat{\mathbf{B}}_M)$, 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

In this section, the POD model developed here is applied to 2D plus time cases. Error estimate is carried out for the validation of the POD model.

5.1 Case 1: flow past a cylinder

The POD model developed here is first applied to the flow past a cylinder. In the case, the adaptive mesh is adopted in the full model. The reference fix mesh for the POD simulation is shown in Figure ?. The cylinder had a radius of 1 and the computational domain is 29 by 10. The domain has a depth of 1 and one element along the vertical direction. An inlet velocity of

1 is imposed normal to the boundary on the left side, and the centre of the cylinder is placed 5 from the inlet boundary. The Reynolds number is $Re = 100$. No-slip boundary condition is applied to the cylinder and both lateral sides. The spin-up period is 8. The initial condition is set up by running the full model from the 'static' state during the spin-up period. The time step is 0.02 and the mesh adapts every 20 time steps. The maximum and minimum mesh sizes are 0.1 and 0.04. The maximum mesh aspect ratio is 1000.

In this case, 20 snapshots and 10 basis functions are chosen for u , v , w and p , in 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 blowup of the velocity field around the cylinder at the time level $t = 10$. It is indicated that the reduced simulation can provide almost identical details of local flow as does the full simulation. The overall error of the results from the reduced model by the comparison of those from the full model is less than 0.8. The whole simulation by running the reduced model is completed within 9 minutes, whilst 40 minutes by 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 last 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 for 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 full and reduced simulations that two separated jets form downstream of the cylinder, as described in reference (C.Tansley and D.P.Marshall,2001). 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 much less than that (38-39 hrs) required by running the full forward model, that is, the computer cost is significantly reduced by 92%.

[Fig. 3 about here.]

[Fig. 4 about here.]

[Fig. 5 about here.]

5.3 Case 3: Gyre

Another example of applications is wind driven barotropic circulation. The computational domain is taken to be a square box of $1000km$ with a depth of $500m$. 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 $\rho_0 = 1000kgm^{-1}$. 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 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 $3.78E-04$, equivalent to $6hrs$. No-slip boundary conditions are applied to the lateral boundaries. The spin-up period is $0.3024(200days)$. 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 have an overall good concordance with those from the full model.

[Fig. 6 about here.]

To judge the quality of the POD model developed here, the error estimate is undertaken. The percentage of energy represented by the POD bases is listed in table 7. The energy can be captured more than 99% when 60 POD bases are chosen for 81 snapshots, and 97% of energy for the choice of 30 POD bases and 41 snapshots. Above 91% of energy is captured if half the number of POD bases is chosen. In general, the more POD bases and snapshots are chosen, the more energy is represented.

[Table 1 about here.]

The error of numerical results obtained by a different number of POD bases can be calculated by (13). Figure 7 illustrates the eigenvalues and error associated with a corresponding number of POD bases for the velocity components u and v . It is indicated that the first quarter numbers of POD bases have a significant effect 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 a first quarter numbers of POD bases are chosen.

The RMSE and correlation coefficient of results between the full and POD models at different time levels are given in figures 9 and 8. It can be seen that as simulated time goes by, the POD error increases, whilst the correlation decreases. During the first half simulation period, the POD results are consistent with those from the full model. This is supported by the good correlation (mostly larger than 0.8) and small RMSE (mostly less than 1). It is also shown that the increase of the number of snapshots and POD bases can improve the correlation and reduce the RMSE (figure 10).

[Fig. 7 about here.]

[Fig. 8 about here.]

[Fig. 9 about here.]

[Fig. 10 about here.]

6 Conclusion

A POD reduced model has been developed for a finite element adaptive meshes ocean model (here, ICOM). This is the first attempt to apply the POD approach to an adaptive model. To allow the same length of POD modes 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 at the snapshots and stored to find the optimal POD bases.

The POD reduced model is applied to 2D time-dependent ocean cases. The Coriolis effect is considered in the POD model. The 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 .

The 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%, the RMSE of results is less than 1 and 99% of energy can be captured if a suitable number of POD bases is chosen (say, half of total POD bases is chosen). The error of POD results decreases by 70%–80% of its original values if a first quarter numbers of total POD bases are chosen.

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 before running the reduced model. It is proved that the computer cost by running the reduced model can be decreased by 75 – 92% of that required by running the full model.

Further research will address the following issues: (1) goal oriented POD to optimise/weight POD bases; (2) gappy POD which allows the consideration of incomplete data sets; and (3) adjoint (Inverse) POD with adaptivity both in mesh and in controls.

