Journal of Computational Physics 230 (2011) 126–146

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Two-level discretizations of nonlinear closure models for proper orthogonal decomposition

Z. Wang [a,*], I. Akhtar [b], J. Borggaard [a], T. Iliescu [a]

[a] Virginia Polytechnic Institute and State University, Department of Mathematics, Blacksburg, VA 24061, United States
[b] Virginia Polytechnic Institute and State University, Interdisciplinary Center for Applied Mathematics, Blacksburg, VA 24061, United States

ARTICLE INFO

Article history:
Received 10 March 2010
Received in revised form 26 July 2010
Accepted 10 September 2010
Available online 18 September 2010

Keywords:
Proper orthogonal decomposition
Nonlinear closure models
Two-level discretization

ABSTRACT

Proper orthogonal decomposition has been successfully used in the reduced-order modeling of complex systems. Its original promise of computationally efficient, yet accurate approximation of coherent structures in high Reynolds number turbulent flows, however, still remains to be fulfilled. To balance the low computational cost required by reduced-order modeling and the complexity of the targeted flows, appropriate closure modeling strategies need to be employed. Since modern closure models for turbulent flows are generally nonlinear, their efficient numerical discretization within a proper orthogonal decomposition framework is challenging. This paper proposes a two-level method for an efficient and accurate numerical discretization of general nonlinear closure models for proper orthogonal decomposition reduced-order models. The two-level method computes the nonlinear terms of the reduced-order model on a coarse mesh. Compared with a brute force computational approach in which the nonlinear terms are evaluated on the fine mesh at each time step, the two-level method attains the same level of accuracy while dramatically reducing the computational cost. We numerically illustrate these improvements in the two-level method by using it in three settings: the one-dimensional Burgers equation with a small diffusion parameter $\nu = 10^{-3}$, the two-dimensional flow past a cylinder at Reynolds number $Re = 200$, and the three-dimensional flow past a cylinder at Reynolds number $Re = 1000$.

1. Introduction

Proper orthogonal decomposition (POD) is one of the primary methods to construct low-order bases for model reduction of complex, nonlinear problems. The POD has its roots in statistics [1,2] and was initially used in the study of coherent structures in turbulent flows [3,4]. Over the past three decades, POD has been steadily developed in two main directions: (i) understanding the fundamental mechanisms in fluid flows, e.g., [5–13]; and (ii) developing *reduced-order models (ROMs)*, i.e., models that predict the evolution of the dominant structures in the underlying complex system at a very low computational cost [14–31]. In this report, we are concerned with the latter.

POD has had a long and rich history in the reduced-order modeling of complex systems. Its use as a simulation tool is a tremendously active research area. With the risk of being subjective, we list some of these applications below. Arguably the most dynamic research area in which POD has been used as a reduced-order modeling technique is fluid flow control [32–41]. Other applications include climate modeling and data assimilation of atmospheric and oceanic flows [42–45], stochastic PDEs [46,47], surrogate models for optimization [48–50], the complex Ginzburg–Landau equations [5,51], groundwater

* Corresponding author. Tel.: +1 540 808 8874; fax: +1 540 231 5960.
E-mail addresses: wangzhu@vt.edu (Z. Wang), akhtar@vt.edu (I. Akhtar), jborggaard@vt.edu (J. Borggaard), iliescu@vt.edu (T. Iliescu).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.09.015

models [52], control texture in the growth of polycrystals [53], pectoral fin kinematics [54], nonlinear structural systems [55–57], structural vibration [58–61], vehicle interior acoustics [62], structural health monitoring [63], ecosystems [64], and cell population models [65]. Several of these works cite the need to explicitly account for the influence of truncation or slow separation in the POD spectrum, e.g., [24,36,57,65]. The importance of reduced-order modeling in all these applications cannot be overemphasized. Indeed, the POD based ROMs (which we will denote in the sequel POD–ROMs) dramatically reduce the computational cost of a brute force numerical simulation, which for some of the above applications (e.g., climate modeling) is clearly unfeasible.

Despite its success as a general reduced-order modeling tool for complex systems, POD has yet to fulfill its original promise of computationally efficient, yet accurate approximation of coherent structures in high Reynolds number turbulent flows, as put forth in the pioneering work [14]. Indeed, in this context, the ultimate goal of POD–ROM is to capture the coherent structures in turbulent flows at high Reynolds numbers for time intervals that are much longer than those over which the *snapshots* (i.e., the flow field data at different time instants) were collected and at parameter values (e.g., Reynolds number) that are different from those at which the snapshots were generated. To our knowledge, this goal has not yet been achieved. We believe that one of the main reasons is the following. To balance the low computational cost required by a ROM and the complexity of the targeted turbulent flows, appropriate closure modeling strategies need to be employed. Since modern closure models for turbulent flows are generally nonlinear, their efficient numerical discretization within a POD framework is challenging. Indeed, most POD–ROMs employ precomputed matrices and tensors, which precludes a straightforward discretization of nonlinear closure model terms. This report aims at developing a computationally efficient numerical discretization of general, nonlinear closure modeling strategies for POD–ROMs. Before describing this new methodology in detail, we feel that the following remarks regarding the very scope of this report are needed.

We start by emphasizing that, in this paper, we do *not* answer and do *not* try to answer the question of whether or not POD–ROM can be employed to simulate dominant coherent structures in high Reynolds number turbulent flows. We also do *not* find the best closure model for POD–ROM and we do *not* assess the robustness of the POD–ROM with respect to changes in the Reynolds number or length of simulation time. Instead, we take a more modest approach and try to elucidate whether nonlinear closure modeling strategies can be utilized efficiently in a POD context. Although a modest step, we believe that this is a *necessary* step in answering the above open question. Indeed, we believe that without employing modern closure modeling strategies, which are generally nonlinear, POD–ROM does not stand any chance in successfully modeling dominant coherent structures in high Reynolds number turbulent flows.

We also emphasize that POD is *not* intended to compete against well established, successful methodologies for the numerical simulation of general turbulent fluid flows, such as *large eddy simulation (LES)* [66,67]. In fact, such a comparison between POD–ROM and LES would be questionable. Indeed, as shown in [17], POD reduces to the standard Fourier decomposition when homogeneous flows are considered. Thus, for homogeneous, isotropic turbulent flows, POD–ROM (and the associated closure modeling) practically reduces to LES. This clearly suggests that POD–ROM and LES have different goals: the latter targets general, homogeneous, isotropic turbulent flows, whereas the former aims at highly anisotropic turbulent flows, whose dynamics are dominated by *coherent structures*. Moreover, the spatial resolutions employed by POD–ROM are much coarser than those used in LES. These suggest that POD–ROM and LES could actually be used together: Indeed, for turbulent flows at high Reynolds numbers (for which a direct numerical simulation (DNS) is unfeasible), LES could be used to generate the snapshot matrix for the POD–ROM. This idea has been exploited in [12].

We also note that the POD–ROM approach that we pursue in this report has several similarities to the stochastic adaptive large eddy simulation (SCALES) introduced in [68]. SCALES combines the coherent vortex simulation (CVS) methodology proposed in [69,70] with subgrid-scale modeling ideas imported from LES. Specifically, the CVS applies a wavelet filter to a turbulent vorticity field and separates it into energetic coherent structures and a residual, subgrid field that is homogeneous. To overcome the relatively high computational cost of CVS [71], SCALES employs a wider support for the wavelet filter and then takes advantage of LES-type modeling techniques to account for the effect of those coherent structures that are now treated as subgrid-scale terms. Thus, POD–ROM and SCALES are similar in that they both target the coherent structures in the underlying turbulent flow. They are different, however, in that POD–ROM employs POD to generate these coherent structures, whereas SCALES uses wavelets. Moreover, POD–ROM aims at highly non-homogeneous, anisotropic turbulent flows, whereas SCALES considers general turbulent flows. Finally, POD–ROM employs spatial resolutions that are coarser than those used by SCALES and CVS.

The last point we emphasize is that, although all numerical tests in this report involve fluid flow problems, the two-level methodology we propose can be applied to *any nonlinear* closure model and *any* application in which POD is used as a reduced-order modeling technique.

We now present the new two-level methodology for the numerical discretization of nonlinear closure modeling in POD–ROMs. We start by briefly describing the POD methodology (for more details, the reader is referred to [4,17]). To this end, we consider the numerical solution of the incompressible *Navier–Stokes equations (NSE)*:

$$\begin{cases} \mathbf{u}_t - Re^{-1}\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = 0, \\ \nabla\cdot\mathbf{u} = 0, \end{cases} \tag{1}$$

where $\mathbf{u}$ is the velocity, p the pressure and Re the Reynolds number. The POD basis is generated by post-processing *typical* data from the numerical simulation of (1). If $\mathcal{Y} = \{\mathbf{y}(\cdot,t) \in \mathcal{H} \mid t \in (0,T)\}$ (with $\mathcal{H}$ a Hilbert space) represents a

simulation of the NSE, then the first POD basis vector is the function that maximizes the time-averaged projection of $\mathcal{Y}$ onto itself:

$$\boldsymbol{\phi}_1 = \max_{\boldsymbol{\phi}\in\mathcal{H},\|\boldsymbol{\phi}\|_{\mathcal{H}}=1} \frac{1}{T}\int_0^T \left|\langle \mathbf{y}(\cdot,t),\boldsymbol{\phi}(\cdot)\rangle_{\mathcal{H}}\right|^2 \, dt. \tag{2}$$

Subsequent vectors, $\boldsymbol{\phi}_k$, are determined by seeking the above maximum in the orthogonal complement to:

$$\Phi_{k-1} = \operatorname{span}\{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_{k-1}\}, \quad 2 \leqslant k \leqslant N, \quad \text{in } \mathcal{H}, \tag{3}$$

where N is the rank of $\mathcal{Y}$. If we choose $\mathcal{H} = \mathcal{L}_2$ and $\mathcal{Y}$ represents a single simulation, the POD basis functions satisfy the Fredholm integral equation:

$$\int_\Omega \mathbf{R}(\mathbf{x},\mathbf{x}')\boldsymbol{\phi}_i(\mathbf{x}') \, d\mathbf{x}' = \lambda_i \boldsymbol{\phi}_i(\mathbf{x}), \tag{4}$$

where

$$\mathbf{R}(\mathbf{x},\mathbf{x}') = \frac{1}{T}\int_0^T \mathbf{y}(\mathbf{x},t)\mathbf{y}^*(\mathbf{x}',t) \, dt, \tag{5}$$

is the spatial autocorrelation kernel. There are natural extensions of this definition that accommodate multiple simulations. In practice, either the time average of each simulation or the steady state solution is removed, so that $\mathcal{Y}$ contains fluctuation from the mean (or centering trajectory), e.g., $\mathbf{y}(\mathbf{x},t) = \mathbf{u}(\mathbf{x},t) - \bar{\mathbf{u}}(\mathbf{x})$, cf. [17,72]. Note that each POD basis vector $\boldsymbol{\phi}_k$ represents a weighted time average of the data $\mathcal{Y}$. Thus, these basis vectors preserve linear properties (such as the divergence-free property).

A POD basis enables a reduced representation of the simulated data, and thus can be viewed as a compression algorithm. Utilizing the POD basis to obtain efficient approximations to (1) is achieved using the POD basis in a Galerkin approximation, and employing the fact that the POD basis vectors are mutually orthogonal. A *reduced-order model (ROM)* of the flow is constructed from the POD basis by writing:

$$\mathbf{u}(\mathbf{x},t) \approx \mathbf{u}_r(\mathbf{x},t) \equiv \bar{\mathbf{u}}(\mathbf{x}) + \sum_{j=1}^{r} a_j(t)\boldsymbol{\phi}_j(\mathbf{x}), \tag{6}$$

where $\bar{\mathbf{u}}(\mathbf{x})$ is the centering trajectory, $\{\boldsymbol{\phi}_j\}_{j=1}^r$ are the first r POD basis vectors, and $\{a_j(t)\}_{j=1}^r$ are the sought time-varying coefficients that represent the POD–Galerkin trajectories.

