Journal of Computational Physics 263 (2014) 1–18

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Non-linear model reduction for the Navier–Stokes equations using residual DEIM method

D. Xiao [a,b], F. Fang [a], A.G. Buchan [a], C.C. Pain [a], I.M. Navon [c,*], J. Du [d], G. Hu [b]

[a] Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London SW7 2BP, UK [1]
[b] State Key Laboratory of Geological Processes and Mineral Resources, China University of Geosciences, Wuhan 430074, China
[c] Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, USA
[d] Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China

ARTICLE INFO

Article history:
Received 6 February 2013
Received in revised form 1 December 2013
Accepted 7 January 2014
Available online 14 January 2014

Keywords:
Non-linear model reduction
Empirical interpolation method
Petrov–Galerkin
Proper orthogonal decomposition
Navier–Stokes

ABSTRACT

This article presents a new reduced order model based upon proper orthogonal decomposition (POD) for solving the Navier–Stokes equations. The novelty of the method lies in its treatment of the equation's non-linear operator, for which a new method is proposed that provides accurate simulations within an efficient framework. The method itself is a hybrid of two existing approaches, namely the quadratic expansion method and the Discrete Empirical Interpolation Method (DEIM), that have already been developed to treat non-linear operators within reduced order models. The method proposed applies the quadratic expansion to provide a first approximation of the non-linear operator, and DEIM is then used as a corrector to improve its representation. In addition to the treatment of the non-linear operator the POD model is stabilized using a Petrov–Galerkin method. This adds artificial dissipation to the solution of the reduced order model which is necessary to avoid spurious oscillations and unstable solutions.

A demonstration of the capabilities of this new approach is provided by solving the incompressible Navier–Stokes equations for simulating a flow past a cylinder and gyre problems. Comparisons are made with other treatments of non-linear operators, and these show the new method to provide significant improvements in the solution's accuracy.

1. Introduction

Reduced order models (ROMs) have become important to many fields of physics as they offer the potential to simulate dynamical systems with substantially increased computation efficiency in comparison to traditional techniques. Among the model reduction techniques, the proper orthogonal decomposition (POD) method has proven to be an efficient means of deriving the reduced basis for high-dimensional non-linear flow systems. The POD method and variants of it have been successfully applied to a number of research fields. In signal analysis and pattern recognition it is known as the Karhunen–Loève method [1], in statistics it is referred to as principal component analysis (PCA) [2], and in geophysical fluid dynamics and meteorology it is termed empirical orthogonal functions (EOF) [3,4]. The POD method has since been applied to ocean models in Cao et al. [5], Vermeulen and Heemink [6] and also shallow water equations, this includes the work of Daescu and Navon [7], Chen et al. [8,9], Altaf et al. [10], Du et al. [11], as well as Fang et al. [12].

* Corresponding author.
E-mail address: inavon@fsu.edu (I.M. Navon).
[1] http://amcg.ese.imperial.ac.uk.

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2014.01.011

In this paper we develop a reduced order model for the incompressible Navier–Stokes equations using the POD approach. The full Navier–Stokes equations are first discretized via a finite element Bubnov Galerkin discretization method [13] and the POD model is generated through the method of snapshots. In this approach, solutions of the full model are recorded (as a sequence of snapshots), and from this data appropriate basis functions are formed that optimally represent the problem. This method is quite standard and has been applied successfully throughout the literature. However, due to the high non-linearities of the 3-D Bubnov–Galerkin Navier–Stokes equation, the computational complexity of the reduced model still depends on dimension of the full Navier–Stokes discretization [14]. To mitigate this problem, one approach is to apply the discrete empirical interpolation method (DEIM) to address the reduction of the non-linear components and reduce the computational complexity by implementing it with the POD/DEIM method. DEIM is a discrete variant of the empirical interpolation method (EIM) [15] proposed by Barrault et al. for constructing approximation of a non-affine parameterized function, which was proposed in the context of reduced-basis model order reduction. DEIM methods have been demonstrated to be able to obtain factors of 10–100 speed up in CPU time over the original non-reduced model. The economy in CPU time is proportional to the dimension of the reduced order model (see for instance Stefanescu and Navon, 2013 [16]) and therefore to the number of mesh points. The application was suggested and analyzed by Chaturantabut and Sorensen [17–19] for application to POD in the framework of Discrete Empirical Interpolation Method (DEIM). Other important contributions to the Empirical Interpolation Method (EIM) include that by Barrault et al. [15] and Patera related to another model reduction approach namely the reduced basis approach [20–26].

Regarding the use of hyper-reduced order models i.e. DEIM like approaches, they presented a strategy for choosing the optimal set of sampling points at the discrete level. The algorithm consists of selecting the sampling components that minimize the distance between the recovered reduced basis coefficients and the optimal coefficients (which are obtained by projecting the snapshots onto the reduced order subspace). The main advantage of their algorithm is that only values at the nodes of the finite element mesh are required for the gappy reconstruction [27–29], but these sampling components can be guaranteed to be optimal. This results in a strategy very convenient for the reconstruction of non-smooth functions, like the right-hand-side of the system of equations arising from the reduced order strategy for the incompressible Navier–Stokes equations with the formulation used herein.

An alternative treatment of the non-linear terms of PDEs is through the quadratic expansion method [11]. This method is suitable for the treatment of discretized quadratic non-linear operators as the method represents them through expansions of precomputed matrices. Critically, as these matrices are precomputed they can easily be transformed into reduced equation sets. However the method's drawback is that its accuracy will decay with higher order non-linear operators which may arise through Petrov–Galerkin projections and/or additional stabilizing terms.

Both the novel quadratic expansion method and the DEIM have been developed in order to maintain the ROM's efficiency. In this article a new method is proposed which is a new hybrid of both schemes that refer to as residual DEIM. It is based on initially applying the quadratic expansion method to the non-linear terms and then applying the DEIM approach to resolve the residual between it and the full model. That is, the DEIM is used to absorb the remaining errors left over from the quadratic expansion approach. This approach means that the method can still exactly represent discrete quadratic non-linearities – unlike DEIM – but can also be used for highly non-linear discrete systems – unlike the quadratic expansion approach. In addition to this a non-linear Petrov–Galerkin discretization [30,12] is used to form the ROM and stabilize the reduced system of equations, which would otherwise become unstable especially for moderate/high Reynolds number flows. Both the stabilization term and the Petrov–Galerkin projection introduce additional high order non-linearities, and this makes the residual DEIM method well suited to dealing with the resulting discrete systems of equations, see, for example, Baiges [31] for similar approaches. This paper demonstrates the improved accuracy of residual DEIM over the quadratic expansion method when simulating problems involving high order non-linear operators.

The structure of the paper is as follows. Section 2 presents the governing equations, followed by the description of the finite element Bubnov–Galerkin discretization of the Navier–Stokes equations. Section 3 presents the derivation of the POD model reduction and re-formulation of the Navier–Stokes equations using the method of snapshots. The section concludes with the stabilization of the POD model reduction by the introduction of an adequately chosen dissipation term. Section 4 focuses on the non-linear operator treatment of the Navier–Stokes equations and describes the methods of DEIM and quadratic expansion. This section then presents the mixed residual DEIM formulation. This is based on a DEIM representation of a residual term that is left over from first applying a quadratic representation of the non-linear operator. Section 5 illustrates the methodology derived via two numerical examples. This is based on two test problems where the flow past a cylinder and flow within a gyre are resolved. Finally in Section 6 conclusions are presented and the novelty of the present manuscript is duly summarized and illuminated.

2. Governing equations

This article considers the non-hydrostatic Navier–Stokes equations describing the conservation of mass and momentum of a fluid,

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

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

In these equations the terms $\mathbf{u} \equiv (u_x, u_y, u_z)^T$ denote the velocity vector, p the perturbation pressure ($p := p/\rho_0$, ρ_0 is the constant reference density) and f the Coriolis inertial force. The stress tensor $\boldsymbol{\tau}$ included in the diffusion term represents the viscous forces, and this is defined in terms of a deformation rate tensor $\mathbf{S}$ which is given as,

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

In this expression μ denotes the kinematic viscosity and it is assumed that there is no summation over repeated indices. The horizontal (μ_{xx}, μ_{yy}) and vertical (μ_{zz}) kinematic viscosities are assumed to take constant values and define the off diagonal components of $\boldsymbol{\tau}$ in Eq. (3) by $\mu_{ij} = (\mu_{ii}\mu_{jj})^{1/2}$. For barotropic flow, 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 the constant buoyancy force exactly, and so both terms are neglected at this stage. The momentum equation can be expressed more fully as,

$$A_t\frac{\partial \mathbf{u}}{\partial t} + A_x(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x} + A_y(\mathbf{u})\frac{\partial \mathbf{u}}{\partial y} + A_z(\mathbf{u})\frac{\partial \mathbf{u}}{\partial z} + f\mathbf{k}\times\mathbf{u} + \nabla p - \nabla\cdot\boldsymbol{\tau} = 0, \tag{4}$$

where the time term A_t and streaming operators A_x, A_y and A_z denote diagonal matrices that are given by,

$$A_t = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

and

$$A_x = \begin{pmatrix} u_x & 0 & 0 \\ 0 & u_x & 0 \\ 0 & 0 & u_x \end{pmatrix}, \qquad A_y = \begin{pmatrix} u_y & 0 & 0 \\ 0 & u_y & 0 \\ 0 & 0 & u_y \end{pmatrix}, \qquad A_z = \begin{pmatrix} u_z & 0 & 0 \\ 0 & u_z & 0 \\ 0 & 0 & u_z \end{pmatrix}, \tag{6}$$

respectively.