7 Appendix

The second-order Crank-Nicolson time stepping algorithm is used, and the sub-matrices are:

$$\begin{aligned}
\hat{A}_{0,i,j} &= \text{frac}12\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. \\
&\quad \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} &= \text{frac}12\Delta t \sum_{k=1}^{\mathcal{N}} \frac{\partial p_k}{\partial x}, \\
\hat{A}_{0,\mathcal{M}+i,j} &= \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. \\
&\quad \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}, \\
\hat{A}_{0,\mathcal{M}+i,j} &= 0 \\
\hat{A}_{0,\mathcal{M}+i,\mathcal{M}+j} &= 0 \\
\hat{A}_{0,\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. \\
&\quad \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,\mathcal{M}+i,3\mathcal{M}+j} &= \frac{1}{2} \Delta t \sum_{k=1}^{\mathcal{N}} \frac{\partial p_k}{\partial z}, \\
\hat{A}_{0,3\mathcal{M}+i,j} &= \sum_{k=1}^{\mathcal{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{25}$$

$$\begin{aligned}
\hat{B}_{0,i,j} &= \sum_{k=1}^N u^{n-1} - \frac{1}{2} \Delta t \sum_{k=1}^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} + \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}^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}^N \frac{\partial p_k}{\partial x}, \\
\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}^N v^{n-1} - \frac{1}{2} \Delta t \sum_{k=1}^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} + \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}^N \frac{\partial p_k}{\partial y}, \\
\hat{B}_{0,\mathcal{M}+i,j} &= \frac{\partial \bar{w}_k}{\partial x} \Phi_{u,j,k} \\
\hat{B}_{0,\mathcal{M}+i,\mathcal{M}+j} &= \frac{\partial \bar{w}_k}{\partial y} \Phi_{v,j,k} \\
\hat{B}_{0,\mathcal{M}+i,2\mathcal{M}+j} &= \sum_{k=1}^N w^{n-1} - \frac{1}{2} \Delta t \sum_{k=1}^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} + \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} \right], \\
\hat{B}_{0,\mathcal{M}+i,3\mathcal{M}+j} &= \frac{1}{2} \Delta t \sum_{k=1}^N \frac{\partial p_k}{\partial z},
\end{aligned} \tag{26}$$

$$\begin{aligned}
\hat{A}_{i,j} &= \frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{N}} \left[\Phi_{u,l,k} \frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k} \frac{\partial \Phi_{u,j,k}}{\partial y} \Phi_{w,l,k} \frac{\partial \Phi_{u,j,k}}{\partial z} \right] \\
\hat{A}_{\mathcal{M}+i,\mathcal{M}+j} &= \frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{N}} \left[\Phi_{u,l,k} \frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k} \frac{\partial \Phi_{u,j,k}}{\partial y} \Phi_{w,l,k} \frac{\partial \Phi_{u,j,k}}{\partial z} \right] \\
\hat{A}_{2\mathcal{M}+i,2\mathcal{M}+j} &= \frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{N}} \left[\Phi_{u,l,k} \frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k} \frac{\partial \Phi_{u,j,k}}{\partial y} \Phi_{w,l,k} \frac{\partial \Phi_{u,j,k}}{\partial z} \right]
\end{aligned} \tag{27}$$

$$\begin{aligned}
\hat{B}_{i,j} &= -\frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{N}} \left[\Phi_{u,l,k} \frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k} \frac{\partial \Phi_{u,j,k}}{\partial y} \Phi_{w,l,k} \frac{\partial \Phi_{u,j,k}}{\partial z} \right] \\
\hat{B}_{\mathcal{M}+i,\mathcal{M}+j} &= -\frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{N}} \left[\Phi_{u,l,k} \frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k} \frac{\partial \Phi_{u,j,k}}{\partial y} \Phi_{w,l,k} \frac{\partial \Phi_{u,j,k}}{\partial z} \right] \\
\hat{B}_{2\mathcal{M}+i,2\mathcal{M}+j} &= -\frac{1}{2}\Delta t \sum_{l=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{N}} \left[\Phi_{u,l,k} \frac{\partial \Phi_{u,j,k}}{\partial x} + \Phi_{v,l,k} \frac{\partial \Phi_{u,j,k}}{\partial y} \Phi_{w,l,k} \frac{\partial \Phi_{u,j,k}}{\partial z} \right]
\end{aligned} \tag{28}$$

where, \mathcal{M} represents the number of basis functions, \mathcal{N} is the number of nodes.

Acknowledgments

The authors would like to thank ?? for many helpful discussions. This work was carried out under funding from ??.