A *dynamical system* for the vector of time coefficients, $\mathbf{a}(t)$, is constructed using (6) and a Galerkin projection of (1) onto Φ_r. This leads to an autonomous system of the form:

$$\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A}\mathbf{a} + \mathbf{a}^T\mathbf{B}\mathbf{a}, \tag{7}$$

where $\mathbf{b}$, $\mathbf{A}$, and $\mathbf{B}$ correspond to the constant, linear, and quadratic terms in the numerical discretization of the NSE (1), respectively. The initial conditions are obtained by projection:

$$a_j(0) = \left\langle \boldsymbol{\phi}_j, \mathbf{u}(\cdot,0) - \bar{\mathbf{u}}(\cdot)\right\rangle_{\mathcal{H}}, \quad j = 1,\ldots,r. \tag{8}$$

It is important to note that the quadratic nonlinearity in (1) allows for easy precomputation of the vector $\mathbf{b}$, the matrix $\mathbf{A}$ and the tensor $\mathbf{B}$, leading to efficient ROMs.

The dynamical system (7) is often accurate for laminar flows at parameter values that are well represented in the input collection. From the earliest stages of POD of *turbulent* flows [14], it was recognized that a simple Galerkin truncation will generally produce inaccurate results, even if the retained modes capture most of the system's energy [15]. Thus, *closure modeling* (i.e., modeling the effect of the discarded POD modes $\{\boldsymbol{\phi}_{r+1},\ldots,\boldsymbol{\phi}_N\}$ on the modes retained in the ROM $\{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_r\}$) has always played a central role in POD reduced-order model strategies. Closure modeling has developed in two main directions (that often overlap): (i) improving the *numerical stability* [24–26,30,73,74]; and (ii) improving the *physical accuracy* [9,14,16–21,75]. This paper is concerned with the latter. There are several new strategies that aim at more physical closure models, e.g., the finite-time thermodynamics formalism of Noack et al. [76]. The closure models that we consider are similar to those used in LES of *turbulent* flows and target: (i) *a severe truncation of the POD basis*, (ii) *a wide range of Reynolds numbers*, and (iii) *time intervals longer than those over which snapshots were collected*. These features are essential for the POD reduced-order modeling of turbulent flows.

The first closure model for POD - used in the pioneering work of Aubry et al. [14] – was the *mixing length* model, one of the simplest models used in LES of turbulent flows [66,67,77]. This model employs the concept of an energy cascade and represents the effect of the discarded POD modes by adding a dissipative term to the ROM. This POD closure model has been used in numerous studies [9,16,18–21,75]. The validity of the energy cascade concept in a POD setting was confirmed in recent numerical studies [12,78]. This study, together with the long list of recent reports in which variations of the simple mixing length eddy viscosity model in [14] were successfully used, clearly indicate that LES ideas could be used in the development of closure models for POD. One of the main computational hurdles in the development of more sophisticated

closure models for POD–ROMs of coherent structures in turbulent flows has been the lack of efficient computational strategies for the discretization of *nonlinear* closure modeling terms.

Indeed, the main advantage of the ROM (7) over the usual discretizations of (1) (such as finite elements or finite differences) is that the dimension of the dynamical system (7) is usually extremely small. Thus, the time integration of (7) can be effected very efficiently. There is, however, a caveat. *The matrices in (7) are computed only once at the beginning of the simulation.* This is essential for the computational efficiency of the ROM. Indeed, if we had to recompute the matrices at every time step, this would increase the computational time dramatically, since these matrices are *full* and the POD basis functions ϕ_i are *global*. Thus, current POD closure models use matrices that are *precomputed*. *This restriction has been a major computational hurdle in the development of modern closure strategies for POD of turbulent flows dominated by coherent structures.* Many of these closure modeling strategies introduce nonlinear, non-polynomial terms that need to be efficiently approximated. Existing methods to treat terms of this form are interpolatory and thus not directly amenable to the chaotic nature of turbulent flows.

In this paper, we propose a novel way to overcome this major computational hurdle. We emphasize that this paper is *not* concerned with the derivation of improved POD closure models [79]. Instead, for a given closure model (one of the most basic LES closure models), we develop a new algorithm that allows an efficient and accurate numerical discretization of that POD closure model. Our new approach is based on the observation that only a small subset of the original POD basis is used in the ROM and this basis is typically dominated by coarse, large scale structures. Thus, the numerical discretization of the nonlinear closure term can be effected on a mesh at a resolution that is significantly lower than that of the mesh utilized in the derivation of the full POD basis. Based on the above observation, we develop a two-level algorithm for the discretization of the nonlinear closure model. This two-level algorithm allows for an efficient computation of the nonlinear POD closure model at each time step without compromising its accuracy. Numerical results supporting our new approach are presented for three test problems.

The rest of the paper is organized as follows. In Section 2, we describe the new two-level algorithms. Section 3 presents results using these new algorithms on three test problems: a one-dimensional (1 D) Burgers equation with a very small viscosity parameter ($\nu = 10^{-3}$), a two-dimensional (2 D) flow past a cylinder at a Reynolds number $Re = 200$ and a three-dimensional (3 D) turbulent flow past a cylinder at a moderate Reynolds number $Re = 1000$. Conclusions are presented in Section 4.

2. Two-level algorithms

In this section, we describe two-level algorithms for efficient and accurate discretization of nonlinear POD closure models.

We start by describing the standard POD Galerkin algorithm. Using a Galerkin projection of the NSE (1) onto the space Φ_r defined in (3), we obtain the finite dimensional system (7). This system can be written componentwise as follows: For all $k = 1,\ldots,r$:

$$\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km} a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t), \tag{9}$$

where

$$b_k = -(\phi_k, \bar{\mathbf{u}}\cdot\nabla\bar{\mathbf{u}}) - \frac{2}{Re}\left(\nabla\phi_k, \frac{\nabla\bar{\mathbf{u}} + \nabla\bar{\mathbf{u}}^T}{2}\right), \tag{10}$$

$$A_{km} = -(\phi_k, \bar{\mathbf{u}}\cdot\nabla\phi_m) - (\phi_k, \phi_m\cdot\nabla\bar{\mathbf{u}}) - \frac{2}{Re}\left(\nabla\phi_k, \frac{\nabla\phi_m + \nabla\phi_m^T}{2}\right), \tag{11}$$

$$B_{kmn} = -(\phi_k, \phi_m\cdot\nabla\phi_n). \tag{12}$$

The *POD Galerkin (POD-G)* algorithm reads:

$$\begin{array}{l} \ell = 0; \quad \texttt{compute } \mathbf{b}, \mathbf{A}, \mathbf{B}; \\ \texttt{for } \ell = 0 \quad \texttt{to } M-1 \\ \quad \mathbf{a}^{\ell+1} := F(\mathbf{a}^{\ell}); \\ \texttt{endfor} \end{array} \tag{13}$$

POD-G algorithm

where the superscript ℓ denotes the current time step, M is the total number of time steps, and F is the function corresponding to the time-advancement method (e.g., Euler's method).

As mentioned in the introduction, closure modeling is needed when using the POD-G model (7) for turbulent flows. One of the most popular closure models used in large eddy simulation of turbulent flows is the Smagorinsky model [80], which augments the diffusion coefficient by $C\|\mathbb{D}(\mathbf{u})\|$, where $\|\mathbb{D}(\mathbf{u})\|$ is the Frobenius norm of the deformation tensor $\mathbb{D}(\mathbf{u}) = \frac{\nabla\mathbf{u}+\nabla\mathbf{u}^T}{2}$. Thus, the weak form of the NSE (1) equipped with the Smagorinsky model now reads: For all $i = 1,\ldots,r$:

$$(\mathbf{u}_t, \phi_i) + \left(\left(\frac{2}{Re} + 2C\|\mathbb{D}(\mathbf{u})\|\right)\mathbb{D}(\mathbf{u}), \nabla\phi_i\right) + ((\mathbf{u}\cdot\nabla)\mathbf{u}, \phi_i) = 0. \tag{14}$$

We note that, since we are using the non-dimensionalized NSE (1), the constant C in (14) is non-dimensional. Since the model used is the Smagorinsky model, we are motivated to identify the term above with $(C_S\delta)^2\|\mathbb{D}(\mathbf{u})\|$. This leads to the identification of the corresponding C_S and δ values using the following equation:

$$C = (C_S\delta)^2\sqrt{2}, \tag{15}$$

where the factor $\sqrt{2}$ comes from the formula used in the Smagorinsky model for the computation of the Frobenius norm [66]. To determine δ in (15), we use a definition similar to that proposed in [14] (although other definitions can be used):

$$\delta := \left(\frac{\int_0^{L_1}\int_0^{L_2}\langle u_{i>}\, u_{i>}\rangle\, dx_1\, dx_2}{\int_0^{L_1}\int_0^{L_2}\langle u_{i>j}\, u_{i>j}\rangle\, dx_1\, dx_2}\right)^{1/2}, \tag{16}$$

where repeated indices denote summation, the subscript $>$ denotes unresolved POD modes (fluctuations), $\langle f\rangle = \frac{1}{L_3}\int_0^{L_3} f(\mathbf{x},t)\,dx_3$ denotes the spatial average of f in the homogeneous direction, and L_1 and L_2 are the two inhomogeneous directions of the computational domain. We note that formula (16) corresponds to the case of 3 D flow past a cylinder (Section 3.3); for the 2 D case (Section 3.2), the obvious modifications should be made. Using a Galerkin truncation $\mathbf{u}(\mathbf{x},t)\approx\mathbf{u}_r(\mathbf{x},t)\equiv\bar{\mathbf{u}}(\mathbf{x})+\sum_{j=1}^{r}a_j(t)\boldsymbol{\phi}_j(\mathbf{x})$ in (14), yields the following finite dimensional dynamical system:

$$\dot{\mathbf{a}} = \Big(\mathbf{b}+\tilde{\mathbf{b}}(\mathbf{a})\Big) + \Big(\mathbf{A}+\widetilde{\mathbf{A}}(\mathbf{a})\Big)\mathbf{a} + \mathbf{a}^T\mathbf{B}\mathbf{a}, \tag{17}$$

where $\tilde{\mathbf{b}}(\mathbf{a})$ and $\widetilde{\mathbf{A}}(\mathbf{a})$ correspond to the discretization of the nonlinear closure model. In componentwise form, system (17) can be written as

$$\dot{a}_k(t) = \Big(b_k+\tilde{b}_k(\mathbf{a})\Big) + \sum_{m=1}^{r}\Big(A_{km}+\widetilde{A}_{km}(\mathbf{a})\Big)a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r}B_{kmn}a_n(t)a_m(t), \tag{18}$$

where b_k, A_{km}, and B_{kmn} are the same as those in Eq. (9) and

$$\tilde{b}_k(\mathbf{a}) = -2C\left(\nabla\boldsymbol{\phi}_k, \|\mathbb{D}(\mathbf{u})\|\frac{\nabla\bar{\mathbf{u}}+\nabla\bar{\mathbf{u}}^T}{2}\right),$$
$$\widetilde{A}_{km}(\mathbf{a}) = -2C\left(\nabla\boldsymbol{\phi}_k, \|\mathbb{D}(\mathbf{u})\|\frac{\nabla\boldsymbol{\phi}_m+\nabla\boldsymbol{\phi}_m^T}{2}\right).$$

We emphasize that we are concerned with closure models (such as (14)) that *cannot* be cast as a quadratic nonlinearity and thus cannot be simply included in $\mathbf{B}$. The brute force approach in the numerical discretization of (17) yields the *one-level algorithm*:

$\ell = 0$; compute $\mathbf{b},\mathbf{A},\mathbf{B}$ on the fine mesh;
for $\ell = 0$ to $M-1$
  compute $\tilde{\mathbf{b}}(\mathbf{a}^\ell)$, $\widetilde{\mathbf{A}}(\mathbf{a}^\ell)$ on the fine mesh (19)
  $\mathbf{a}^{\ell+1} := \widetilde{F}(\mathbf{a}^\ell)$;
endfor

one-level algorithm

where $\widetilde{F}$ is a modification of F in (13) accounting for the additional nonlinear closure terms. Because we need to recompute (reassemble) $\tilde{\mathbf{b}}(\mathbf{a})$ and $\widetilde{\mathbf{A}}(\mathbf{a})$, algorithm (19) is impractical. Indeed, this algorithm removes a major advantage of ROM – the decrease of computational time. This is because although $r \ll N$, the POD basis functions are global and reassembling $\tilde{\mathbf{b}}(\mathbf{a})$ and $\widetilde{\mathbf{A}}(\mathbf{a})$ at each time step dramatically increases the CPU time.