In this article a finite element Bubnov–Galerkin discretization of the Navier–Stokes equations [13] is employed. In this formulation the velocity components and pressure terms of the solution are represented by the expansions,

$$u_x = \sum_j^{F_u} N_j u_{xj}, \qquad u_y = \sum_j^{F_u} N_j u_{yj}, \qquad u_z = \sum_j^{F_u} N_j u_{zj}, \tag{7}$$

and

$$p = \sum_j^{F_p} M_j p_j, \tag{8}$$

respectively, where N_j and M_j denote the finite element basis functions. To solve for the coefficients $u_{.j}$ and p_j the discretized equations are formed by weighting equations (1) and (2) by M_i and N_i, respectively, and integrating over space,

$$\int_v M_i(\nabla\cdot\mathbf{u})\,dv = 0, \tag{9}$$

$$\int_v N_i\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + f\mathbf{k}\times\mathbf{u}\right)dv = -\nabla p + \nabla\cdot\boldsymbol{\tau}. \tag{10}$$

When the approximations (7) and (8) are inserted into these equations the following systems are formed,

$$\begin{aligned} &C^t u = 0, \\ &N\frac{\partial \mathbf{u}}{\partial t} + A(\mathbf{u})\mathbf{u} + K\mathbf{u} + Cp = s. \end{aligned} \tag{11}$$

In these equations the matrix C denotes the pressure gradient matrix, N is the mass matrix involving the finite element basis functions N_i, $A(u)$ is the solution dependent discretized streaming operator, K is the matrix related to the rest of the linear terms of velocity, and s is the vector accounting for the forces acting upon the solution. In the momentum equation the time term is treated using the θ-method to yield,

$$N\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + A\left(\mathbf{u}^n\right)\mathbf{u}^{n+\theta} + K\mathbf{u}^{n+\theta} + Cp^{n+1} = 0, \tag{12}$$

where $\theta \in [0, 1]$ and the term $\mathbf{u}^{n+\theta}$ is given by,

Table 1
The table lists the variables and their definition as used in the present article.

Variable	Definition
S	Total number of arbitrary snapshot set.
S_u	Total number of snapshot for velocity components.
S_p	Total number of snapshot for pressure components.
F	Total number of nodes on arbitrary finite element discretization.
F_u	Total number of nodes on finite element discretization of the velocity.
F_p	Total number of nodes on finite element discretization of the pressure.
P	Total function in an arbitrary POD basis set.
P_u	Total number of functions in the velocity POD basis set.
P_p	Total number of functions in the pressure POD basis set.
Φ	General POD basis functions.
Φ_p	Standard POD basis functions.
Φ_d	DEIM POD basis functions.

$$\mathbf{u}^{n+\theta} = \theta\mathbf{u}^{n+1} + (1-\theta)\mathbf{u}^{n}. \tag{13}$$

The full system of equations can now be grouped together to form a single combined linear system for each time step,

$$\begin{bmatrix} B & C \\ C^T & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} B' & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n} \\ p^{n} \end{bmatrix} + \begin{bmatrix} s \\ 0 \end{bmatrix}, \tag{14}$$

where B and B' are matrices of similar form but differ through the choice of θ. In this system the matrix B is non-linear as it depends on the solution $\mathbf{u}$. On the RHS, the vector $[s, 0]^T$ contains the discretized sources and the terms within the matrix system account for the solution from the previous time step.

Alternative but identical expressions of the above system of equations can be written in order to help develop the methods derived in the following sections. One expression is derived by condensing the expression into the single system of equations given by,

$$P(y)y^{n+1} = Q(y)y^{n} + s, \tag{15}$$

where y^n denotes the full solution vector at time step n, i.e. the concatenation of all velocity and pressure components. Another representation of this equation is to re-write system (15) with separated linear (L superscript) and non-linear (N superscript) terms,

$$\left(P^L + P^N\right)y^{n+1} = \left(Q^L + Q^N\right)y^{n} + s. \tag{16}$$

The following theory will be built upon one of these identical expressions of the discretized Navier–Stokes equations.

3. POD method for the Navier–Stokes equations

In the following subsections the POD method is described together with its application to the modeling of the Navier–Stokes equation and its stabilization. In the theory that follows in this and the next section the notation becomes quite involved and so Table 1 has been included in order to make clear the variable definition.

3.1. The POD model

In the POD formulation a new set of basis functions are constructed from a collection of snapshots that are taken at a number of time instances of the full model solution. That is, the model described in Eq. (14) is solved and snapshots of the solution are taken as it evolves through time. In the formulation presented here snapshots of each component of the velocity vector (u_x, u_y, u_z) and pressure p are recorded individually. Each snapshot is a vector of size F_u or F_p (depending on whether it is of a velocity component of pressure term) and holds the values of the respective solution component at the nodes of the finite element mesh. For each direction or pressure component, these snapshots are collated together over all time instances to form four separate matrices $\mathcal{U}^x$, $\mathcal{U}^y$, $\mathcal{U}^z$ and $\mathcal{U}^p$ (where the superscripts denote direction or pressure). From here on each snapshot matrix will be treated separately but in an identical manner, and so the superscripts are omitted and the details are given for a general snapshot matrix $\mathcal{U}$.

The dimensions of $\mathcal{U}$ is $F \times S$, where F denotes the general number of nodes on the finite element mesh and S the total number of snapshots (this will be of value S_u and S_p for the velocity and pressure, respectively). Once the full set of snapshots has been collated, it is then customary to remove from each snapshot the mean value of all snapshots. That is, a modified snapshot matrix $\overline{\mathcal{U}}$ is generated by,

$$\overline{\mathcal{U}}_{k,j} = \mathcal{U}_{k,j} - \overline{\Phi}_j, \quad j \in \{1, 2, \ldots, S\}, \tag{17}$$

where the vector $\overline{\Phi}$ (of size F) holds the average value of all snapshot on each node i:

$$\overline{\Phi}_i = \frac{1}{N_k}\sum_{j=1}^{N_k} \mathcal{U}_{j,i}, \quad i \in \{1, 2, \ldots, F\}. \tag{18}$$

A reduced-order basis set of functions $\{\Phi_u\}$ are now obtained by means of the Proper Orthogonal Decomposition method. This involves performing a Singular Value Decomposition (SVD) of the snapshot matrix $\overline{\mathcal{U}}$ given by the form,

$$\overline{\mathcal{U}} = U \Sigma V^T. \tag{19}$$

The terms U and V are unitary matrices of dimension $F \times F$ and $S \times S$, respectively, and Σ is a diagonal matrix of size $F \times S$. The nonzero values of Σ are the singular values of $\overline{\mathcal{U}}$, and these are assumed to be listed in increasing or decreasing order of their magnitude. It can be shown [17] that the POD functions can be defined as the column vectors of the matrix U,

$$\Phi_j = U_{:,j}, \quad \text{for } j \in \{1, 2, \ldots, S\}, \tag{20}$$

and the optimal basis set of size P are the functions corresponding to the largest P singular values (i.e. the first P columns of U). These functions are optimal in the sense that no other rank P set of basis functions can be closer to the snapshot matrix $\overline{\mathcal{U}}$ in the Frobenius norm. That is, if one used only the first P dominant singular values in Eq. (20) (and so the first P vectors in U), the resulting matrix is the closest possible (in the relevant norm) to the matrix $\overline{\mathcal{U}}$. Another relevant property is that due to U being unitary, the POD vectors are orthonormal.

To efficiently construct the POD vectors defined by U one of two approaches may be taken. Depending on the dimensions of $\overline{\mathcal{U}}$ a reduced symmetric linear system can be formed by the pre or post multiplication of $\overline{\mathcal{U}}$ by its transpose. If the number of finite element nodes is smaller than the number of snapshots ($F \ll S$), then post multiplying by $\overline{\mathcal{U}}^T$ results in an $F \times F$ system with the property,

$$\overline{\mathcal{U}}\overline{\mathcal{U}}^T = U \Lambda^2 U^T. \tag{21}$$

This enables one to perform an eigenvalue decomposition on the system (21) to obtain U directly. Alternatively, if the number of snapshots is smaller than the number of finite element functions ($S \ll F$), the pre multiplication of $\overline{\mathcal{U}}$ by $\overline{\mathcal{U}}^T$ results in,

$$\overline{\mathcal{U}}^T\overline{\mathcal{U}} = V^T \Lambda^2 V. \tag{22}$$

In this case one can perform the eigenvalue decomposition of system (22) to obtain the matrix V and singular values Λ. Once these are available, the vectors of U can be formed by substituting into system (19).

As mentioned previously, only a small number of P POD functions are used in the reduced order model. Although these POD functions provide an optimal representation of the snapshot matrix, some information is inevitably lost. This loss of information can be quantified by the following ratio, which is usually termed energy, of the squared singular values,

$$I = \frac{\sum_{i=1}^{P} \Lambda_{i,i}^2}{\sum_{i=1}^{S} \Lambda_{i,i}^2}. \tag{23}$$

The value of I will tend to 1 as P is increased to the value S, and so this value can be used to provide an appropriate truncation point of the POD expansion set. Having set the size P, the P POD functions can now be used to form a basis that represents the snapshot data set. That is, a vector u of size F can be represented by the expansion,

$$u = \overline{\Phi} + \sum_{j}^{P} \alpha_j \Phi_j, \tag{24}$$

where α_j denote the expansion coefficients.

3.2. Forming the POD formulation of the Navier–Stokes equations

To form the reduced order system of the Navier–Stokes equations, the velocity and pressure components are expanded over their respective POD basis functions. Their finite element solution variables (Eq. (14)) are re-written in the form of Eq. (24) to give,

$$u_x^n = \overline{\Phi}^x + \sum_{j}^{P_u} \alpha_j^{x,n} \Phi_j^x, \qquad u_y^n = \overline{\Phi}^y + \sum_{j}^{P_u} \alpha_j^{y,n} \Phi_j^y, \qquad u_z^n = \overline{\Phi}^z + \sum_{j}^{P_u} \alpha_j^{z,n} \Phi_j^z, \tag{25}$$

for the velocities and,

$$p^n = \overline{\Phi}^p + \sum_{j}^{P_p} \alpha_j^{p,n} \Phi_j^p, \tag{26}$$

for the pressure. The POD expansion sizes of the velocity and pressure terms are denoted by P_u and P_p respectively, and the α terms denote the expansion coefficients. These expansions can be represented in the following matrix vector form,

$$u^n = \overline{\Phi}^x + \Phi^x \alpha^{x,n}, \qquad v^n = \overline{\Phi}^y + \Phi^y \alpha^{y,n}, \qquad w^n = \overline{\Phi}^z + \Phi^z \alpha^{z,n}, \qquad p^n = \overline{\Phi}^p + \Phi^p \alpha^{p,n}, \tag{27}$$

where Φ^x, Φ^y and Φ^z denote matrices of size $F_u \times P_u$, Φ^p is a matrix of size $F_p \times P_p$, $\overline{\Phi}^x$, $\overline{\Phi}^y$ and $\overline{\Phi}^z$ are vectors of size F_u and $\overline{\Phi}^p$ is a vector of size F_p.