List of Figures

1	Linear and quadratic mixed finite element. ●: u, v, w, p_{ng} ; ○: p_g	20
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).	21
3	Case1: 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 $t = 8$; middle panel: at the time level $t = 10$; bottom panel: at the time level $t = 12$).	22
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)	23
5	Case2–cylinder on a β plane ($\beta = 7.5$): 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 $t = 2$).	24
6	Case3: comparison of velocity field between the full and reduced models (left panel: the full model; right panel: the reduced model; top panel: at the time level $t = 200days$; middle panel: at the time level $t = 300days$; bottom panel: at the time level $t = 400days$).	25
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).	26
8	Case3: Correlation at time levels (left panel: 41 snapshots; right panel: 81 snapshots).	27
9	Case3: RMS at time levels (left panel: 41 snapshots; right panel: 81 snapshots).	28
10	Case3: Comparison of correlation and RMS with different snapshot numbers (left panel: Correlation; right panel: RMS).	29

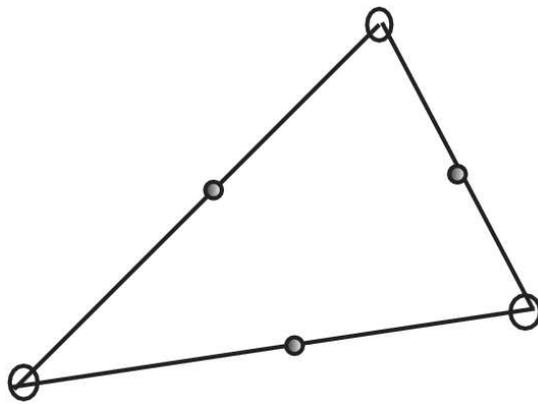


Fig. 1. Linear and quadratic mixed finite element. ●: u, v, w, p_{ng} ; ○: p_g

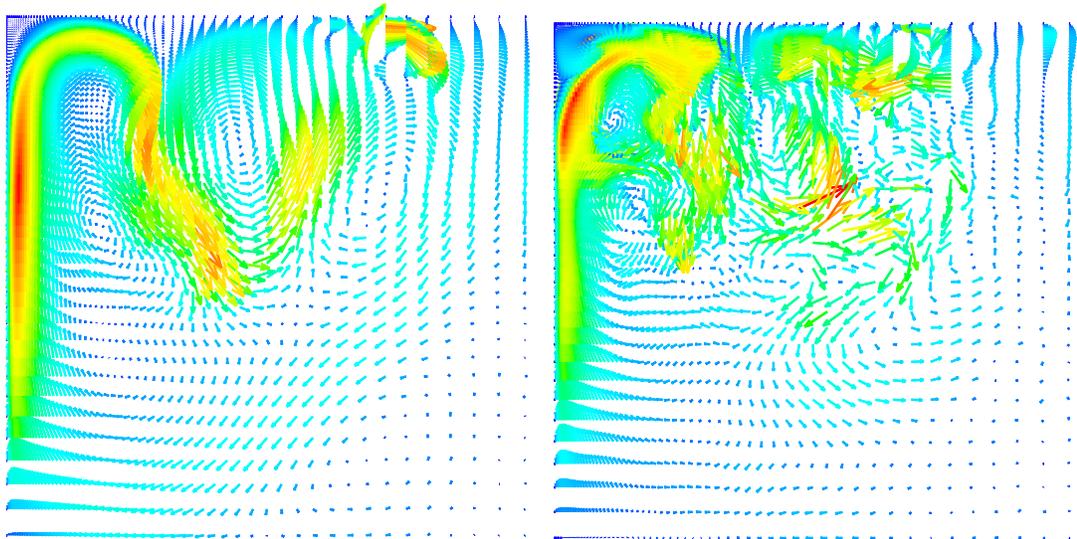


Fig. 2. Effect of geostrophic balance on results from POD simulation (left panel: with the new geostrophic pressure method; right panel: without the new geostrophic pressure method).