To make algorithm (19) practical, we must avoid the computational price required by the matrix reassembly. The two-level algorithm we propose is based on the observation that only a small subset of the original POD basis is used in the ROM. Indeed, if a computation on a fine mesh of size h is employed to obtain the POD basis $\{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_N\}$, one is usually able to use a coarser mesh of size H to represent the POD basis $\{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_r\}$, where $r \ll N$. Fig. 1 clearly supports this statement. Indeed, the numerical resolution needed for an accurate approximation of $\boldsymbol{\phi}_{10}$ is *much lower* than that needed for the approximation of $\boldsymbol{\phi}_{70}$. We also note that a similar conclusion would be drawn if we used Fourier basis functions [17].

Thus, to avoid the high computational cost required by the one-level algorithm (19), we propose an algorithm that computes $\tilde{\mathbf{b}}(\mathbf{a})$ and $\widetilde{\mathbf{A}}(\mathbf{a})$ at each time step on a *coarse* mesh of size H, and not on the same fine mesh of size h used in the computation of the POD basis functions $\{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_N\}$. Thus, we present a *two-level algorithm*:

$\ell = 0$; compute $\mathbf{b},\mathbf{A},\mathbf{B}$ on the fine mesh;
for $\ell = 0$ to $M-1$
  compute $\tilde{\mathbf{b}}(\mathbf{a}^\ell)$, $\widetilde{\mathbf{A}}(\mathbf{a}^\ell)$ on the coarse mesh (20)
  $\mathbf{a}^{\ell+1} := \widetilde{F}(\mathbf{a}^\ell)$;
endfor

hybrid two-level algorithm

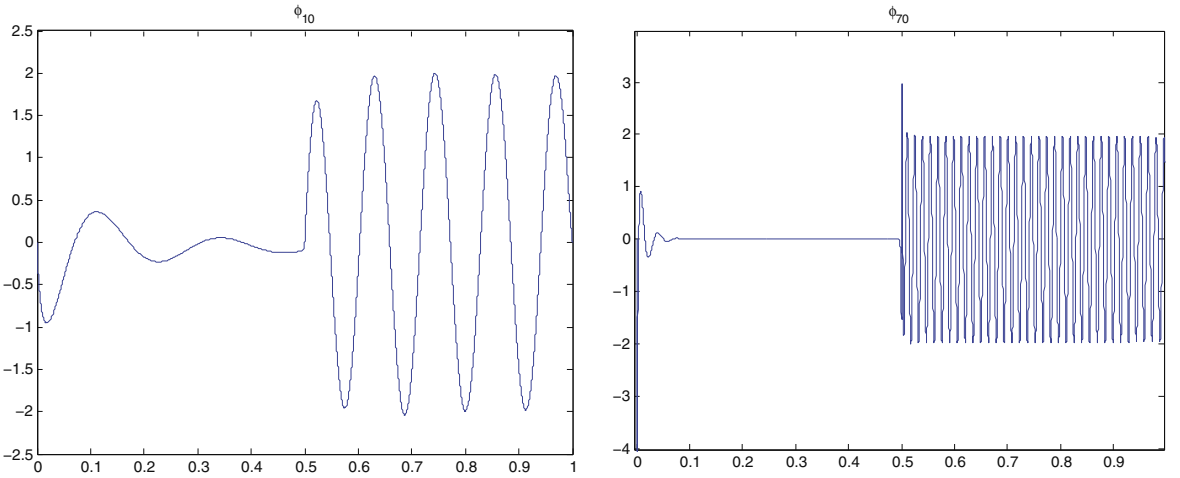

Fig. 1. 1 D Burgers equation with $\nu = 10^{-3}$. POD basis functions: (left) ϕ_{10}; (right) ϕ_{70}. Note that the numerical resolution needed for an accurate approximation of ϕ_{10} is *much lower* than that needed for the approximation of ϕ_{70}.

The advantage of the two-level algorithm (20) over the one-level algorithm (19) is obvious: the former achieves *the same level of accuracy* as the latter while decreasing the computational cost by an *order of magnitude*:

$$\boxed{\begin{array}{l}\texttt{error}_{\,\textit{two-level}} \sim \texttt{error}_{\,\textit{one-level}} \\ \texttt{CPU}_{\,\textit{two-level}} \ll \texttt{CPU}_{\,\textit{one-level}}\end{array}}$$

The dramatic reduction in CPU time for the two-level algorithm is due to the efficient computation of $\tilde{\mathbf{b}}(\mathbf{a})$ and $\widetilde{\mathbf{A}}(\mathbf{a})$ on the *coarse* mesh at *each* time step.

For completeness, we also investigate a modification of the *hybrid two-level algorithm* (20). This algorithm is denoted as the *coarse two-level algorithm*:

$$\boxed{\begin{array}{l}\ell = 0; \quad \texttt{compute } \mathbf{b}, \mathbf{A}, \mathbf{B} \texttt{ on the coarse mesh;} \\ \texttt{for } \ell = 0 \texttt{ to } M-1 \\ \quad \texttt{compute } \tilde{\mathbf{b}}(\mathbf{a}^\ell),\ \widetilde{\mathbf{A}}(\mathbf{a}^\ell) \texttt{ on the coarse mesh} \\ \quad \mathbf{a}^{\ell+1} := \widetilde{F}(\mathbf{a}^\ell); \\ \texttt{endfor}\end{array}} \quad (21) \quad \begin{array}{l}\texttt{coarse} \\ \texttt{two-level} \\ \texttt{algorithm}\end{array}$$

Note that the coarse two-level algorithm (21) is actually the one-level algorithm (19) applied to POD bases that have been interpolated on a coarse mesh. Thus, once the fine grid is used in the generation of the snapshots, it is no longer used in the coarse two-level algorithm (21).

A central role in our new two-level POD–ROM algorithm is played by the coarsening and interpolation procedures. Since this study represents the first investigation of the new two-level algorithm, we chose the *simplest* approach – uniformly skipping mesh points to coarsen the structured mesh and interpolate on the coarse mesh. We emphasize, however, that there are obvious ways to improve the coarsening and interpolation procedures; we plan to pursue these approaches in a future study.

We also note that the two-level methodology introduced here is different from modeling approaches in which the POD computed on a coarse mesh is used as a surrogate for the entire underlying system (see, e.g., [46]). Indeed, the POD basis in our two-level methodology is computed on the fine mesh. After extracting the POD basis from the fine mesh computation, we interpolate it on the coarse mesh and compute the POD closure model. Employing a POD basis computed directly on the coarse mesh yields highly inaccurate results for the complex flows considered in this report.

Next we present numerical experiments that test the improvement of the two-level algorithms (20) and (21) over the one-level algorithm (19).

3. Numerical results

In this section, we investigate the two-level algorithms in the numerical simulation of three test problems: (i) the 1 D Burgers equation with a small diffusion coefficient (Section 3.1); (ii) the 2 D flow past a circular cylinder at $Re = 200$ (Section 3.2); and (iii) the 3 D flow past a circular cylinder at $Re = 1000$ (Section 3.3). Both two-level algorithms (20) and (21) employ *two meshes*: (i) a *fine mesh* of size h (the mesh on which the POD basis $\{\phi_1, \ldots, \phi_N\}$ was computed); and (ii) a *coarse mesh* of size H (the mesh on which the computation of the POD closure model is effected at each time step). For a given fine

mesh of size h, we consider different coarse meshes with sizes H. The coarsening factor R_c represents the ratio H/h. We investigate whether the coarse two-level algorithm (21) and the hybrid two-level algorithm (20) can achieve the same order of accuracy as the standard one-level algorithm (19), while significantly decreasing the CPU time. The accuracy of the three algorithms is benchmarked against a DNS. The relative error in the algorithms is computed as

$$\text{error} = \frac{\frac{1}{L}\sum_{\ell=1}^{L}\left\|\mathbf{u}^{\text{POD-ROM}}(\mathbf{x},t_\ell)-\mathbf{u}^{\text{DNS}}(\mathbf{x},t_\ell)\right\|_0^2}{\frac{1}{L}\sum_{\ell=1}^{L}\left\|\mathbf{u}^{\text{DNS}}(\mathbf{x},t_\ell)\right\|_0^2}, \tag{22}$$

where L is the number of snapshots and $\|\cdot\|_0$ is the L^2 norm. To ensure the fairness of our numerical investigation, we also performed numerical experiments with the POD-G algorithm (13) (i.e., without any closure model) with a small number of POD basis functions (small r). Poor results with POD-G indicate the need for the closure modeling employed in the two-level and one-level algorithms. We also mention that, for both the one-level and the two-level algorithms, we only recompute $\tilde{\mathbf{b}}$ and $\widetilde{\mathbf{A}}$ every one hundred time steps. The reason is that, had we computed $\tilde{\mathbf{b}}$ and $\widetilde{\mathbf{A}}$ at every time step for the one-level algorithm, this would have increased the CPU time for the one-level algorithm beyond our present computational limits. We emphasize that these computational savings were achieved *without* sacrificing the numerical accuracy. Indeed, the same qualitative results (on a shorter time interval) were achieved by evaluating $\tilde{\mathbf{b}}$ and $\widetilde{\mathbf{A}}$ at every time step. This is not surprising, since we used a very small time step in the time discretization. In practical computations with the two-level algorithm (when the one-level algorithm does not need to be used) $\tilde{\mathbf{b}}$ and $\widetilde{\mathbf{A}}$ will be evaluated at every time step. Finally, we mention that in all the models tested (both POD-G and POD–ROM), the time discretization was effected by using the explicit Euler method with a small time-step. Since the main focus of the paper is the efficient spatial discretization for POD closure modeling, we chose this simple time discretization method; using the same time-discretization for both POD-G and POD–ROM allowed for a fair comparison. Obviously, higher-order methods can (and should) be used for further computational savings. We plan to pursue this research direction in a forthcoming study.

3.1. 1 D Burgers equation with $\nu = 10^{-3}$

In this section, we investigate the two-level algorithms in the numerical simulation of the 1 D *Burgers equation* with a small diffusion coefficient $\nu = 10^{-3}$:

$$u_t - \nu u_{xx} + u u_x = f. \tag{23}$$

We use a computational setting that is similar to that used by Kunisch and Volkwein [81]. The initial condition on domain $\Omega = [0,1]$ is defined as $u_0(x) = 1$ if $x \in (0,1/2]$ and $u_0(x) = 0$ otherwise. The forcing term is $f = 0$ and the time interval is $[0,T] = [0,1]$. The boundary conditions are homogeneous Dirichlet.

A DNS is used as benchmark. This is computed using the finite element method with piecewise linear polynomials to discretize the problem in space, the implicit Euler method to discretize in time, and Newton's method to solve the nonlinear system. We use a mesh-size $\Delta x = 1/8192$ and a time step $\Delta t = 10^{-3}$. A mesh refinement study indicates that the DNS convergence has been achieved. The CPU time for DNS is 6155 s.

We use the DNS data consisting of 1001 snapshots to generate the POD basis. This POD basis is then used in all POD reduced-order models that we investigate next.

The qualitative behavior of all the models tested is represented by the time evolution of the POD coefficients $a_1(\cdot)$ and $a_6(\cdot)$ plotted in Fig. 3. We note that the other POD coefficients have a similar behavior; for clarity of exposition, we included only $a_1(\cdot)$ and $a_6(\cdot)$. For comparison purposes, Fig. 3 include the time evolution of the POD coefficients obtained by projecting the DNS onto each POD mode.

The POD-G (13) with low r ($r = 10$) performs poorly, although it is computationally efficient (CPU time of 19 s). Indeed, the temporal evolutions of POD coefficients in Fig. 3 are inaccurate. The amplitude of the temporal evolution of the POD coefficient $a_6(\cdot)$ for POD-G with $r = 10$ is *two times larger* than that for DNS. The relative error of POD-G with $r = 10$ is 7.54×10^{-2}. Thus, although the first 10 POD modes capture most (99.44%) of the system's energy, an additional term to stabilize the computations is needed [82].

We now consider the Smagorinsky model (14) with an artificial viscosity coefficient $C = 7 \times 10^{-4}$ as a closure model. Although the Smagorinsky model targets 3 D turbulent flows, for this test problem it plays the role of numerical stabilization. For the numerical discretization of the resulting nonlinear POD–ROM (17), we will use three different strategies: (i) the one-level algorithm (19); (ii) the hybrid two-level algorithm (20); and (iii) the coarse two-level algorithm (21). In comparing the three algorithms, the time discretization was effected by using the explicit Euler method with $\Delta t = 10^{-5}$.