If the solution variables are also represented by a single vector, as in Eq. (15), then the reduced order representation can also read as,

$$y^n = \begin{bmatrix} u_x \\ u_y \\ u_z \\ p \end{bmatrix}^n = \overline{\Phi}^y + \Phi^y \alpha^{y,n} = \begin{bmatrix} \overline{\Phi}^x \\ \overline{\Phi}^y \\ \overline{\Phi}^z \\ \overline{\Phi}^p \end{bmatrix} + \begin{bmatrix} \Phi^x & 0 & 0 & 0 \\ 0 & \Phi^y & 0 & 0 \\ 0 & 0 & \Phi^z & 0 \\ 0 & 0 & 0 & \Phi^p \end{bmatrix} \begin{bmatrix} \alpha^{x,n} \\ \alpha^{y,n} \\ \alpha^{z,n} \\ \alpha^{p,n} \end{bmatrix}. \tag{28}$$

This is an identical expression to those of Eq. (27) since the terms $\overline{\Phi}^y$ and Φ^y are formed from the combination of all the POD matrices and vectors. The coefficients in the discretized system (14) (or (15)) are then replaced by their POD representation of Eq. (27) (or (28)), and the resulting system is pre-multiplied by the transpose of the POD matrices in order to form the reduced system. For now we work on the equation in the form of (14) and the reduced system reads as,

$$\begin{bmatrix} B^{POD} & C^{POD} \\ (C^{POD})^T & 0 \end{bmatrix} \begin{bmatrix} \alpha^{u,n+1} \\ \alpha^{p,n+1} \end{bmatrix} = \begin{bmatrix} B'^{POD} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha^{u,n} \\ \alpha^{p,t} \end{bmatrix} - \begin{bmatrix} \bar{s}^u \\ \bar{s}^p \end{bmatrix} + \begin{bmatrix} s^{POD} \\ 0 \end{bmatrix}. \tag{29}$$

It can be seen that the system retains the same structure of the original full system in Eq. (14). The reduced matrix B^{POD} is of size $3P_u \times 3P_u$ and can be written as $B^{POD} = (\Phi^u)^T B \Phi^u$ (with a similar expression of B'^{POD}) where,

$$\Phi^u = \begin{bmatrix} \Phi^x & 0 & 0 \\ 0 & \Phi^y & 0 \\ 0 & 0 & \Phi^z \end{bmatrix}. \tag{30}$$

Similarly the matrix C^{POD} is a reduced system of size $3P_u \times P_p$ and this is given by $C^{POD} = (\Phi^u)^T C \Phi^p$. The reduced source terms is a vector of size P_u and is formulated as $s^{POD} = (\Phi^u)^T s$. The additional source terms in Eq. (29) result from contribution of the average snapshot vectors, and these are given by,

$$\begin{bmatrix} \bar{s}^u \\ \bar{s}^p \end{bmatrix} = \begin{bmatrix} (\Phi^u)^T B \overline{\Phi}^u + (\Phi^u)^T C \overline{\Phi}^p \\ (\Phi^p)^T C \overline{\Phi}^u \end{bmatrix}. \tag{31}$$

3.3. Stabilization of the POD model

The POD model described in Eq. (29) will often require an additional stabilization term to ensure its solutions remain bounded by it correctly modeling the proper amount of physical dissipation. Without this most POD models of high-Reynolds shear flow are liable to explode. Typically the ROMs are augmented with eddy viscosities [32] and this approach is adopted here. The details of this stabilization are described in full in [30], and so only the main concepts are presented in this article. The stabilization terms are added to the model through an additional diffusion operator which has the same effect as adding an additional viscous term. It is included into the LHS matrix of Eq. (29) by adding to it the block diagonal diffusion matrix,

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_x & 0 & 0 & 0 \\ 0 & \mathbf{D}_y & 0 & 0 \\ 0 & 0 & \mathbf{D}_z & 0 \\ 0 & 0 & 0 & \mathbf{D}_p \end{pmatrix}. \tag{32}$$

This matrix has size $(3P_u + P_p) \times (3P_u + P_p)$ and is composed of the three sub-matrices,

$$\mathbf{D}_{xij} = \int_V \nabla \Phi_i^x \mu_x \nabla \Phi_j^x \, dV, \tag{33}$$

$$\mathbf{D}_{yij} = \int_V \nabla \Phi_i^y \mu_y \nabla \Phi_j^y \, dV, \tag{34}$$

$$\mathbf{D}_{zij} = \int_V \nabla \Phi_i^z \mu_z \nabla \Phi_j^z \, dV, \tag{35}$$

that are of size $P_u \times P_u$ and the matrix,

$$\mathbf{D}_{p_{ij}} = \int_V \nabla \Phi_i^p \mu_p \nabla \Phi_j^p \, dV, \tag{36}$$

which is of size $P_p \times P_p$. Each matrix places diffusion in their respective velocity or pressure terms, and the amount of diffusion is governed by the coefficients μ. The diffusion terms ensure the ROMs have more stable solutions. However it is important to stress that its inclusion introduces strong non-quadratic non-linearities to the formulation. This in turn must be resolved efficiently and accurately by the reduced order schemes. In this work it was not necessary to stabilize the pressure and so the diffusion coefficient relating to it has been set to zero.

4. Efficient treatments of the non-linear operators

In the following sections a review of the current approaches to resolving the non-linear operators are presented. This is then followed by the details of the proposed residual DEIM method.

4.1. DEIM treatment of the non-linear operator

This section presents the application of DEIM which will be used in the following sections to resolve the non-linear terms of the Navier–Stokes equations within an efficient reduced order model. For now the method is described in the general sense for an arbitrary differential equation that is expressed in terms of its linear (L) and non-linear (N) components. When discretized through a finite element representation the following model is formed,

$$\frac{dy}{dt}(t) = Ly(t) + N\big(y(t)\big), \quad A \in \mathbb{R}^{F\times F}. \tag{37}$$

In this equation the new solution variables $y(t) = [y_1(t), y_2(t), \ldots, y_F(t)] \in \mathbb{R}^F$ define the solution's values over the F nodes of the finite element mesh. As the term N is a non-linear function it requires $y(t)$ to be evaluated component wise at each time instance t, i.e. $N = [N(y_1(t)), \ldots, N(y_F(t))]$. It is this re-evaluation that makes the reduced order modeling inefficient if the standard POD approach (as described in the previous section) is applied. This is because if the non-linear term is represented through a POD model, i.e. it is pre and post multiplied by Φ_p^T and Φ_p respectively (the subscript p denotes standard POD functions), the resulting reduced space formulation will read as,

$$\tilde{N}(\tilde{y}) = \underbrace{\Phi_p^T}_{P\times F} \underbrace{N\big(\Phi_p \tilde{y}(t)\big)}_{F\times 1} = \underbrace{\Phi_p^T}_{P\times F} \underbrace{f(t)}_{F\times 1}. \tag{38}$$

This expression shows that even in the reduced space, the model requires an operation with complexity of order F at each time step. The model is therefore no longer efficient to compute, i.e. it is the same order of the high-fidelity full model, and so an alternative approach must be applied.

The DEIM approach is one such method used to treat the non-linear terms of PDEs within a reduced order framework. The approach is to use a separate POD model to construct a basis of the space spanned by the non-linear components of the equation. That is, a snapshot matrix $\mathcal{U}_d$ of the non-linear terms is constructed by,

$$\mathcal{U}_d = \big\{N(y_1), N(y_2), \ldots, N(y_n)\big\}, \tag{39}$$

from which a POD model is built using the approach described in the previous section. Using Φ_d to denote this POD basis set, which is analogous to that in Eq. (20), the term $f(t)$ in Eq. (38) can be represented by,

$$f(t) = \Phi_d c(t). \tag{40}$$

The dimension of Φ_d is of $F \times D$, where D denotes the size of the reduced representation of the non-linear terms ($D \ll F$) and $c(t)$ is a coefficient vector of size D that has yet to be determined. If one now inserts (40) into (38) the following results,

$$\tilde{N}(\tilde{y}) = \underbrace{\Phi_p^T \Phi_d}_{P\times D} \underbrace{c(t)}_{D\times 1}, \tag{41}$$

for which the matrix in this expression is time independent. The matrix can therefore be precomputed, which in turn means that once $c(t)$ is known, the non-linear expression can be computed with order of complexity D.