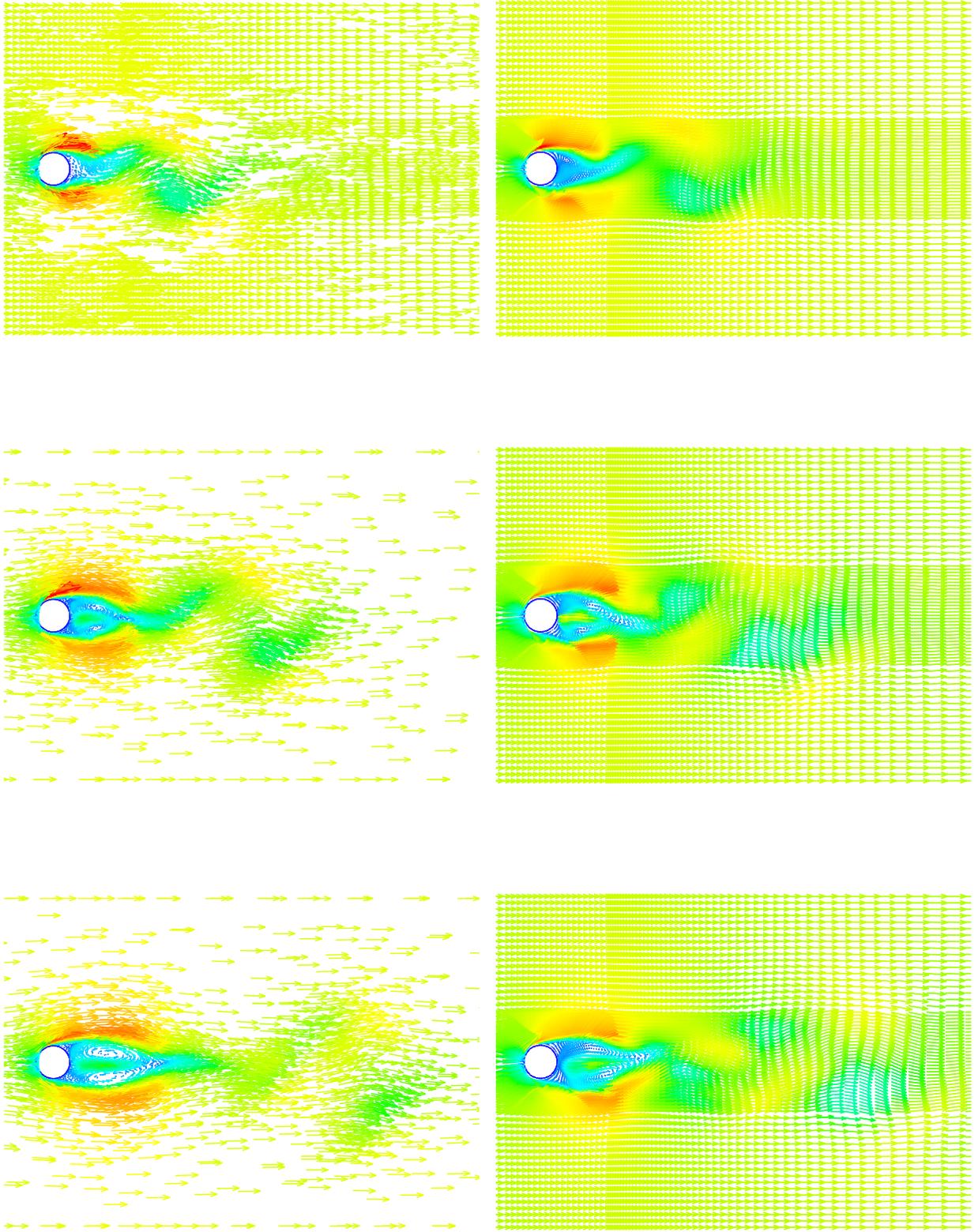


Fig. 3. Case1: comparison of velocity field between the full and reduced models (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t = 8$; middle panel: at the time level $t = 10$; bottom panel: at the time level $t = 12$).