The one-level algorithm (19) with $r = 10$ yields the temporal evolution of POD coefficients in Fig. 3. Comparing these results with those for the DNS projection clearly shows that the one-level algorithm yields relatively accurate results although it uses only a few ($r = 10$) POD basis functions. Indeed, the one-level algorithm (19) produces results that are much closer to those for the DNS than the results for POD-G with $r = 10$. The relative error of the one-level algorithm is 2.55×10^{-2}, whereas that of POD-G with $r = 10$ is 7.54×10^{-2}. This, however, comes at a high computational cost. The CPU time is 19 s for POD-G and is 858 s for the one-level algorithm.

The coarsening and interpolation procedures used in the two-level algorithms (20) and (21) were straightforward: The coarsening of the mesh was effected by skipping nodes and the interpolation of the POD basis function ϕ_i on the coarse mesh

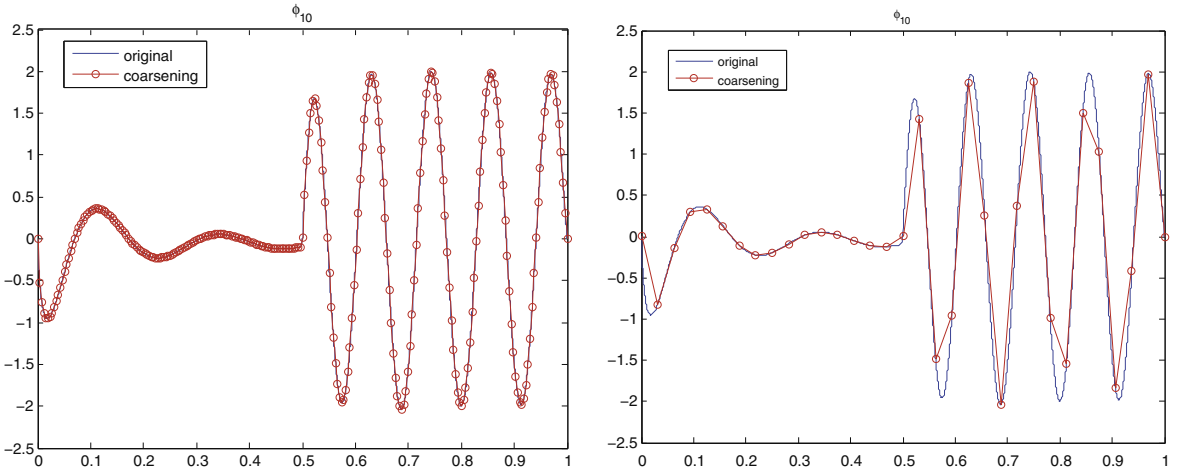

Fig. 2. 1 D Burgers equation with $\nu = 10^{-3}$. Coarsening and interpolation of ϕ_{10} for two rates of coarsening: 32 (left) and 256 (right). Notice that the structure of the POD basis is preserved remarkably well.

was carried out by simply keeping the values of ϕ_i corresponding to the nodes of the coarse mesh. The results in Fig. 2 indicate that the coarsening and interpolation preserve the structure of the POD basis remarkably well.

The *coarse two-level algorithm* (21) with $r = 10$ and $R_c = 16$, 32 produces the temporal evolution of POD coefficients in Fig. 3. Comparing these results with those for the one-level algorithm (19), we conclude that the coarse two-level algorithm (21) and the one-level algorithm (19) yield results that are qualitatively similar (see Table 1).

The *hybrid two-level algorithm* (20) with $r = 10$ and $R_c = 16$, 32 yields the temporal evolution of POD coefficients in Fig. 3. Comparing these results with those for the one-level algorithm (19), we conclude that the hybrid two-level algorithm (20) yields an improved accuracy while keeping the CPU time at a modest level (see Table 1).

To measure the efficiency of our two-level algorithms, we define a *speed-up factor*:

$$S_f \equiv \frac{\text{CPU time of one-level algorithm(19)}}{\text{CPU time of two-level algorithm(21)or(20)}}. \tag{24}$$

We present the speed-up factors along with relative errors for both two-level algorithms in Table 1. The relative error of the one-level algorithm is 2.55×10^{-2}. The data in Table 1 clearly shows that both two-level algorithms significantly decrease the CPU time (by up to an *order of magnitude*) while maintaining similar accuracy as the more expensive one-level algorithm. We also emphasize that, as expected, the two-level algorithms decrease significantly the CPU time of the underlying DNS, by a factor of more than 300.

Thus, for this test problem, we conclude that the two-level algorithms achieve the same order of accuracy as the standard one-level algorithm, while decreasing the CPU time by more than *an order of magnitude*. We note that, for this test case, the coarse two-level algorithm (21) performs better than the hybrid two-level algorithm (20).

3.2. 2 D Flow past a circular cylinder at Re = 200

POD has been implemented successfully in the low-dimensional modeling of laminar cylinder wakes [83,22,84]. Deane et al. [10] modeled the dynamics of the flow past a cylinder with an eight-dimensional Galerkin model. Ma and Karniadakis [22] generalized the POD Galerkin models to predict the 3 D transition initiated by the Mode A instability [85] at $Re \approx 180$.

In this section, we investigate the two-level algorithm using the numerical simulation of 2 D flow past a circular cylinder at $Re = 200$. We simulate 2 D flow past a cylinder using a parallel CFD solver on a 4-processor platform and record the snapshot data of the flow field. The details of the numerical algorithm employed are provided in Section 3.3 and Appendix A. The reader is referred to [37,86,87] for validation and verification results of the 2 D version of the CFD solver.

The qualitative behavior of all the models tested is represented by the time evolution of the POD coefficients $a_1(\cdot)$ and $a_3(\cdot)$ and the 2 D projections of the limit cycle on the phase planes – (a_1, a_i), $i = 2, 3$ presented in Figs. 4 and 5, respectively. We note that the other POD coefficients have a similar behavior; for clarity of exposition, we included only $a_1(\cdot)$ and $a_3(\cdot)$.

After we collect the DNS data, we construct a structured quadratic finite element mesh with nodes coinciding with the nodes used in the original DNS finite volume discretization. For the last period, we generate 40 snapshots, then use the *method of snapshots* [4] to generate the POD basis. Deane et al. [10] observed that 20 snapshots are sufficient for the construction of the first eight eigenfunctions at $Re = 100$–200. In general, numerical studies [88] suggest that the first r POD modes, where r is even, resolve the first $r/2$ temporal harmonics and require a $2r$ number of snapshots for convergence.

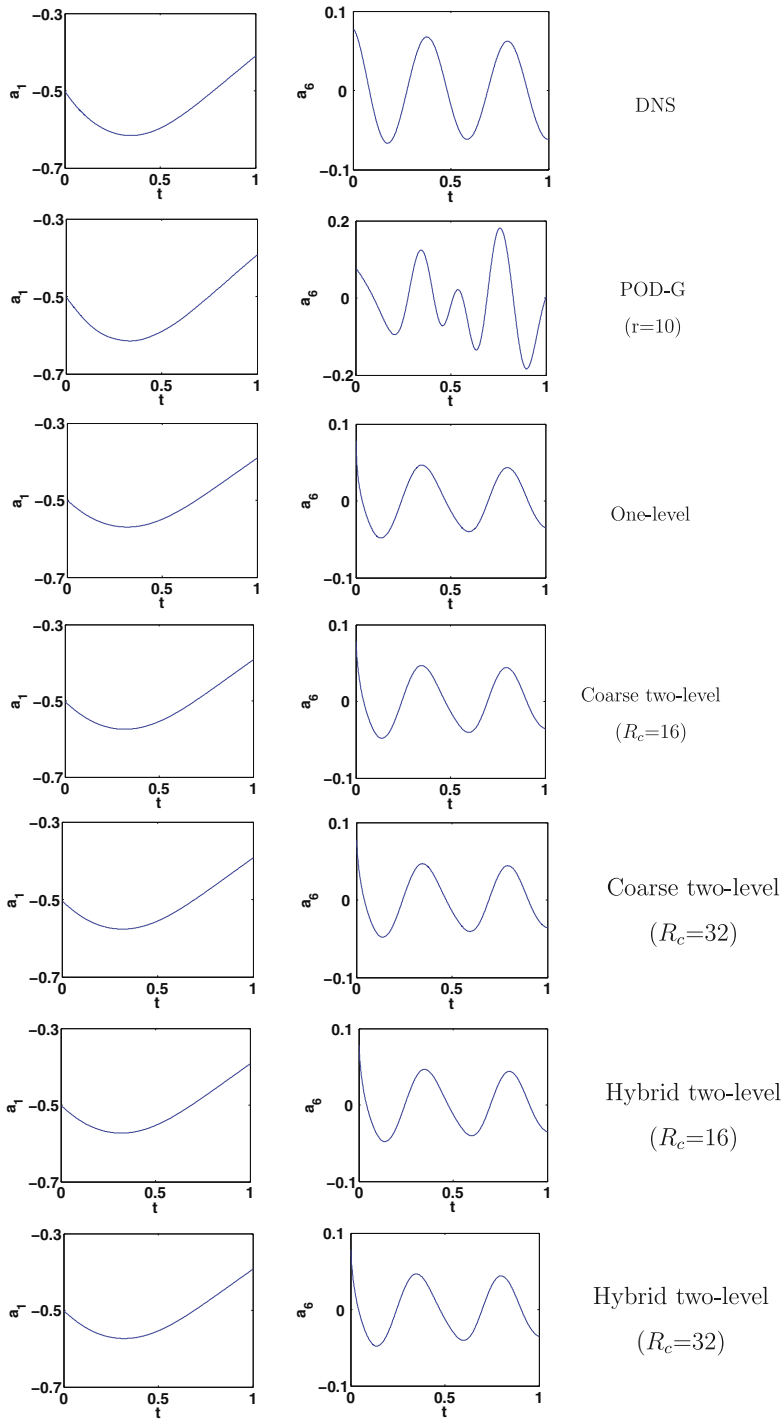

Fig. 3. 1 D Burgers equation with $\nu = 10^{-3}$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_6(\cdot)$ over [0,1]. Notice that POD-G with $r = 10$ performs poorly compared to DNS. The one-level algorithm for the nonlinear closure model improves the results of POD-G with $r = 10$, but the computational cost is high. The two-level algorithms keep the same accuracy, but decrease the computational cost by more than *one order of magnitude*.

This POD basis is then used in all POD reduced-order models for the 2D flow past a circular cylinder. For all these POD models, the time discretization was performed using the explicit Euler method with $\Delta t = 1.4 \times 10^{-3}$.

For comparison purposes, Figs. 4 and 5 include the time evolution of the POD coefficients obtained by projecting the DNS onto each POD mode and the 2 D projection of the limit cycle on the phase planes, respectively.

Table 1
1 D Burgers equation with $\nu = 10^{-3}$. Speed-up factors S_f and relative errors for different coarsening factors R_c. Note that the relative errors for the two-level algorithms are comparable to the error for the one-level algorithm (2.55×10^{-2}). The two-level algorithms, however, significantly decrease the CPU time (by up to an order of magnitude) of the one-level algorithm (858 s).

R_c	Coarse two-level		Hybrid two-level	
	S_f	Error	S_f	Error
2	1.92	2.43×10^{-2}	1.87	2.43×10^{-2}
4	3.68	2.36×10^{-2}	3.59	2.36×10^{-2}
8	6.76	2.31×10^{-2}	6.60	2.33×10^{-2}
16	12.62	2.27×10^{-2}	12.08	2.30×10^{-2}
32	21.45	2.20×10^{-2}	19.50	2.27×10^{-2}

The POD-G (13) with r small ($r = 4$) yields poor results, although it is computationally efficient (its CPU time is 246 s). Indeed, the temporal evolution of POD coefficients and the 2 D projections of the limit cycle on the phase planes in Figs. 4 and 5, respectively, are inaccurate. The amplitude of the temporal evolution of the POD coefficient $a_3(\cdot)$ for POD-G with $r = 4$ is *more than eight times larger* than that for the DNS. The relative error of POD-G with $r = 4$ is 9.57×10^{-2}. In fact, the solution diverges to a spurious limit cycle. Furthermore, the 2 D projections of the limit cycle on the phase planes – (a_1, a_i), $i = 2, 3$ for POD-G with $r = 4$ are significantly different from the projected DNS. Thus, although the first 4 POD modes capture 98.74% of the system's energy, these results clearly indicate the need for closure modeling.