The vector $c(t)$ is constructed by solving a reduced form of the over determined system (40), and this is given by,

$$\underbrace{\Phi_{d\rho}}_{\mathbb{R}^{D\times D}} \underbrace{c(t)}_{\mathbb{R}^D} = \underbrace{f_\rho(t)}_{\mathbb{R}^D}, \tag{42}$$

where the $c(t)$ vector remains the same as that in the original system. The new system in Eq. (42) is of size $D \times D$, and this is formed by extracting D rows from the original $F \times D$ system. The selection of which rows (or interpolation points)

to use is discussed in [17,33], but once determined the selected row indices are indicated by the indexing vector $\hat{\rho}$, which is of size D (i.e. if the kth selected point corresponds to row i then $\hat{\rho}_k = i$). Using this vector the elements of the system (42) are given as,

$$\{\Phi_{d\rho}\}_{ij} = \{\Phi_d\}_{\hat{\rho}_i \hat{\rho}_j}, \quad \text{and} \quad f_\rho(t)_i = f(t)_{\hat{\rho}_i}, \tag{43}$$

respectively. Provided that the matrix is invertible, or that a pseudo-inverse is used instead, the system (42) can be solved to obtain $c(t)$,

$$c(t) = \Phi_{d\rho}^{-1} f_\rho(t). \tag{44}$$

One can now generate the matrix $P = [e_{\rho_1}, \ldots, e_{\rho_m}] \in \mathbb{R}^{F \times D}$, which is formed from the vectors e_i which have the value 1 in their component ρ_i and zero elsewhere, that is $e_{\rho_i} = [0, 0, \ldots, \underbrace{1}_{\rho_i}, \ldots, 0, 0]^T \in \mathbb{R}^F$. This matrix can be used to represent the reduced components in Eq. (44) through their original matrices and vectors by,

$$\Phi_{d\rho} = P^T \Phi_d, \tag{45}$$

and

$$f_\rho(t) = P^T f(t), \tag{46}$$

respectively. Expressions (44), (45) and (46) can now be combined and used within (40) to give the following formulation,

$$f(t) = \Phi_d \left(P^T \Phi_d\right)^{-1} P^T f(t). \tag{47}$$

The final form of the reduced model of the non-linear component can now be formed by inspecting Eq. (38) and noting that,

$$f_\rho(t) = P^T f(t) = P^T F\left(\Phi_p \tilde{y}(t)\right) = F\left(P^T \Phi_p \tilde{y}(t)\right). \tag{48}$$

Replacing this expression inside (48) gives,

$$f(t) = \Phi_d \left(P^T \Phi_d\right)^{-1} F\left(P^T \Phi_p \tilde{y}(t)\right), \tag{49}$$

which is then substituted within (38) to give the final form of the non-linear reduced order model,

$$\tilde{N}(\tilde{y}) \approx \underbrace{\Phi_p^T \Phi_d \left(P^T \Phi_d\right)^{-1}}_{\text{precomputed } P \times D} \underbrace{F\left(P^T \Phi_p \tilde{y}(t)\right)}_{D \times 1}. \tag{50}$$

As indicated, the computationally expensive matrix can be precomputed due to its time independence. Therefore, at each time instance of the reduced order model, only a matrix vector multiplication involving a system of size $P \times D$ is required.

4.2. Quadratic expansion of the non-linear operator

An alternative approach for efficiently treating the non-linear terms within a reduced order model is through the quadratic expansion method proposed in [11]. The approach is reviewed here by considering the matrix operator B in Eq. (14), for which the non-linear components arise from the streaming operator in the Navier–Stokes equation. The matrix is re-written by the following summation involving the $P_u + 1$ sub-matrices,

$$B = \overline{B} + \sum_{i=1}^{P_u} \hat{B}_i. \tag{51}$$

In this expression the matrix $\overline{B}$ is of size $3F_u \times 3F_u$, and this is dependent on the average velocity components $\overline{u}$, i.e. $\overline{B} = B(\overline{u})$. The matrices $\hat{B}_i$ in the summation are also of size $3F_u \times 3F_u$ and these are decomposed further into the following form,

$$\hat{B}_i = \begin{bmatrix} \alpha_i^1 \hat{B}_i^x & 0 & 0 \\ 0 & \alpha_i^2 \hat{B}_i^y & 0 \\ 0 & 0 & \alpha_i^3 \hat{B}_i^z \end{bmatrix}. \tag{52}$$

Here the sub-matrices $\hat{B}_i^j$ are of size $F_u \times F_u$ and these are dependent on the ith POD basis function that is associated with direction component j. In this expression the expansion multiplies the matrices by their respective POD coefficients α_i^j, however the sub-matrices themselves are fixed and so can be precomputed. The precomputing can be accomplished by

considering a perturbation to the average vector $\bar{u}$. The perturbations are defined in the POD space, and so three vectors of size P_u are created that relate to all POD basis functions for each of the three velocity components. Their values are all set to zero except for a small perturbation ϵ in one element of a vector. For example, in the vector relating to the direction x, the ith perturbation is given by,

$$\epsilon_i^x = \{0,\ldots,0,\underbrace{\epsilon}_{i},0\ldots,0\},\qquad \epsilon_i^y = \{0,\ldots,0,\ldots,0\},\qquad \epsilon_i^z = \{0,\ldots,0,\ldots,0\}. \tag{53}$$

These perturbed POD coefficients provide a perturbed velocity solution given as,

$$\tilde{u} = \begin{bmatrix} u^x \\ u^y \\ u^z \end{bmatrix} = \bar{u} + \begin{bmatrix} \epsilon\Phi^x \\ 0 \\ 0 \end{bmatrix}, \tag{54}$$

which can be used in expression (51) to give the expression of $\hat{B}_i^j$,

$$\hat{B}_i^j = \frac{1}{\epsilon}\big(B(\tilde{u}) - B(\bar{u})\big). \tag{55}$$

The same approach can be applied to obtain all matrices in the summation (51), and once computed they can be represented in the reduced space by their projection with the POD functions. That is,

$$\hat{B}_i^{POD} = \big(\Phi^u\big)^T \hat{B}_i \Phi^u, \tag{56}$$

which are in turn used to define the reduced B matrix,

$$B^{POD} = \bar{B}^{POD} + \sum_i^{P_u} \hat{B}_i^{POD}, \tag{57}$$

where $\bar{B}^{POD} = (\Phi^u)^T \bar{B} \Phi^u$ is the projection of $\bar{B}$ onto the reduced space.

4.3. The residual DEIM method

A mixed DEIM-quadratic expansion formulation of the non-linear reduced order operator is now derived. It is based on a DEIM representation of a residual term that is left over from first applying a quadratic representation of the non-linear operator. The method is derived from Eq. (16) which represents the full model discretized into its linear and non-linear components. The system is now rewritten as,

$$\big(P^L + P_f^N + P_q^N - P_q^N\big)y^{n+1} = \big(Q^L + Q_f^N + Q_q^N - Q_q^N\big)y^n + s, \tag{58}$$

where the subscripts f and q denote the full and quadratic operators, respectively. This is then re-arranged into the form of

$$\big(P^L + P_q^N\big)y^{n+1} = \big(Q^L + Q_q^N\big)y^n + s + \big\{\big(-Q_f^N + Q_q^N\big)y^n + \big(-P_f^N + P_q^N\big)y^{n+1}\big\}. \tag{59}$$

The system on the left hand side now only contains a linear and a non-linear component that is represented by the quadratic expansion. This and the first two terms on the RHS (which also contain only quadratic representations of the non-linear terms) are therefore re-cast into an efficient reduced order model using the method described in Sections 3.2 and 4.2. These reduced systems (denoted with a $\tilde{.}$) are given by,

$$\tilde{P}^L = \Phi_p^T P^L \Phi_p,\qquad \tilde{Q}^L = \Phi_p^T Q^L \Phi_p \quad\text{and}\quad \tilde{b} = \Phi_p^T b \tag{60}$$

for the linear terms and

$$\tilde{P}^N = \Phi_p^T P^N \Phi_p \quad\text{and}\quad \tilde{Q}^L = \Phi_p^T Q^L \Phi_p, \tag{61}$$

for the non-linear quadratic terms (note that the terms P^N and Q^N have representations as given in Eq. (51)).

The remaining term on the RHS of (59) (closed within the brackets) is the residual that is formed from applying the quadratic approximation of the non-linear operator. These operators are highly non-linear, and so in this work they are represented by the DEIM method. That is, the non-linear operator relating to Eq. (37) in the DEIM formulation is expressed as,

$$N\big(y(t)\big) = \big(-Q_f^N + Q_q^N\big)y^n + \big(-P_f^N + P_q^N\big)y^n, \tag{62}$$

which in turn has the reduced order formulation,

$$\tilde{N}(\tilde{y}) = \Phi_p^T \Phi_d \big(P^T \Phi_d\big)^{-1} P^T F\big(\Phi_p \tilde{y}(t)\big). \tag{63}$$

The new reduced order model, which has been named the residual DEIM model, now reads as,

$$\big(\tilde{P}^L + \tilde{P}^N\big)\tilde{y}^{n+1} = \big(\tilde{Q}^L + \tilde{Q}^L\big)\tilde{y}^n + \tilde{b} + \tilde{N}(\tilde{y}). \tag{64}$$

5. Numerical examples

A demonstration of the use of the reduced order modeling scheme is presented in this section. This is based on solving the incompressible Navier–Stokes equations for two test problems where the flow past a cylinder and flow within a gyre are resolved. To obtain the full solutions, both problems were solved using the fluidity model [13]. This applied linear Discontinuous Galerkin (DG) finite elements to the velocity and quadratic continuous finite elements to the pressure terms. In both test cases unstructured triangular meshes were used with sufficient resolution to ensure an accurate solution was obtained. From these full model simulations the snapshots of both the solution variables and the non-linear terms were taken. Using this snapshot data the reduced order models were then formed and used to re-solve the problems.

In this demonstration a comparison between the quadratic treatment of the non-linear terms and the residual DEIM approach has been made. A comparison with DEIM alone has not been included as the method was found to be unstable with the problems presented. This is due to DEIM being formed explicitly, however for incompressibility the constraints need to be treated implicitly. In addition to comparing solution profiles the analysis compares the solution errors as well as correlation coefficients. The measured error is given by the root mean square error (RMSE) which is calculated for each time step n by,

$$RMSE^n = \sqrt{\frac{\sum_{i=1}^{F}(\psi_i^n - \psi_{o,i}^n)^2}{F}}. \tag{65}$$

In this expression ψ_i^n and $\psi_{o,i}^n$ denote the POD (mapped onto the full mesh) and full model solution at the node i, respectively, and F represents the number of nodes on the full mesh. The correlation coefficient is computed for each time step, and is defined for given expected values μ_{ψ^n} and $\mu_{\psi_o^n}$ and standard deviations σ_{ψ^n} and $\sigma_{\psi_o^n}$,

$$corr\left(\psi^n, \psi_o^n\right)^n = \frac{cov(\psi^n, \psi_o^n)}{\sigma_{\psi^n}\sigma_{\psi_o^n}} = \frac{E(\psi^n - \sigma_{\psi^n})(\psi_o^n - \sigma_{\psi_o^n})}{\sigma_{\psi^n}\sigma_{\psi_o^n}}. \tag{66}$$

5.1. Case 1: Flow past a cylinder

In the first numerical example a 2 dimensional flow past a cylinder is simulated. The problem domain is 50 units in length and 10 units in width, and it possesses a cylinder of radius 3 units positioned over the point $(5, 5)$. The dynamics of the fluid flow is driven by an in-flowing liquid, and this enters the domain through the left boundary. The fluid is allowed to flow past the cylinder and out the domain through the right boundary. No slip and zero outward flow conditions are applied to the upper and lower edges of the problem whilst Dirichlet boundary conditions are applied to the cylinder's wall. The properties of the fluid are such that the Reynolds number for this problem is calculated to be $Re = 3200$.