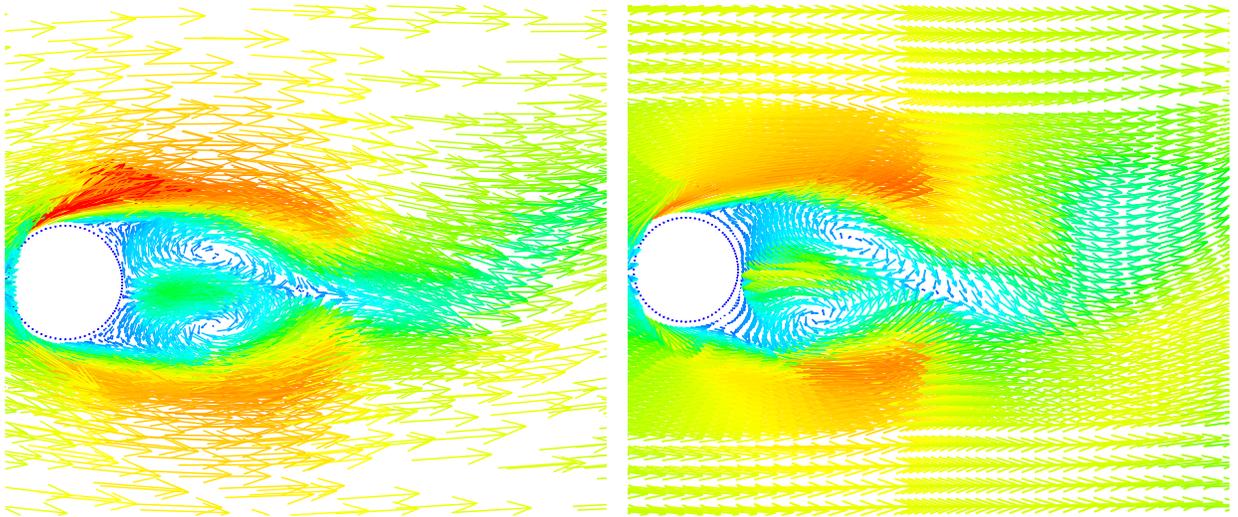


Fig. 4. Case1: Blowup of the velocity field in figure 3 around the cylinder at the time level $t = 10$. (left panel: the full model; right panel: the reduced model)