We will use the Smagorinsky model (14) with an artificial viscosity coefficient $C = 1.9 \times 10^{-3}$ as a closure model. Although the Smagorinsky model targets 3 D turbulent flows, for this test problem it plays the role of numerical stabilization. Nevertheless, by using formulas (15) and (16), we can determine the values for the corresponding Smagorinsky constant and filter radius: $C_S = 0.1845$ and $\delta = 0.1987$, respectively. For the numerical discretization of the resulting nonlinear POD–ROM (17), we will use three different strategies: (i) the one-level algorithm (19); (ii) the hybrid two-level algorithm (20); and (iii) the coarse two-level algorithm (21).

The one-level algorithm (19) with $r = 4$ yields the temporal evolution of POD coefficients and the 2 D projections of the limit cycle on the phase planes – (a_1, a_i), $i = 2, 3$ in Figs. 4 and 5. Comparing these results with those for the DNS, it is clear that the one-level algorithm yields relatively accurate results, although it uses only a few ($r = 4$) POD basis functions. Indeed, the one-level algorithm (19) produces results that are much closer to those for DNS than the results for POD-G with $r = 4$. The relative error of the one-level algorithm is 9.0×10^{-3}, whereas that of POD-G with $r = 4$ is 9.57×10^{-2}. This, however, comes at a high computational cost. The CPU time is 246 s for POD-G and is 4.01×10^4 s for the one-level algorithm.

To implement our two-level algorithms (20) and (21), we again used the simplest coarsening and interpolation procedures. The coarsening of the mesh was effected by skipping nodes in the radial and azimuthal directions. For example, when the coarsening factor $R_c = 2$, we skip every other node in both directions; when $R_c = 4$, we skip over three nodes in both directions (and so on). To interpolate a POD basis function ϕ_i on the coarse mesh, we just keep the values of ϕ_i corresponding to the nodes of the coarse mesh. The effect of the mesh coarsening and interpolation is shown in Fig. 6 for two coarsening factors ($R_c = 2$ and $R_c = 4$). It is clear from Fig. 6 that, although the accuracy of the POD modes degrades as the mesh is coarsened, the *geometric structure is preserved*.

The *coarse two-level algorithm* (21) with $r = 4$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients and the 2 D projections of the limit cycle on the phase planes – (a_1, a_i), $i = 2, 3$ in Figs. 4 and 5. Comparing these results with those for the one-level algorithm (19), we conclude that the coarse two-level algorithm (21) and the one-level algorithm (19) yield results that are qualitatively similar. The coarse two-level algorithm, however, reduces that computational time by an order of magnitude (see Table 2).

The *hybrid two-level algorithm* (20) with $r = 4$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients and 2 D projections of the limit cycle on the phase planes – (a_1, a_i), $i = 2, 3$ in Figs. 4 and 5. Comparing these results with those for the one-level algorithm (19), we conclude that the hybrid two-level algorithm (20) yields an improved accuracy (a shorter initial transition in the temporal evolution of the POD coefficients $a_i(\cdot)$ and tighter 2D projections of the limit cycle on the phase planes) while keeping the CPU time at a modest level (see Table 2).

To measure the efficiency of our two-level algorithms, we present the speed-up factors S_f defined in (24) along with relative errors for both two-level algorithms in Table 2. Again, the relative error of the one-level algorithm is 9.0×10^{-3}. The data in Table 2 clearly shows that both two-level algorithms significantly decrease the CPU time (by up to an order of magnitude), while maintaining similar accuracy as the more expensive one-level algorithm.

Thus, for this test problem, we conclude that the two-level algorithms achieve the same order of accuracy as the standard one-level algorithm, while decreasing the CPU time by more than *an order of magnitude*. We also note that, in this case, the hybrid two-level algorithm (20) performs better than the coarse two-level algorithm (21).

We now compare the computational costs of the DNS and the POD–ROM options. This comparison, however, is challenging, since the discretizations used in the two approaches are completely different. Indeed, the spatial discretization used in the DNS was the finite volume method, whereas for POD–ROMs we used a finite element method. Furthermore, the time-

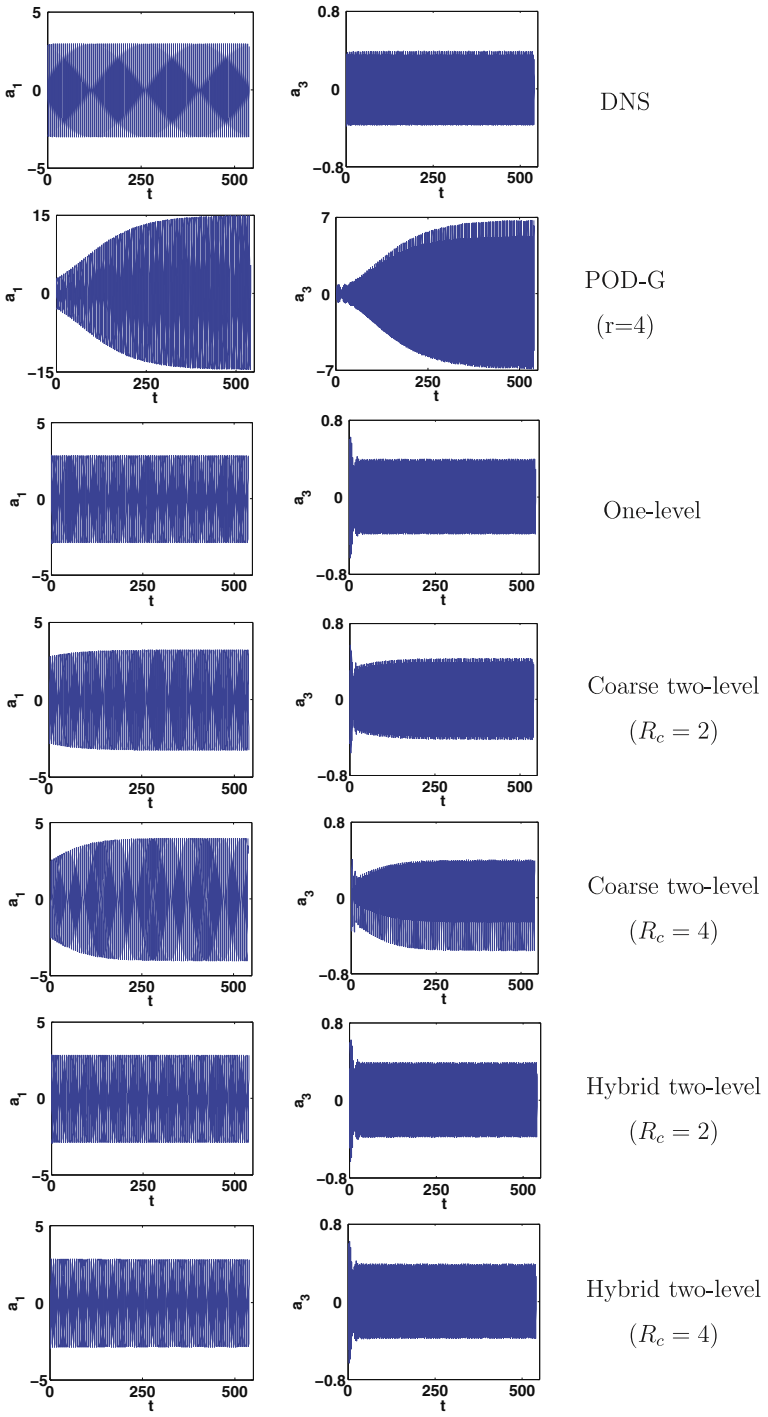

Fig. 4. 2 D flow past a cylinder at $Re = 200$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_3(\cdot)$ over 100 vortex shedding cycles. Notice that POD-G with $r = 4$ performs poorly compared to DNS. The one-level algorithm for the nonlinear closure model improves the results of POD-G with $r = 4$, but the computational cost is high. The two-level algorithms keep the same accuracy, but decrease the computational cost by more than *one order of magnitude*.

discretization used in DNS was second-order (Crank–Nicolson and Adams–Bashforth), whereas in POD–ROM we used a first-order time discretization (explicit Euler). The time-steps employed were also different: $\Delta t = 2 \times 10^{-3}$ in the DNS and $\Delta t = 1.4 \times 10^{-3}$ in the POD–ROMs. Finally, the DNS was performed on a parallel machine (on 4 processors), whereas the POD–ROMs were carried out on a single-processor machine.

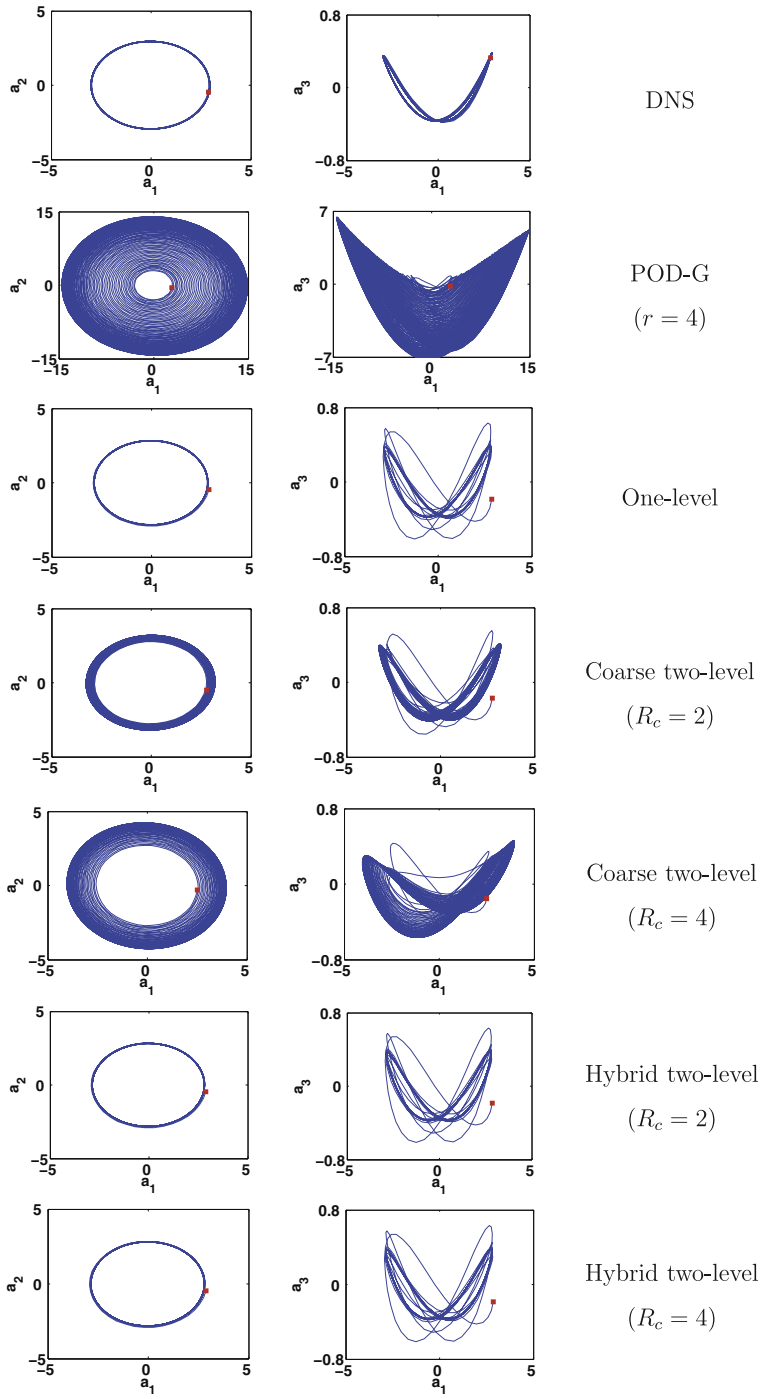

Fig. 5. 2 D flow past a cylinder at $Re = 200$. 2 D projections of the limit cycle on the phase planes (a_1, a_2) and (a_1, a_3) over 100 vortex shedding cycles. The red square denotes the initial position of $(a_1(0), a_i(0))$, $i = 2, 3$. Notice that POD-G with $r = 4$ performs poorly compared to DNS. The one-level algorithm for the nonlinear closure model improves the results of POD-G with $r = 4$, but the computational cost is high. The two-level algorithms keep the same accuracy, but decrease the computational cost by more than *one order of magnitude*.