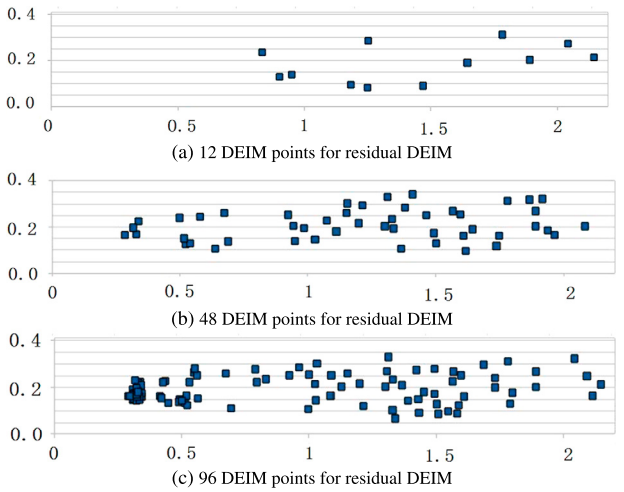
The problem was simulated for a period of 10 time units, and for all models a time step size of $\Delta t = 0.01$ units was used. From the full model simulation, with a mesh of 3213 nodes, 375 snapshots were obtained at equal time intervals for each of the u, v and p solution variables. Similarly, 375 snapshots of the corresponding non-linear residual profiles, as described in Eq. (62), were also taken. Fig. 1 presents the distribution of the interpolation points when using DEIM to represent the residual terms. These points can be thought as providing an indication as to where the quadratic expansion method provides the least accurate reconstruction of the non-linear operators. The stabilizing diffusion term discussed in Section 3.3 was also used in this example, and this added an equivalent viscosity of up to 6.6^{-4}.

Figs. 2–4 present the simulated flow patterns at time instances 3.52 and 10.0. They compare the full solution against reduced order model using both the quadratic expansion and the residual DEIM methods. In each of the figures the number of POD functions used in the simulation increases from 12 to 48 and then 96 functions. In the residual DEIM calculations the number of interpolation points was set to the same number of POD functions used (Table 2). From these flow patterns it is shown that both the quadratic expansion and residual DEIM methods are capable of capturing the solution's main structural details. It is also shown that the residual DEIM performs very well using as few as 12 POD basis functions. In addition, the magnitude of the residual DEIM profiles appears to be in closer agreement to the full model solutions. This is highlighted in the graphs presented in Fig. 5 which show the solution velocities at 3 points in the domain. The results highlight how the residual DEIM method improves the quadratic expansion method by suppressing the over and under shoots that form in its solution.

The graphs in Figs. 6 and 7 show the two reduced order method's errors and correlation coefficients. These show a noticeable improvement in accuracy is gained when using the residual DEIM method, whereby the errors are reduced by approximately 80% in comparison to quadratic expansion. The correlation graphs show the quadratic expansion's coefficient to vary about the values 0.6–0.8 whereas for residual DEIM the values remain around 0.9–0.98. This again illustrates the improved accuracy gained by using this new approach. The errors between the two reduced models and the full solutions are presented in Fig. 8. These show the residual DEIM method to be more accurate than the quadratic expansion method.

Fig. 9 presents the CPU times required to compute a single time step with varying mesh size. It shows the cost of the ROM models to remain static with increased resolution, and that significant efficiencies are gained using meshes with over 3000 nodes. For the largest mesh the CPU costs were reduced by 98%.

Fig. 10 shows the CPU times to compute a single time step with varying POD basis in reduced order model.

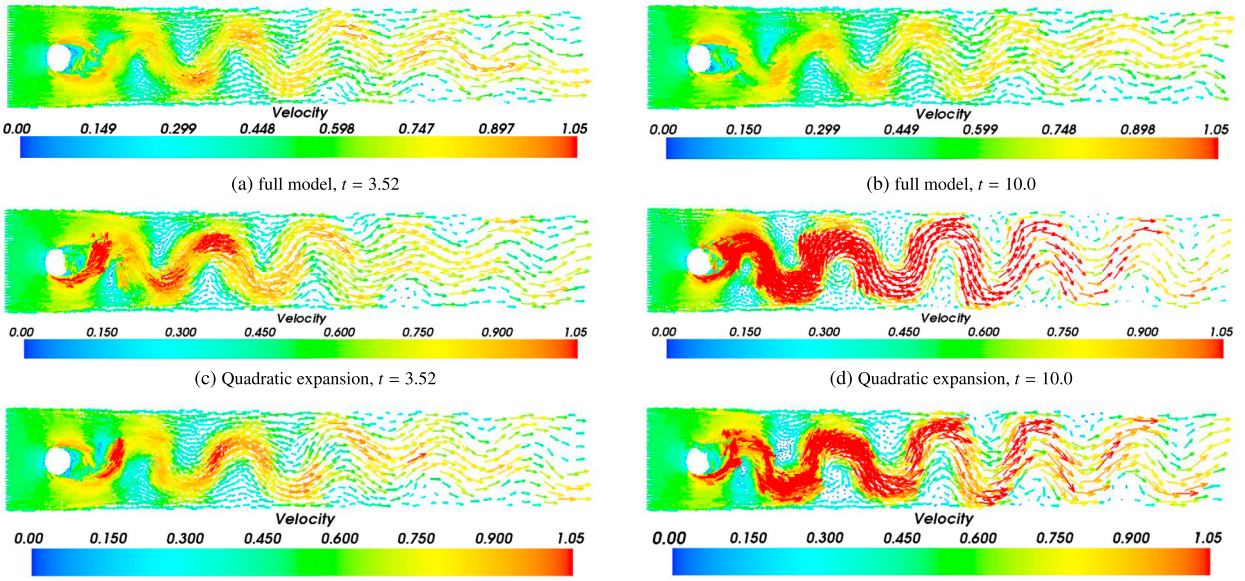


(a) 12 DEIM points for residual DEIM

(b) 48 DEIM points for residual DEIM

(c) 96 DEIM points for residual DEIM

Fig. 1. DEIM point distributions in the x–y plane (horizontal–vertical axis respectively) for the flow past the cylinder problem ($Re = 3200$).



(a) full model, t = 3.52

(b) full model, t = 10.0

(c) Quadratic expansion, t = 3.52

(d) Quadratic expansion, t = 10.0

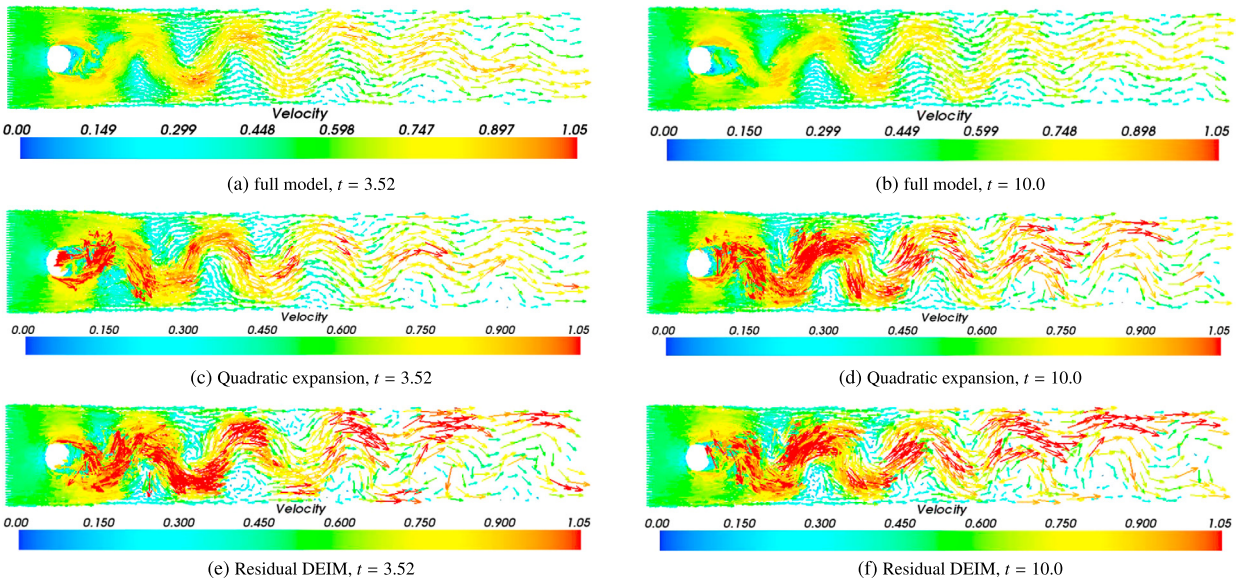
(e) Residual DEIM, t = 3.52

(f) Residual DEIM, t = 10.0

Fig. 2. Velocities of the flow past a cylinder problem at time instances 3.52 and 10.0 with 12 POD basis functions.

Table 2
CPU (unit: s) required for running the full model and 2 ROMS for a time step.