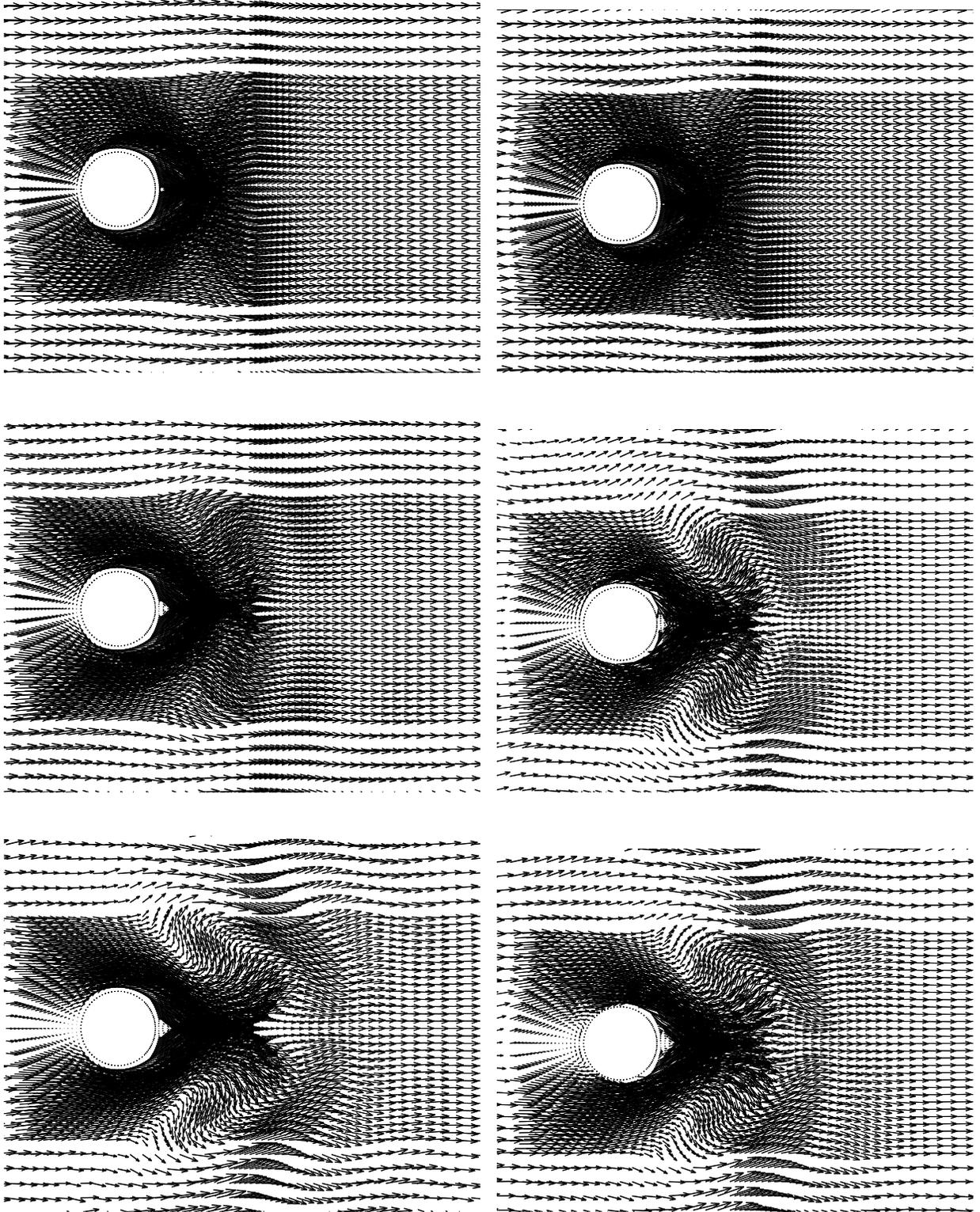


Fig. 5. Case2–cylinder on a β plane ($\beta = 7.5$): comparison of velocity field between the full and reduced models (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t = 0.4$; middle panel: at the time level $t = 0.8$; bottom panel: at the time level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

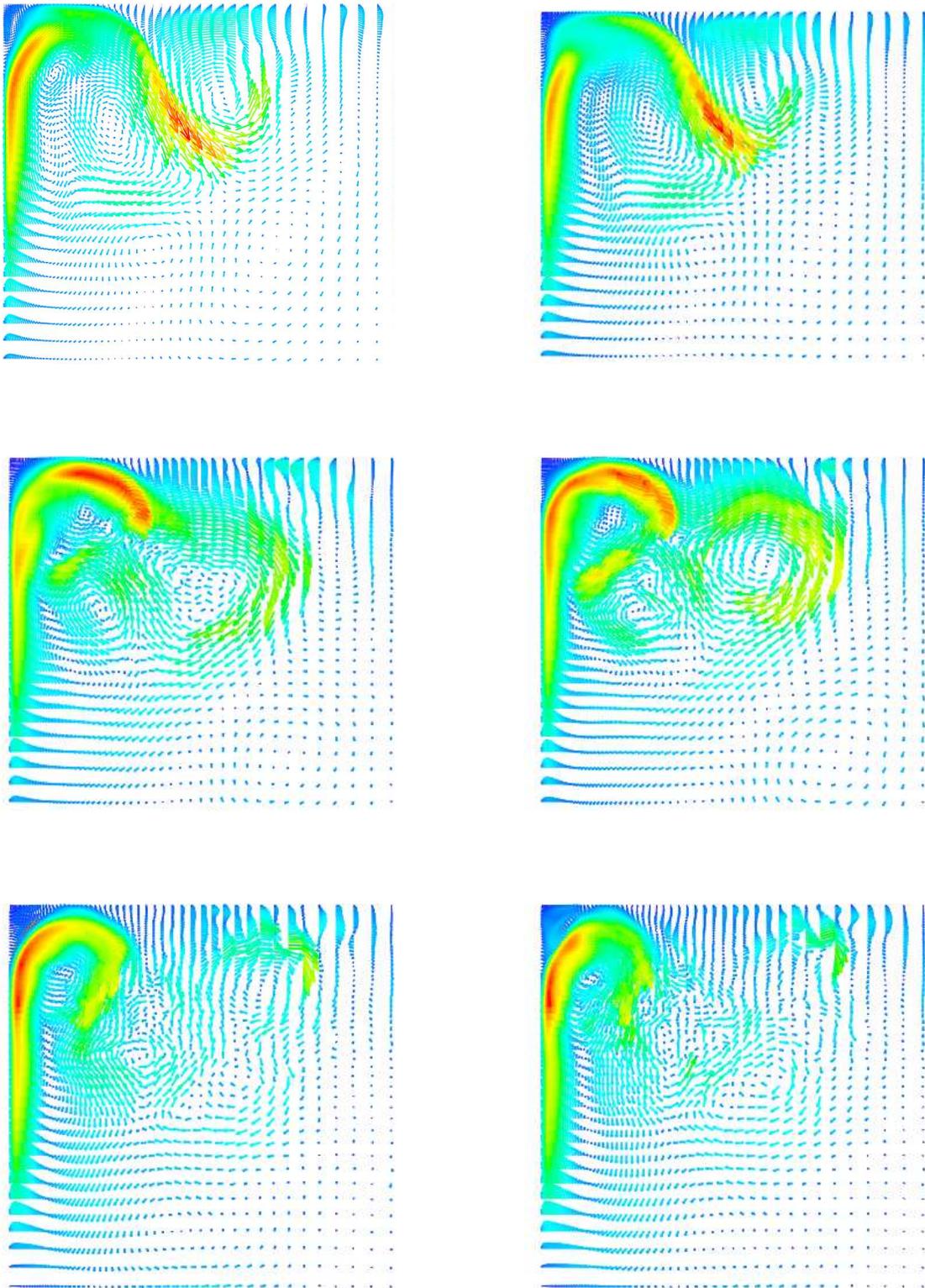


Fig. 6. Case3: comparison of velocity field between the full and reduced models (left panel: the full model; right panel: the reduced model; top panel: at the time level $t = 200days$; middle panel: at the time level $t = 300days$; bottom panel: at the time level $t = 400days$).