Keeping in mind these significant differences between the two types of discretization employed, the two-level algorithm significantly reduced the computational time of the DNS: The CPU time for the DNS was 1.17×10^4 s, whereas for the POD–ROM with the two-level discretization it was 2.52×10^3 s.

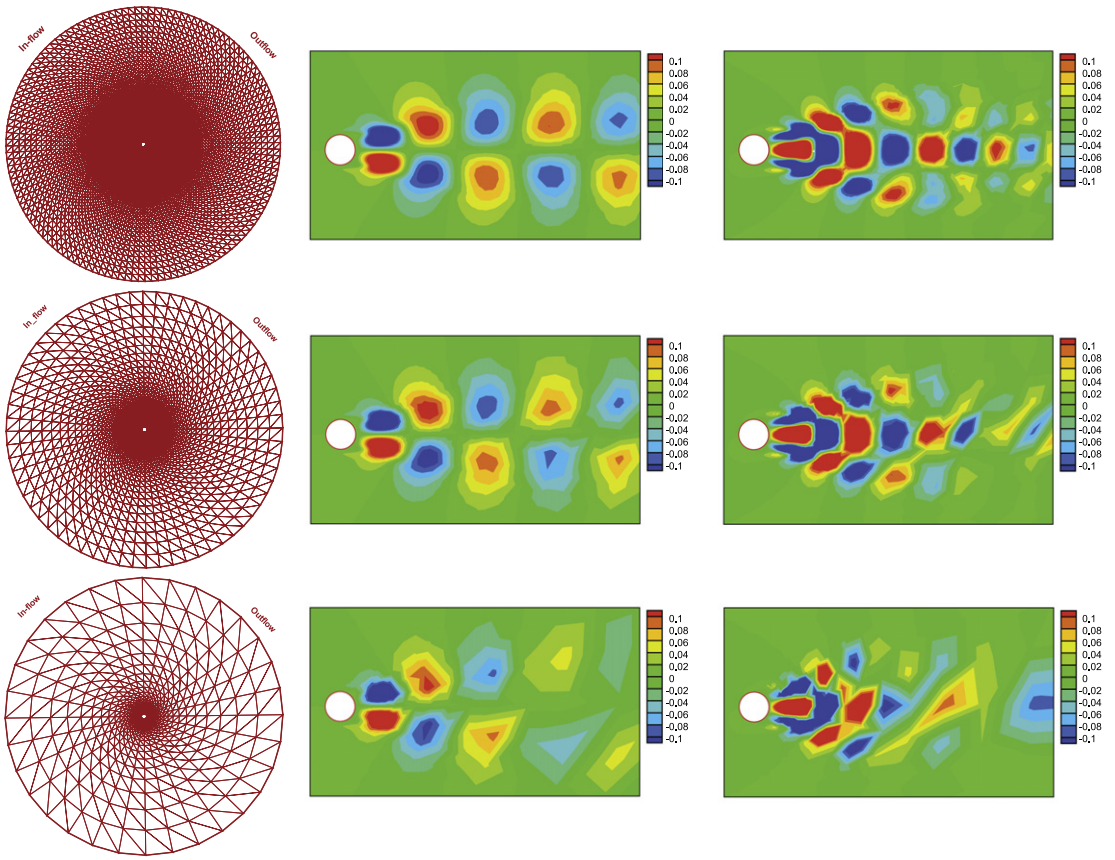

Fig. 6. 2 D flow past a cylinder at $Re = 200$: Mesh (left column), streamwise component of ϕ_1 (middle column), and streamwise component of ϕ_3 (right column). (For clarity, the POD modes are plotted only in a neighborhood of the cylinder.) Original fine mesh (top row), mesh coarsened by a factor of 2 (middle row), and mesh coarsened by a factor of 4 (bottom row). Notice that, although the accuracy of the POD modes degrades as the mesh is coarsened, the *geometric structure is preserved*.

Table 2
2 D flow past a cylinder at $Re = 200$. Speed-up factors S_f and relative errors for different coarsening factors R_c. Note that the relative errors for the two-level algorithms are comparable to the error for the one-level algorithm (9.0×10^{-3}). The two-level algorithms, however, significantly decrease the CPU time (by up to an order of magnitude) of the one-level algorithm (4.01×10^4 s).

R_c	Coarse two-level		Hybrid two-level	
	S_f	Error	S_f	Error
2	4.03	1.02×10^{-2}	3.97	9.0×10^{-3}
4	15.91	1.12×10^{-2}	14.58	9.0×10^{-3}

3.3. 3 D Flow past a circular cylinder at Re = 1000

In this section, we investigate the two-level algorithms in the numerical simulation of 3 D flow past a circular cylinder at $Re = 1000$. A parallel CFD solver is employed to generate the DNS data [37]. For details on numerical discretization, the reader is referred to Appendix A. The qualitative behavior of all reduced-order models investigated is represented by the time evolution of the POD coefficients $a_1(\cdot)$ and $a_4(\cdot)$ in Fig. 8. We note that the other POD coefficients have a similar behavior; for clarity of exposition, we included only $a_1(\cdot)$ and $a_4(\cdot)$.

We record snapshot data of the flow field to compute the POD basis functions. At $Re = 1000$, the wake is fully turbulent and the flow is not periodic anymore as observed in the lower Re regime. We notice that there is a spectrum of frequencies present in the flow field. This requires a large snapshot data set comprising many shedding cycles.

By using multiple processors, we reduce the load per processor and are able to record a large data set of the flow field. Each processor records 1000 snapshots of the velocity field (u_1, u_2, u_3) over 15 shedding cycles on its local grid points. Thus, parallel computing is a key enabling technology for model reduction in 3 D flows. Using the method of snapshots [4], we

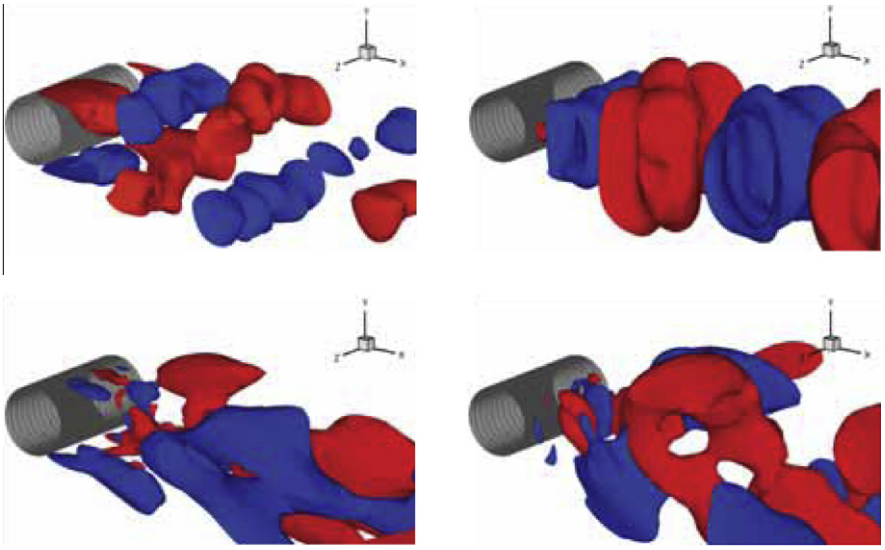

Fig. 7. 3 D flow past a cylinder at $Re = 1000$. First streamwise POD mode (top left), first normal POD mode (top right), third streamwise POD mode (bottom left), and third normal POD mode (bottom right).

construct a correlation matrix and compute the POD basis $\{\phi_1, \ldots, \phi_N\}$ using MPI-ALLREDUCE (SUM) operation [91]. We observe that the 3 D POD modes are not symmetric anymore due to presence of multiple frequencies in the dynamical system as shown in Fig. 7.

These POD modes are then interpolated onto a structured quadratic finite element mesh with nodes coinciding with the nodes used in the original DNS finite volume discretization. The POD basis is then used in all POD reduced-order models that we investigate next. For all these POD models, the time discretization was effected by using the explicit Euler method with $\Delta t = 7.5 \times 10^{-4}$. We note that POD–ROMs have also been extended to the pressure field [84,87]. The key advantage of pressure ROM is the computation of hydrodynamic forces on the structure. Akhtar et al. [87] developed a pressure-Poisson based ROM that requires the POD modes of the pressure fields in addition to those for the velocity field. Their model predicted the lift and drag forces on a cylinder at $Re = 100$. In the current study, however, we restrict ourselves to the ROM of the velocity field. Thus, lift and drag forces cannot be computed and compared with those from DNS.

For comparison purposes, we include the time evolution of the POD coefficients obtained by projecting the DNS onto each POD mode.

The POD-G (13) with r small ($r = 6$) is not appropriate, although it is computationally efficient (its CPU time is 486 s). Indeed, the temporal evolutions of POD coefficients in Fig. 8 are inaccurate. The amplitude of the temporal evolution of the POD coefficient $a_4(\cdot)$ for $r = 6$ is *nine times larger* than that for the DNS projection. The relative error of POD-G with $r = 6$ is 1.23×10^{-1}. Thus, although the first 6 POD modes capture 84% of the system's energy, these results clearly indicate the need for closure modeling.

We will use the Smagorinsky model (14) with an artificial viscosity coefficient $C = 4 \times 10^{-4}$ as a POD closure model for turbulent flows. By using formulas (15) and (16), we can determine the values for the corresponding Smagorinsky constant and filter radius: $C_S = 0.1426$ and $\delta = 0.1179$, respectively. For the numerical discretization of the resulting nonlinear POD–ROM (17), we will use three different strategies: (i) the one-level algorithm (19); (ii) the hybrid two-level algorithm (20); and (iii) the coarse two-level algorithm (21).

The one-level algorithm (19) with $r = 6$ yields the temporal evolution of POD coefficients in Fig. 8. Comparing these results with those for the DNS, it is clear that the one-level algorithm yields relatively accurate results although it uses only a few ($r = 6$) POD basis functions. Indeed, the one-level algorithm (19) produces results that are much closer to those for the DNS than the results for POD-G with $r = 6$. The relative error of the one-level algorithm is 4.46×10^{-2}, whereas that of POD-G with $r = 6$ is 1.23×10^{-1}. This, however, comes at a high computational cost. Indeed, the CPU time is 486 s for POD-G with $r = 6$ and is 5.32×10^5 s for the one-level algorithm.

We used the simplest coarsening and interpolation procedures. The coarsening of the mesh was effected by skipping nodes in the radial and azimuthal directions. For example, when the coarsening factor $R_c = 2$, we skip every other node in both directions; when $R_c = 4$, we skip over three nodes in both directions (and so on). For this test problem, we only coarsened the mesh in the x and y directions, since the mesh resolution in the z direction was already low. To interpolate a POD basis function ϕ_i on the coarse mesh, we just keep the values of ϕ_i corresponding to the nodes of the coarse mesh.