Nodes	Full model	Quadratic POD 12 POD basis	Residual DEIM 12 points	Quadratic POD 6 POD basis	Residual DEIM 6 points
1243	0.188	0.044	0.044	0.008	0.008
1842	0.240	0.044	0.044	0.008	0.008
2346	0.316	0.044	0.044	0.008	0.008
3213	0.440	0.044	0.044	0.008	0.008
7266	1.048	0.048	0.048	0.008	0.008
8117	1.180	0.048	0.048	0.008	0.008
10 195	1.548	0.048	0.048	0.008	0.008

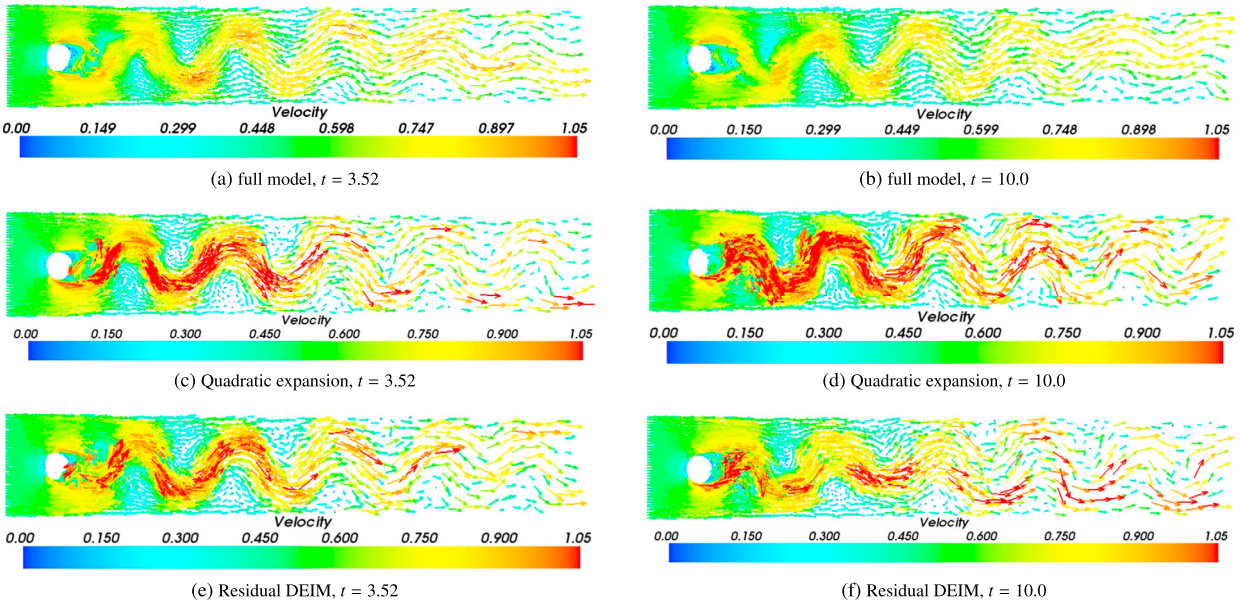


(a) full model, $t = 3.52$ (b) full model, $t = 10.0$

(c) Quadratic expansion, $t = 3.52$ (d) Quadratic expansion, $t = 10.0$

(e) Residual DEIM, $t = 3.52$ (f) Residual DEIM, $t = 10.0$

Fig. 3. Velocities of the flow past a cylinder problem at time instances 3.52 and 10.0 with 48 POD basis functions.



(a) full model, $t = 3.52$ (b) full model, $t = 10.0$

(c) Quadratic expansion, $t = 3.52$ (d) Quadratic expansion, $t = 10.0$

(e) Residual DEIM, $t = 3.52$ (f) Residual DEIM, $t = 10.0$

Fig. 4. Velocities of the flow past a cylinder problem at time instances 3.52 and 10.0 with 96 POD basis functions.

5.2. Case 2: The gyre problem

The second numerical example involves the simulation of a gyre for which a circulating fluid moves across a domain that is 1000×1000 km across and 500 m in depth. The solution's free surface is driven by a wind with a force strength given by the expression,

$$\tau_y = \tau_0 \cos(\pi y/L) \quad \text{and} \quad \tau_x = 0.0, \tag{67}$$

where L is the problem's length scale given by $L = 1000$ km. The terms τ_x and τ_y are the wind stresses on the free surface that act along the x and y directions, respectively. In this example the maximum zonal wind stress was set to $\tau_0 = 0.1\ \mathrm{N\,m^{-1}}$ in the latitude (y) direction. The Coriolis terms are taken into account with the beta-plane approximation ($f = \beta y$) where $\beta = 1.8 \times 10^{-11}$ and the reference density of the fluid set to $\rho_0 = 1000\ \mathrm{kg\,m^{-1}}$. With this setup the Reynolds number of the problem was calculated to be $Re = 250$.

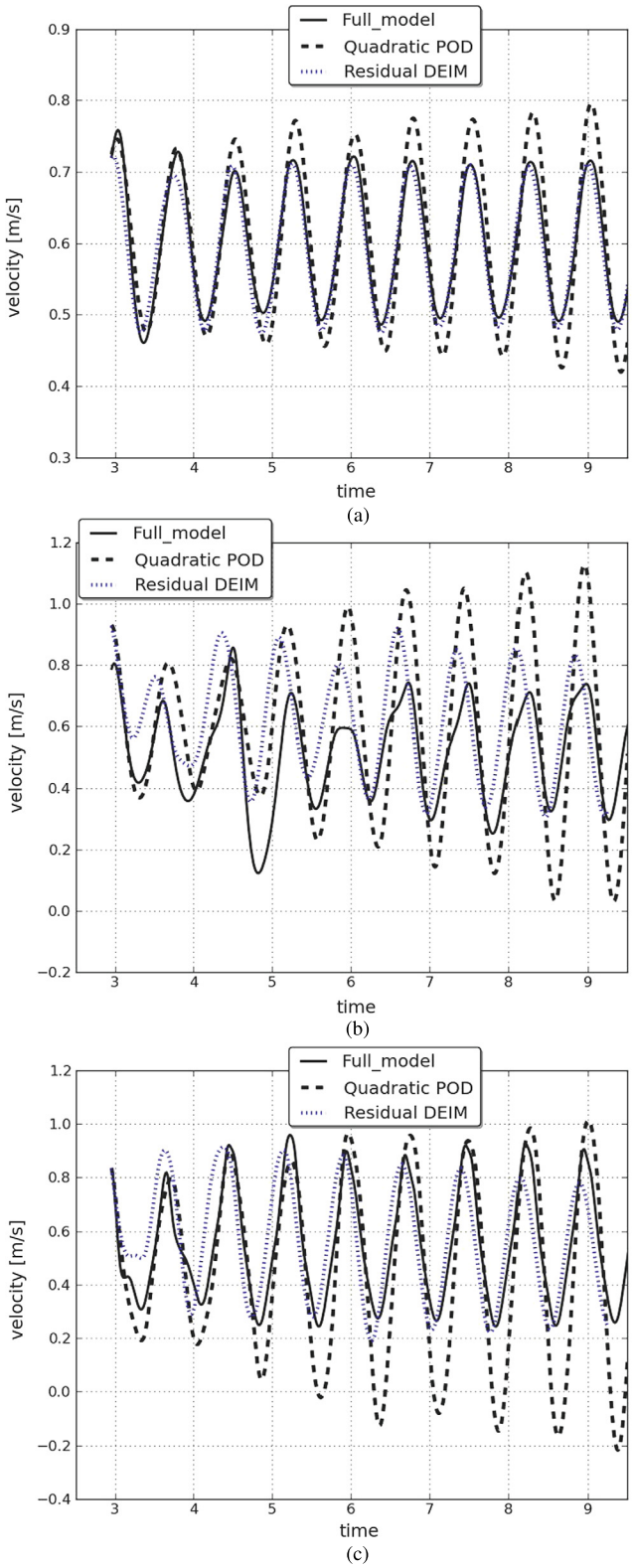


Fig. 5. Velocities predicted by the full model, the quadratic model and the residual DEIM model at positions (a) $(0.5, 0.3)$, (b) $(1.158, 0.315)$ and (c) $(0.574, 0.107)$. These results were obtained using a reduced order model with 48 POD functions and, in the case of residual DEIM, 48 interpolation points.

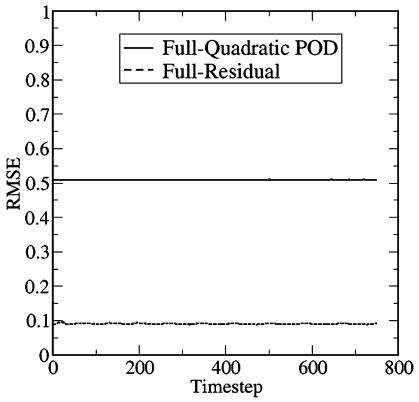


Fig. 6. RMSE errors calculated for the quadratic expansion and residual DEIM methods.

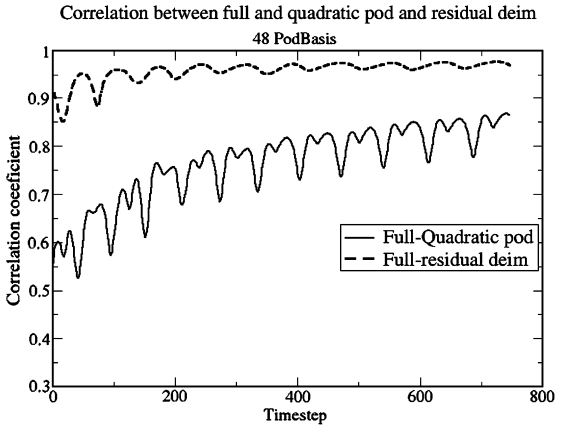


Fig. 7. Correlation coefficient calculated for the quadratic expansion and residual DEIM methods.

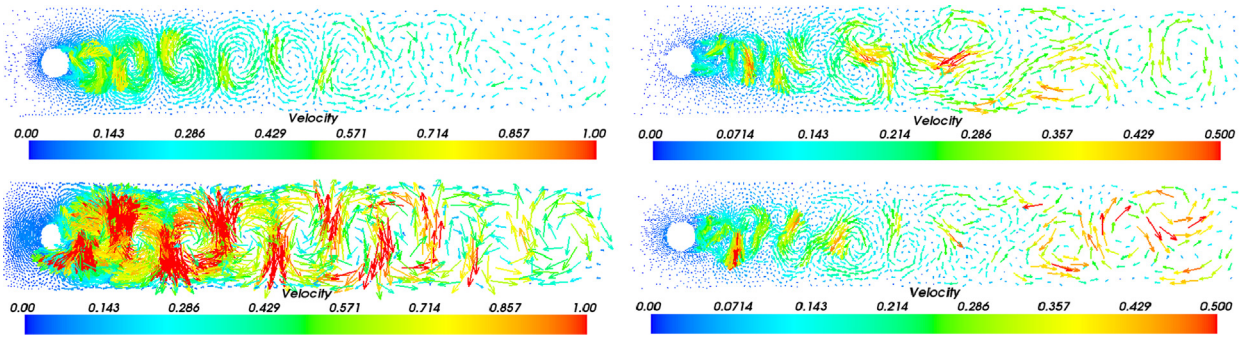


Fig. 8. Velocity errors of the flow past a cylinder problem at time instance 3.52 and 10.0. The solutions compare the errors in the residual DEIM model (left) and the quadratic expansion model (right). Both reduced order models used 12 POD functions.