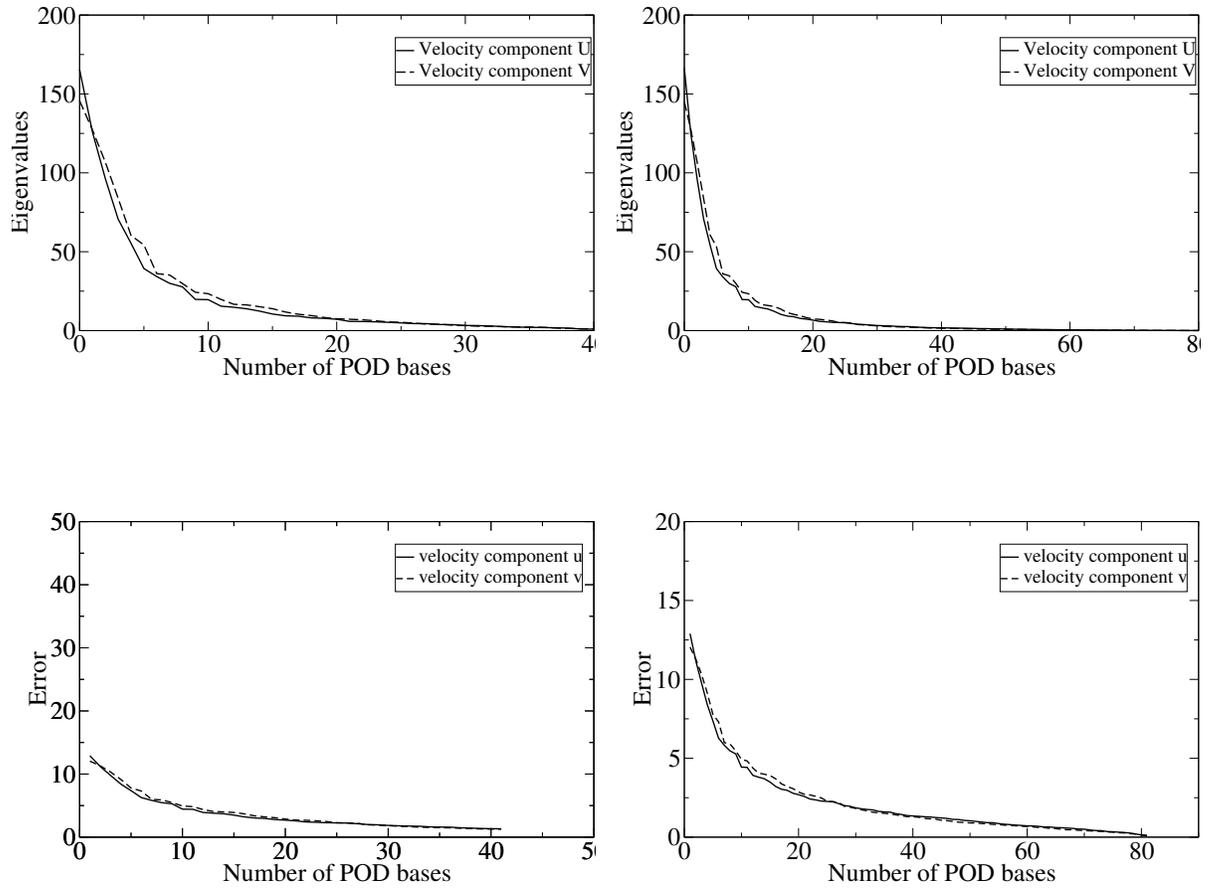


Fig. 7. Case3: Eigenvalues and errors for velocity components u and v (top panel: eigenvalues; bottom panel: error left panel: 41 snapshots; right panel: 81 snapshots).