The *coarse two-level algorithm* (21) with $r = 6$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients in Fig. 8. Comparing these results with those for the one-level algorithm (19), we conclude that the coarse two-level algorithm (21) and

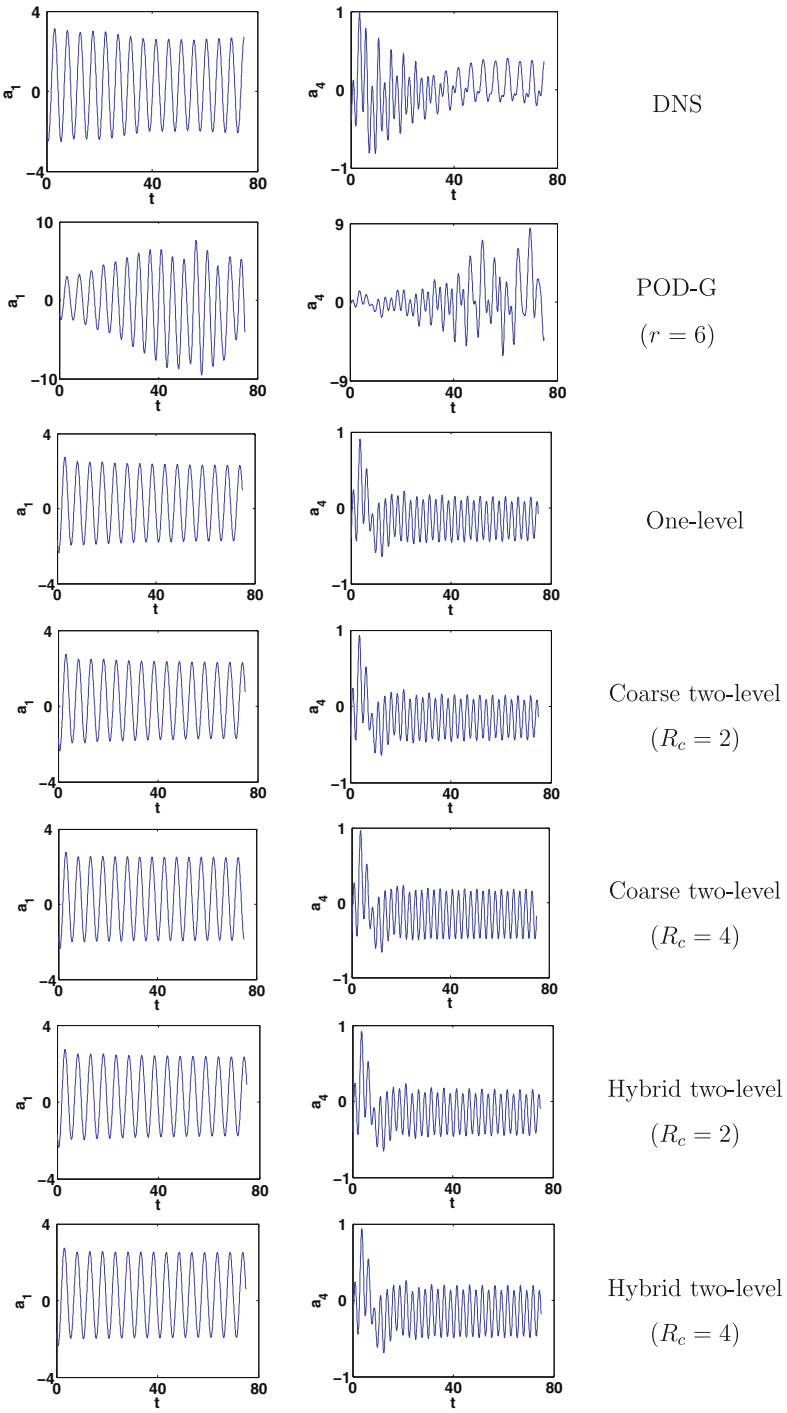

Fig. 8. 3 D flow past a cylinder at $Re = 1000$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_4(\cdot)$ over the time interval $[0, 75]$. Notice that POD-G with $r = 6$ performs poorly compared to DNS. The one-level algorithm for the closure model improves the results of POD-G with $r = 6$, but the computational cost is high. The two-level algorithms keep the same accuracy, but decrease the computational cost by more than *one order of magnitude*.

the one-level algorithm (19) yield results that are qualitatively similar. The coarse two-level algorithm, however, reduces that computational time by an order of magnitude (see Table 3).

The *hybrid two-level algorithm* (20) with $r = 6$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients in Fig. 8. Comparing these results with those for the one-level algorithm (19), we conclude that the hybrid two-level algorithm (20) yields an improved accuracy while keeping the CPU time at a modest level (see Table 3).

Table 3
3 D flow past a cylinder at $Re = 1000$. Speed-up factors S_f and relative errors for different coarsening factors R_c. Note the relative errors for the two-level algorithms are comparable to the error for the one-level algorithm (4.46×10^{-2}). The two-level algorithms, however, significantly decrease (by over an order of magnitude) the CPU time of the one-level algorithm (5.32×10^5 s).

R_c	Coarse two-level		Hybrid two-level	
	S_f	Error	S_f	Error
2	4.97	4.44×10^{-2}	5.22	4.52×10^{-2}
4	24.52	3.85×10^{-2}	24.18	4.73×10^{-2}

Although the main goal of this paper is to investigate whether, for a given closure model (the basic Smagorinsky model), the two-level algorithms represent an efficient, yet accurate alternative to the one-level algorithm, we note that the POD–ROM results are not always accurate when compared with DNS results. Indeed, we note that the time evolutions for the POD–ROMs' a_1 coefficients in Fig. 8 are close to those corresponding to the DNS. The time evolutions for the POD–ROMs' a_4 coefficients, however, are less accurate. We believe that these inaccuracies are due to the fact that the closure model employed is just the standard Smagorinsky model; more accurate closure models (such as the dynamic SGS model and the variational multiscale model) will probably yield improved results. We also emphasize that the plots in Fig. 8 represent instantaneous quantities (not averages), which are generally hard to capture accurately by turbulence closure models.

To measure the efficiency of our two-level algorithms, we present the speed-up factors S_f given in (24) along with relative errors for both two-level algorithms in Table 3. The relative error of the one-level algorithm is 4.46×10^{-2}. The data in Table 3 clearly shows that both two-level algorithms significantly decrease (by over an order of magnitude) the CPU time while maintaining similar accuracy as the more expensive one-level algorithm.

Thus, for this test problem, we conclude that the two-level algorithms achieve the same order of accuracy as the standard one-level algorithm, while decreasing the CPU time by more than *an order of magnitude*. We also notice that, in this case, the coarse two-level algorithm (21) performs better than the hybrid two-level algorithm (20).

As in the 2 D test case, we mention that a comparison between the computational cost of the DNS and that of the POD–ROMs used is challenging, since the discretizations used in the two approaches are completely different. Indeed, the spatial discretization used in the DNS was the finite volume method, whereas for POD–ROMs we used a finite element method. Furthermore, the time-discretization used in DNS was second-order (Crank–Nicolson and Adams–Bashforth), whereas in POD–ROM we used a first-order time discretization (explicit Euler). The time-steps employed were also different: $\Delta t = 2 \times 10^{-3}$ in the DNS and $\Delta t = 7.5 \times 10^{-4}$ in the POD–ROM. Finally, the DNS was performed on a parallel machine (on 16 processors), whereas the POD–ROMs were carried out on a single-processor machine.

Keeping in mind these significant differences between the two types of discretization employed, the two-level algorithm reduced the computational time of the DNS: The CPU time for the DNS was 3.58×10^4 s, whereas for the POD–ROM with the two-level discretization it was 2.17×10^4 s. We believe that by increasing the time-step, using a higher-order time-discretization and migrating the code to parallel machines, the computational cost of the two-level algorithm can be decreased even further. We plan to pursue these research directions in a future study.

4. Conclusions

In this paper, we proposed a two-level algorithm for the numerical discretization of *nonlinear* POD closure models. This two-level algorithm opens new research avenues in the investigation of physically accurate POD closure modeling strategies, similar to the state-of-the-art closure models used in LES of turbulent flows. The main idea behind the new two-level algorithm is the evaluation of the nonlinear POD closure model on a coarse mesh. This allows for an efficient computation of the nonlinear POD closure model at *each* time step. While reduced-order models will never fully replace high resolution simulation methods such as LES, there are a number of important applications where the construction of accurate reduced-order models is critical. This discretization methodology for nonlinear POD closure models creates more strategies for building these models.

We considered two versions of the two-level algorithm: (i) a hybrid two-level algorithm in which only the nonlinear term was evaluated on the coarse mesh; and (ii) a coarse two-level algorithm in which all the terms of the POD–ROM were evaluated on the coarse mesh. Both two-level algorithms were compared to a standard brute-force computational approach, which we denoted the one-level algorithm. We investigated the two-level algorithms in the numerical simulation of three test problems: (i) the one-dimensional Burgers equation with a small diffusion parameter $\nu = 10^{-3}$; (ii) the 2 D flow past a cylinder at Reynolds number $Re = 200$; and (iii) the 3 D flow past a cylinder at Reynolds number $Re = 1000$. For completeness, we also included results with POD Galerkin formulation (i.e., without any closure model) with a low number of POD modes, as well as a DNS projection of the evolution of the POD modes, which served as benchmark for our numerical simulations. We emphasize that the goal of this paper is to propose an efficient numerical discretization of a given nonlinear POD closure model, and *not* to advocate a specific POD closure model. Therefore, we just chose one of the standard closure models used in LES, the Smagorinsky model.

For all three numerical tests, we drew the same conclusions. The two-level algorithms achieved the same order of accuracy as the standard one-level algorithm, but they both decreased the computational cost of the latter by more than an *order of magnitude*. Thus, the two-level algorithms appear as promising approaches in the numerical discretization of nonlinear closure modeling strategies for POD–ROMs. The coarse two-level algorithm performed better than the hybrid two-level algorithm in the 1 D and 3 D tests, whereas the latter performed better in the 2 D test. The differences between the two two-level algorithms in terms of computational efficiency and numerical accuracy, however, were minimal. Since there are no major differences in terms of implementation, we conclude that both two-level algorithms should be considered in practical POD–ROM computations. We also note that, in general, the errors in both two-level algorithms seem to decrease as R_c increases. This behavior, however, is not always observed (e.g., the coarse two-level algorithm in Table 2 and the hybrid two-level algorithm in Table 3). This behavior is counterintuitive, since coarsening the mesh should result in an error increase. We do mention, however, that given the mesh sizes considered in our study (especially those for the 2 D and 3 D numerical tests), we cannot infer the true asymptotic behavior of the error with respect to the mesh size.

We plan to further investigate several research avenues. First, we plan to develop discretization strategies for reducing the computational cost of the two-level algorithm even further. To this end, we plan to develop more appropriate interpolation strategies, based on Galerkin projection rather than simply skipping points. We also plan to investigate more efficient time-discretization approaches, since in this report we used the forward Euler method, which only has first-order accuracy and is only conditionally stable. Higher-order time discretization strategies will provide significant further savings in computational time. Finally, we will take advantage of parallel computing in order to further decrease the computational time and, at the same time, increase the dimension (and thus physical accuracy) of the POD–ROM.

A second research direction we plan to pursue is to employ the new two-level algorithm in the numerical discretization of modern closure modeling strategies for POD–ROMs of coherent structures in turbulent flows. Thus, we will investigate how state-of-the-art LES closure modeling strategies, such as the *dynamic SGS model* and the *variational multiscale method*, can improve the physical accuracy of POD–ROMs of turbulent flows dominated by coherent structures, such as turbulent boundary layers. In particular, we will investigate different approaches for determining a length-scale δ and use a dynamic procedure to determine a closure model term C that varies in both time and space. We will also gradually increase the Reynolds number in the turbulent flows we will consider and assess the validity range of the POD–ROMs. To obtain the snapshot data for the POD–ROM in these high Reynolds number flows, we plan to employ modern numerical simulation strategies, such as LES. Equipped with these modern closure models, the POD–ROMs could then be tested in the numerical simulation of turbulent flows dominated by coherent structures for time intervals that are longer than those on which the POD snapshot matrix was computed.

A third research direction we will investigate is the extension of the two-level methodology to POD–ROMs of geophysical flows. Of course, the nonlinear closure models for these predominantly two-dimensional oceanic and atmospheric flows will be different from most of the LES closure models used in the numerical simulations of three-dimensional turbulent flows, where the concept of energy cascade plays a central role. We will develop POD–ROM closure models based on *approximate deconvolution* [67], which is a mathematical closure modeling strategy that does not rely on the concept of energy cascade.

Finally, we plan to employ the two-level methodology in other applications in which *nonlinear* closure models for POD is needed, such as optimal control, optimization, uncertainty quantification and data assimilation problems. Since the two-level method we propose has a general framework and is not restricted to turbulence closure models, its extension to other nonlinear POD closure models is straightforward.

Acknowledgements

We greatly appreciate the financial support of the Air Force Office of Scientific Research through Grant No. FA9550-08-1-0136. A significant part of the computations were carried out on SystemX at Virginia Tech's Advanced Research Computing center (http://www.arc.vt.edu). The allocation grant and support provided by the staff are gratefully acknowledged. We also thank the two anonymous referees for their comments and suggestions, which significantly improved this paper.