The gyre was simulated through the full finite element model for a period of 194 days using a time step size of $\Delta t = 0.3311$ days. From this simulation 120 snapshots of the solution and non-linear terms were recorded and from this data 12 POD basis functions were generated. It was found that the POD basis set of this size captured over 99% of the energy of the u, v and p snapshot data. The problem was then re-simulated using the reduced order models with their non-linear terms represented through the quadratic expansion and the residual DEIM methods. Fig. 11 presents the velocity profiles obtained through the full model at time instances 91 and 149 days, and these show that the problem has formed several complex flow patterns involving a number of eddies. Included in the figures are the respective solutions obtained through the two reduced order models. Whilst the quadratic expansion method has performed well by resolving the general profile of the solution at these two time instances, some of the finer detail and smaller eddies were not completely captured. These finer solution details were however resolved through application of the residual DEIM approach. In fact the solutions between the residual DEIM and full model are almost visually identical. The errors between the two reduced models and the full solutions are presented in Fig. 12. Again these show the residual DEIM method to be more accurate than the quadratic

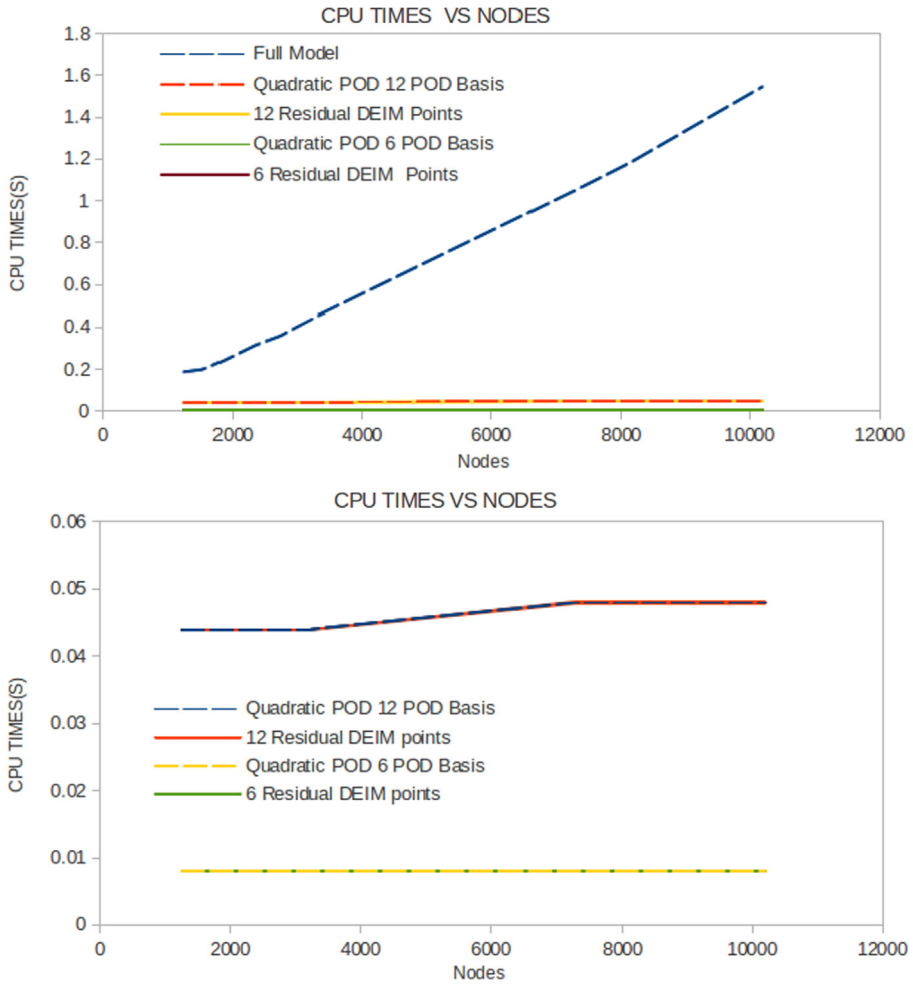


Fig. 9. Computational times to compute each time step as a function of mesh size (number of nodes) in the full model. Comparisons are made between the full model and the two ROMS.

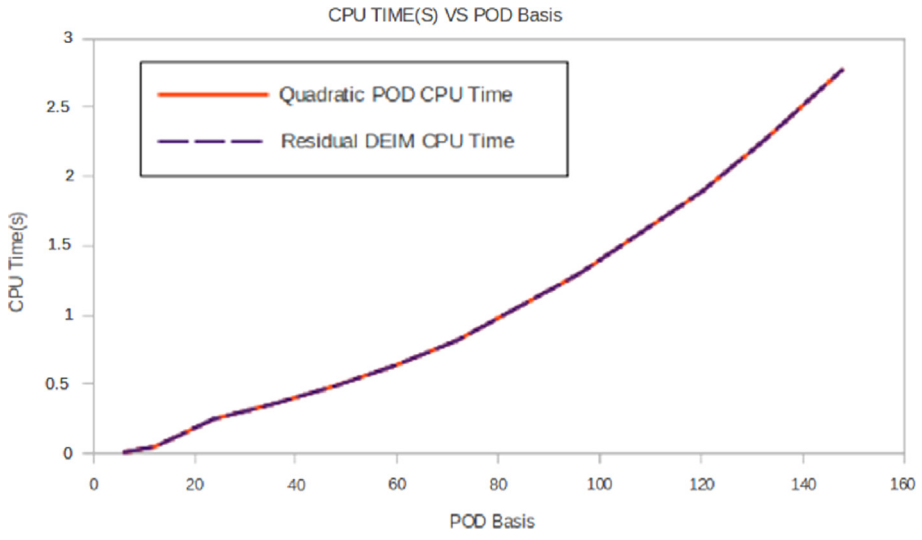


Fig. 10. Computational times to compute each time step as different number of POD basis in the ROM.

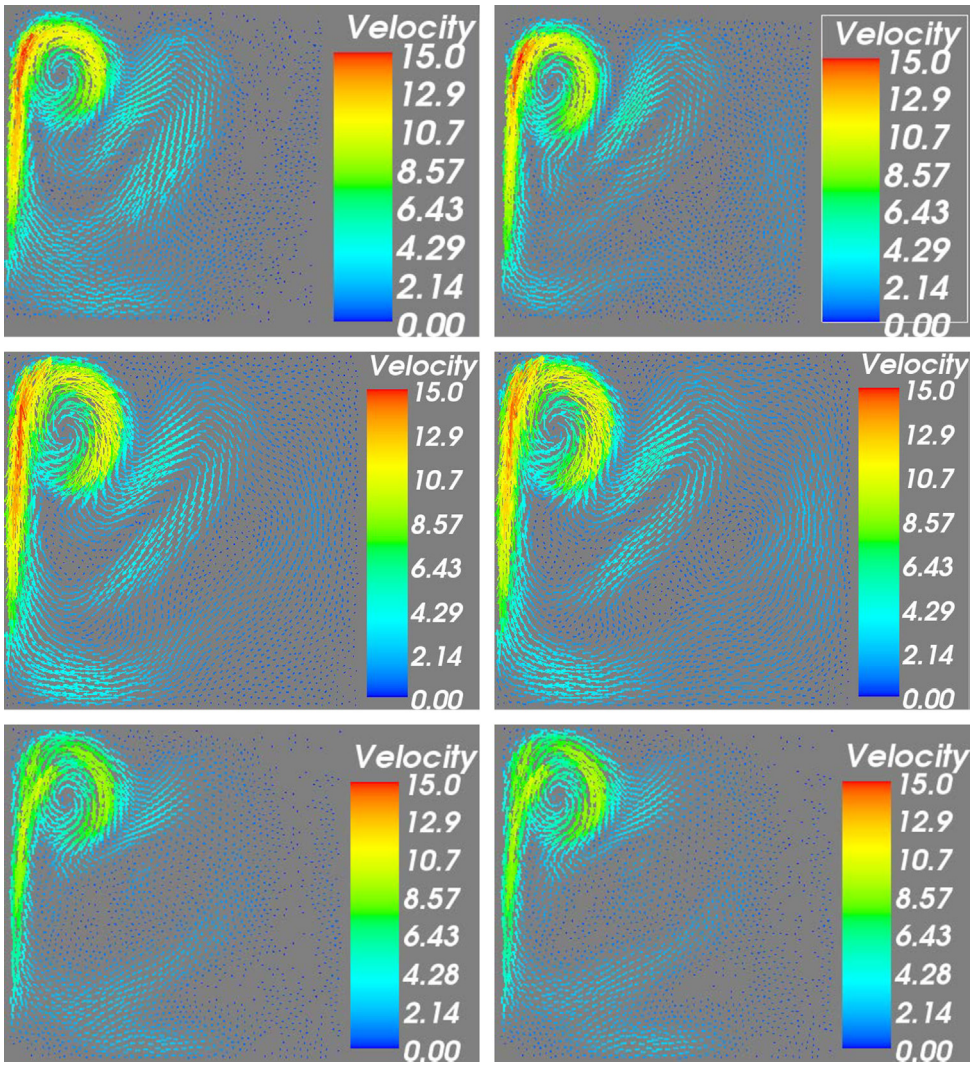


Fig. 11. Solutions of the gyre problem at time instance 91 (left) and 149 days (right). The solutions compare the predictions from the full model (top), the residual DEIM model (middle) and the quadratic expansion model (bottom). Both reduced order models used 12 POD functions.

expansion method. It is shown that the main gyre is more accurately resolved using residual DEIM, but in addition the eddies around the central top region of the problem contain less errors.