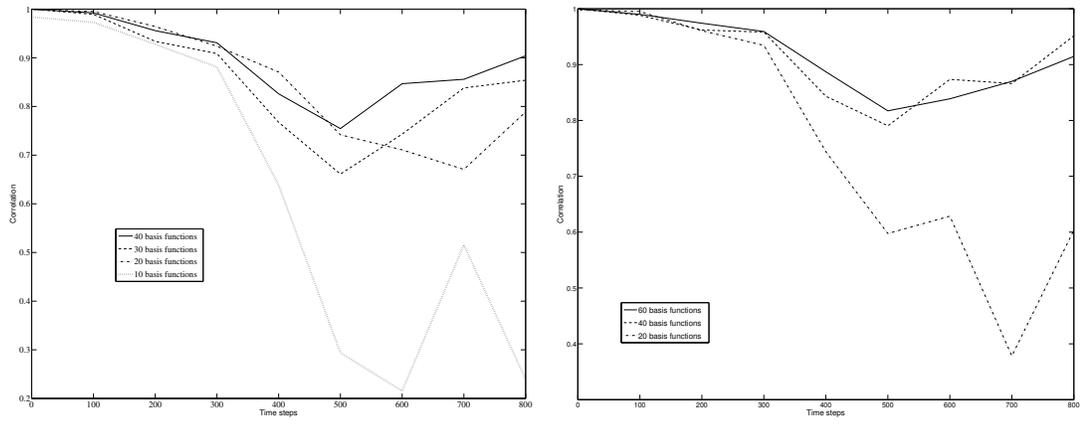


Fig. 8. Case3: Correlation at time levels (left panel: 41 snapshots; right panel: 81 snapshots).

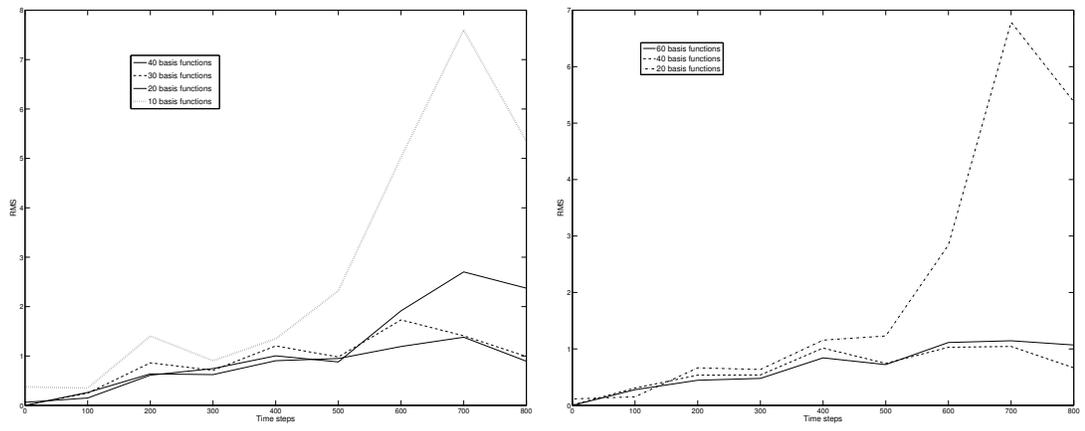


Fig. 9. Case3: RMS at time levels (left panel: 41 snapshots; right panel: 81 snapshots).

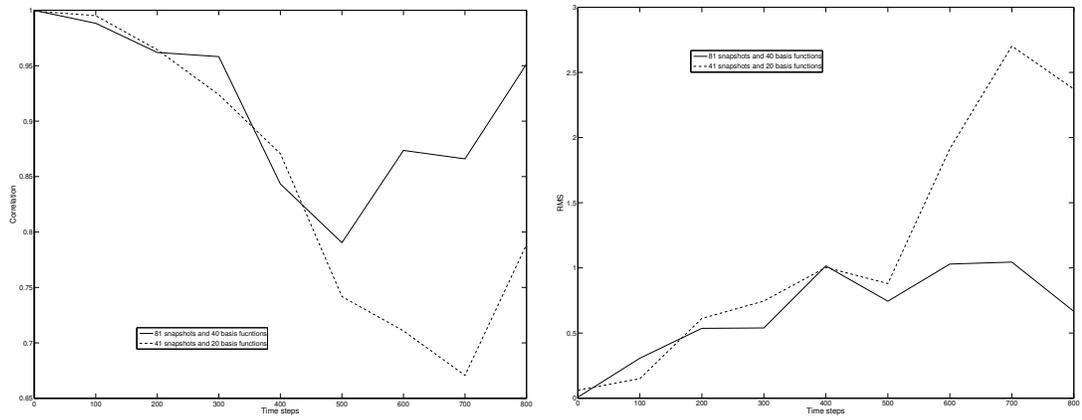


Fig. 10. Case3: Comparison of correlation and RMS with different snapshot numbers (left panel: Correlation; right panel: RMS).

List of Tables

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

Table 2: Energy captured by the POD bases for velocity components, u, v and pressure p .