6. Conclusions

In this article a new reduced order model based upon Proper Orthogonal Decomposition (POD) has been presented. The method is centered on resolving the incompressible Navier–Stokes equation and the novelty of the approach is in how the non-linear terms of the equations are resolved. The treatment of the non-linear terms within a reduced order model requires special attention since the computational costs still depend on the number of variables in the full system. Instead additional techniques such as the quadratic expansion method and DEIM have been developed in order to maintain the ROM's efficiency. In this article a new hybrid scheme has been developed that mixes these two approaches. This is based on initially applying the quadratic expansion method to the non-linear terms and then applying the DEIM approach to resolve the residual between it and the full model. That is, the DEIM is used to mop up the remaining errors left over from the quadratic expansion approach.

This new method, named residual DEIM, has been applied to two 2D fluid flow problems and compared to the ROM approach using a quadratic expansion. The two problems were based on the simulation of flow past a cylinder and wind driven gyres, both of which were of sufficient difficulty with Reynolds numbers large enough to form complex flow patterns and eddies. In these demonstrations the residual DEIM approach showed strong capabilities in resolving the complex flows efficiently. It was also shown to improve the solution obtained from the ROM model using only quadratic expansions of

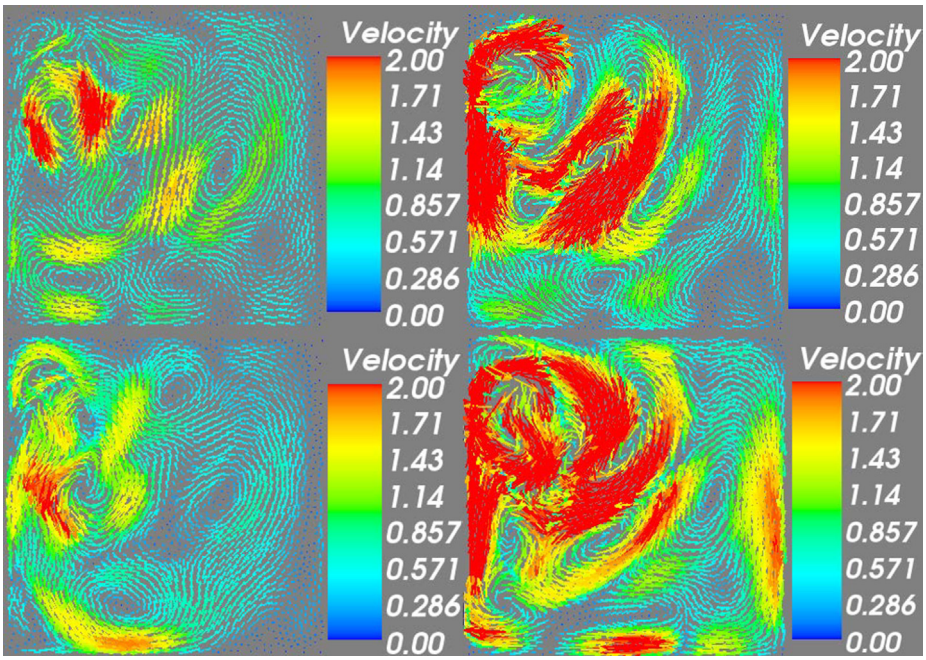


Fig. 12. Solution errors of the gyre problem at time instance 91 (top) and 149 (bottom) days. The solutions compare the errors in the residual DEIM model (left) and the quadratic expansion model (right). Both reduced order models used 12 POD functions.

the non-linear terms. In addition, the reduced order models were developed from full models involving unstructured finite element meshes. It has been previously observed that unstructured meshes can cause stability issues for reduced order models, but this was not the case for the residual DEIM approach.

Finally, it has been shown that the computational costs of the residual DEIM method is similar to that of DEIM and the quadratic expansion method. In addition, in the examples presented the computational times were reduced in comparison to the full model calculations, where for the larger number of mesh points tested the CPU costs were reduced by up to 98%. Future work will monitor an a-posteriori error estimate of this approach.

Acknowledgements

This work was carried out under funding from the UK's Natural Environment Research Council (projects NER/A/S/2003/00595, NE/C52101X/1 and NE/C51829X/1), the Engineering and Physical Sciences Research Council (GR/R60898, EP/I00405X/1 and EP/J002011/1), and the Imperial College High Performance Computing Service. Prof. I.M. Navon acknowledges the support of NSF/CMG grant ATM-0931198. Dunhui Xiao acknowledges the support of China Scholarship Council. Andrew Buchan wishes to acknowledge the EPSRC for funding his contribution to this article through the grant ref: EP/J002011/1. The authors acknowledge the reviewers and Editor for their in depth perspicacious comments that contributed to improving the presentation of this paper. The authors acknowledge Dr. R. Stefanescu's advice on the efficient implementation of the DEIM method.

References

[1] K. Fukunaga, Introduction to Statistical Recognition, second ed., Computer Science and Scientific Computing Series, Academic Press, Boston, MA, 1990, pp. 5–33.
[2] K. Pearson, On lines and planes of closest fit to systems of points in space, Philos. Mag. 2 (1901) 559–572.
[3] D.T. Crommelin, A.J. Majda, Strategies for model reduction: Comparing different optimal bases, J. Atmos. Sci. 61 (2004) 2206–2217.
[4] I.T. Jolliffe, Principal Component Analysis, second ed., Springer, 2002, pp. 559–572.
[5] Y. Cao, J. Zhu, I.M. Navon, Z. Luo, A reduced order approach to four dimensional variational data assimilation using proper orthogonal decomposition, Int. J. Numer. Methods Fluids 53 (2007) 1571–1583.
[6] P.T.M. Vermeulen, A.W. Heemink, Model-reduced variational data assimilation, Mon. Weather Rev. 134 (10) (2006) 2888–2899.
[7] D.N. Daescu, I.M. Navon, A dual-weighted approach to order reduction in 4d-var data assimilation, Mon. Weather Rev. 136 (3) (2008) 1026–1041.
[8] X. Chen, F. Fang, I.M. Navon, A dual weighted trust-region adaptive POD 4D-Var applied to a finite-volume shallow-water equations model, Int. J. Numer. Methods Fluids 65 (2011) 520–541.
[9] X. Chen, S. Akella, I.M. Navon, A dual weighted trust-region adaptive POD 4D-Var applied to a finite-volume shallow-water equations model on the sphere, Int. J. Numer. Methods Fluids 68 (2012) 377–402.
[10] M.U. Altaf, Model reduced variational data assimilation for shallow water flow models, PhD thesis, Delft University of Technology, 2011.
[11] J. Du, F. Fang, C.C. Pain, I.M. Navon, J. Zhu, D.A. Ham, POD reduced-order unstructured mesh modeling applied to 2D and 3D fluid flow, Comput. Math. Appl. 65 (2013) 362–379.

[12] F. Fang, C.C. Pain, I.M. Navon, A.H. ElSheikh, J. Du, D. Xiao, Non-linear Petrov–Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods, J. Comput. Phys. 234 (2013) 540–559.
[13] C.C. Pain, M.D. Piggott, A.J.H. Goddard, et al., Three-dimensional unstructured mesh ocean modelling, Ocean Model. 10 (2005) 5–33.
[14] N.C. Nguyen, J. Peraire, An efficient reduced-order modeling approach for non-linear parametrized partial differential equations, Int. J. Numer. Methods Eng. 76 (2008) 27–55.
[15] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, C. R. Acad. Sci. Paris, Ser. 339 (2004) 667–672.
[16] R. Stefanescu, I.M. Navon, POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model, J. Comput. Phys. 237 (2013) 95–114.
[17] S. Chaturantabut, Dimension reduction for unsteady nonlinear partial differential equations via empirical interpolation methods, Master's thesis, Rice University, 2008.
[18] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (2010) 2737–2764.
[19] S. Chaturantabut, D.C. Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction, SIAM J. Numer. Anal. 50 (2012) 46–63.
[20] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations – application to transport and continuum mechanics, Arch. Comput. Methods Eng. 15 (3) (2008) 229–275.
[21] N.C. Nguyen, G. Rozza, A.T. Patera, Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers equation, Calcolo 46 (3) (2009) 157–185.
[22] S. Boyaval, C. Le Bris, T. Lelièvre, Y. Maday, N.C. Nguyen, A.T. Patera, Reduced basis techniques for stochastic problems, Arch. Comput. Methods Eng. 17 (4) (2010) 435–454.
[23] J.L. Eftang, D.J. Knezevic, A.T. Patera, An hp certified reduced basis method for parametrized parabolic partial differential equations, Math. Comput. Model. Dyn. Syst. 17 (4) (2011) 395–422.
[24] Gianluigi Rozza, Andrea Manzoni, Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics, in: Proceedings of ECCOMAS CFD, 2010.
[25] G. Rozza, A. Manzoni, A. Quarteroni, Shape optimization for viscous flows by reduced basis methods and free-form deformation, Int. J. Numer. Methods Fluids 70 (5) (2012) 646–670.
[26] F. Negri, G. Rozza, A. Manzoni, A. Quarteroni, Reduced basis method for parametrized elliptic optimal control problems, SIAM J. Sci. Comput. 35 (5) (2013) A2316–A2340.
[27] R. Everson, L. Sirovich, The Karhunen–Loeve procedure for gappy data, J. Opt. Soc. Am. 12 (1995) 1657–1664.
[28] Karen Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, Comput. Fluids 35 (2) (2006) 208–226.
[29] C. Bou-Mosleh, K. Carlberg, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Int. J. Numer. Methods Eng. 86 (2011) 155–181.
[30] D. Xiao, F. Fang, J. Du, C.C. Pain, I.M. Navon, A.G. Buchan, A.H. ElSheikh, G. Hu, Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair, Comput. Methods Appl. Mech. Eng. 255 (2013) 147–157.
[31] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 72 (12) (2013) 1219–1243.
[32] N. Aubry, P. Holmes, J.L. Lumley, et al., The dynamics of coherent structures in the wall region of a turbulent boundary layer, J. Fluid Mech. 192 (1) (1988) 115–173.
[33] S. Chaturantabut, Nonlinear model reduction via discrete empirical interpolation, PhD thesis, Rice University, 2011.