Appendix A

We briefly describe the numerical method used to generate the DNS data for the test problems in Sections 3.2 and 3.3. We use a parallel CFD algorithm to solve the incompressible NSE. Using a strong conservative form, the governing equations are written in curvilinear coordinates (ξ_1, ξ_2, ξ_3) as

$$\frac{\partial U_m}{\partial \xi_m} = 0, \tag{25}$$

$$\frac{\partial \left(J^{-1} u_i\right)}{\partial t} + \frac{\partial F_{im}}{\partial \xi_m} = 0, \tag{26}$$

and the flux is defined as

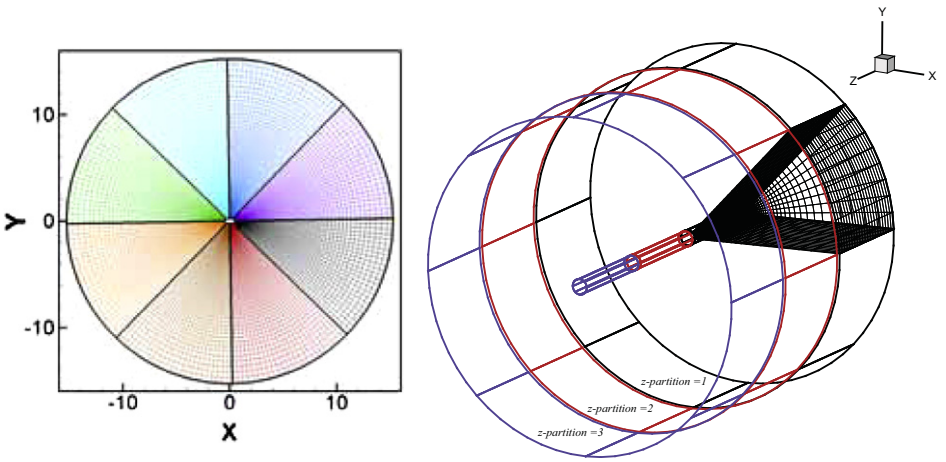

Fig. 9. 3 D flow past a cylinder at $Re = 1000$. A typical "O"-grid distributed over different processors: a 2 D layout of 192×256 grid distributed on an 8-processor platform (left) and a 3 D layout of $128 \times 128 \times 96$ "O"-grid distributed on a 24 (8×3) processor platform (right). The grid is plotted only for the region corresponding to processor "1". (Note that only every 4th grid point is plotted for clarity.)

$$F_{im} = U_m u_i + J^{-1} \frac{\partial \xi_m}{\partial x_i} p - \frac{1}{Re} G^{mn} \frac{\partial u_i}{\partial \xi_n}, \tag{27}$$

where $i, j = 1, 2, 3$, u_i represent the Cartesian velocity components, and p is the pressure. Eqs. (25) and (26) are non-dimensionalized using the diameter (D) of the cylinder as length scale and the freestream velocity (U_∞) as velocity scale. Thus, the Reynolds number is $Re = \frac{DU_\infty}{\nu}$, where $\nu = \frac{\mu}{\rho}$. Other quantities are defined as follows: $J^{-1} = det\left(\frac{\partial x_i}{\partial \xi_j}\right)$ is the inverse of the Jacobian or the volume of the cell; $U_m = J^{-1} \frac{\partial \xi_m}{\partial x_j} u_j$ is the volume flux (contravariant velocity multiplied by J^{-1}) normal to the surface of constant ξ_m; and $G^{mn} = J^{-1} \frac{\partial \xi_m}{\partial x_j} \frac{\partial \xi_n}{\partial x_j}$ is the "mesh skewness tensor".

We use a body conformal "O"-type grid to simulate the flow past a cylinder and employ curvilinear coordinates in an Eulerian reference frame. The governing equations are solved using a methodology similar to that employed by Zang et al. [97]. To implement the algorithm on a distributed-memory, message-passing parallel computer, we use a 2D domain decomposition technique such that each processor gets a "slice" of the grid, as shown in Fig. 9.

A non-staggered-grid layout is employed to solve the transformed NSE. The Cartesian velocity components (u_1, u_2, u_3) and pressure (p) are defined at the center of the control volume in the computational space and the volume fluxes (U_1, U_2, U_3) are defined at the midpoints of the corresponding faces. All spatial derivatives are approximated with second-order accurate central differences except for the convective terms, which are discretized by using a variation of QUICK [92]. Dirichlet and Neumann boundary conditions are used for the inflow and outflow boundary conditions, respectively. No-slip and no-penetration boundary conditions are prescribed on the cylinder surface.

A semi-implicit scheme is employed to advance the solution in time. The diagonal viscous terms are advanced implicitly using the second-order accurate Crank–Nicolson method, whereas all of the other terms are advanced using the second-order accurate Adams–Bashforth method.

In the present formulation, we apply a fractional-step method [93–97] to advance the solution in time. The fractional-step method splits the momentum equation into: (a) an advection–diffusion equation – momentum equation solved without the pressure term; and (b) a pressure Poisson equation – constructed by implicit coupling between the continuity equation and the pressure in the momentum equation, thus satisfying the mass conservation constraint.

We perform a DNS of the flow past a circular cylinder on a $144 \times 192 \times 16$ grid distributed over 16 processors in circumferential direction. Thus, the load per processor is $144 \times 12 \times 16$ grid points. In this simulation, the outer domain is $15D$ with a nondimensional spanwise length of $2D$. A CFL number based on the convection term in curvilinear coordinates is used as a guideline in choosing the time step. The simulations show that a stable time stepping is achieved for a CFL ≈ 0.2, which corresponds to a nondimensional time step size $\Delta t = 2 \times 10^{-3}$ for this grid.

Table 4
3 D flow past a cylinder at $Re = 1{,}000$. Validation of DNS results. Notice that the mean drag coefficient $\overline{C_D}$ and Strouhal number St match those from [89,90].

Data from	$\overline{C_D}$	St
Experiment [89]	1.0	0.21
3-D DNS [90]	1.02	0.202
3-D DNS (present)	1.11	0.205

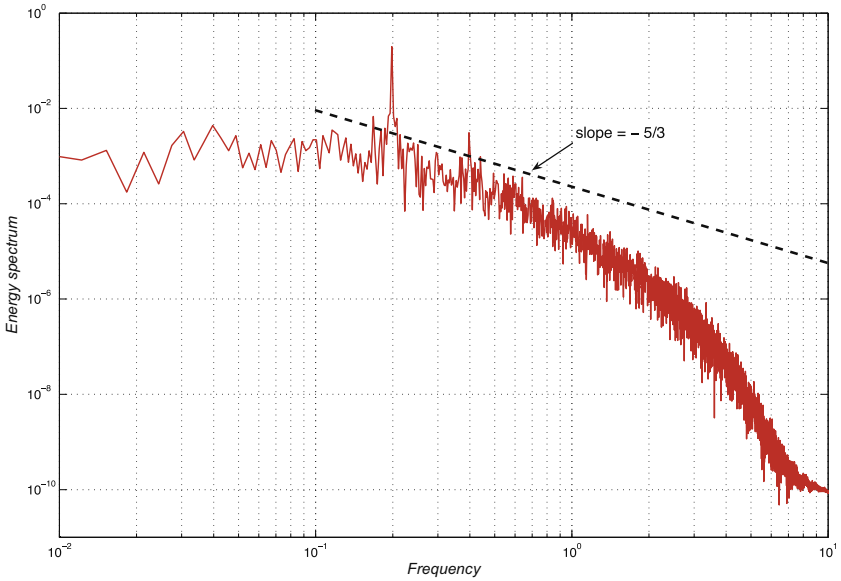

Fig. 10. 3 D flow past a cylinder at $Re = 1000$. Energy spectrum.

To validate our DNS results, we compute the mean drag coefficient $\overline{C_D}$ and the Strouhal number St at $Re = 1000$. In Table 4, these quantities are compared with those obtained by Norberg [89] and Evangelinos and Karniadakis [90].

We also discuss the pointwise statistics focusing on the structure as well as the wake. In Fig. 10, the 1 D power energy spectrum at $(x/D, y/D, z/D) = (1.0, 0.5, 1.0)$ shows a peak at the shedding frequency of 0.205. Moreover, it exhibits a $-\frac{5}{3}$ law in the inertial range, which extends about half a decade in wave number. For more details on discretization, validation, verification, and parallel implementation of the numerical scheme, the reader is referred to [37,87].

References

[1] K. Karhunen, Zur spektraltheorie stochastischer prozesse, Annales Academiae Scientarum Fennicae 37 (1946).
[2] M.M. Loève, Probability Theory, Van Nostrand, Princeton, NJ, 1955.
[3] J.L. Lumley, The structure of inhomogeneous turbulent flows, in: A.M. Yaglom, (Ed.), Atmospheric Turbulence and Radio Wave Propagation, 1967, pp. 166–178.
[4] L. Sirovich, Turbulence and the dynamics of coherent structures, part I: coherent structures, Quarterly of Applied Mathematics 45 (3) (1987) 561–571.
[5] L. Sirovich, J.D. Rodriguez, Coherent structures and chaos: a model problem, Physics Letters A 120 (5) (1987) 211–214.
[6] J.S. Peterson, The reduced basis method for incompressible viscous flow calculations, SIAM Journal on Scientific and Statistical Computing 10 (4) (1989) 777–786.
[7] C. Foias, O. Manley, L. Sirovich, Empirical and Stokes eigenfunctions and the far-dissipative turbulent spectrum, Physics of Fluids A: Fluid Dynamics 2 (1990) 464–467.
[8] J.L. Lumley, Order and disorder in turbulent flows, in: L. Sirovich (Ed.), New Perspectives in Turbulence, Springer-Verlag, 1991, pp. 105–122.
[9] A.E. Deane, L. Sirovich, A computational study of Rayleigh–Bénard convection. Part 1. Rayleigh-number scaling, Journal of Fluid Mechanics 222 (1991) 231–250.
[10] A.E. Deane, I.G. Kevrekidis, G.E. Karniadakis, S.A. Orszag, Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders, Physics of Fluids A: Fluid Dynamics 3 (10) (1991) 2337–2354.
[11] L. Sirovich, A.E. Deane, A computational study of Rayleigh–Bénard convection. Part 2. Dimension considerations, Journal of Fluid Mechanics 222 (1991) 251–265.
[12] M. Couplet, P. Sagaut, C. Basdevant, Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow, Journal of Fluid Mechanics 491 (2003) 275–284.
[13] A. Duggleby, K.S. Ball, M.R. Paul, P.F. Fischer, Dynamical eigenfunction decomposition of turbulent pipe flow, Journal of Turbulence 8 (2007) 1–24.
[14] N. Aubry, P. Holmes, J.L. Lumley, E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, Journal of Fluid Mechanics 192 (1988) 115–173.
[15] N. Aubry, W.Y. Lian, E.S. Titi, Preserving symmetries in the proper orthogonal decomposition, SIAM Journal on Scientific Computing 14 (1993) 483–505.
[16] D. Rempfer, H.F. Fasel, Dynamics of three-dimensional coherent structures in a flat-plate boundary layer, Journal of Fluid Mechanics 275 (1994) 257–283.
[17] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, Cambridge, 1996.
[18] B. Podvin, J. Lumley, A low-dimensional approach for the minimal flow unit, Journal of Fluid Mechanics 362 (1998) 121–155.
[19] W. Cazemier, R. Verstappen, A.E.P. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Physics of Fluids 10 (1998) 1685–1699.
[20] B. Podvin, On the adequacy of the ten-dimensional model for the wall layer, Physics of Fluids 13 (1) (2001) 210–224.
[21] B. Podvin, P. Le Quéré, Low-order models for the flow in a differentially heated cavity, Physics of Fluids 13 (11) (2001) 3204–3214.
[22] X. Ma, G. Karniadakis, A low-dimensional model for simulating three-dimensional cylinder flow, Journal of Fluid Mechanics 458 (2002) 181–190.
[23] B.R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, Journal of Fluid Mechanics 497 (2003) 335–363.
[24] S. Sirisup, G.E. Karniadakis, A spectral viscosity method for correcting the long-term behavior of POD models, Journal of Computational Physics 194 (1) (2004) 92–116.

[25] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD–Galerkin system for fluid flow modelling, Journal of Computational Physics 207 (1) (2005) 192–220.
[26] M. Buffoni, S. Camarri, A. Iollo, M.V. Salvetti, Low-dimensional modelling of a confined three-dimensional wake flow, Journal of Fluid Mechanics 569 (2006) 141–150.
[27] N.M. Arifin, M.S.M. Noorani, A. Kilicman, Modelling of Marangoni convection using proper orthogonal decomposition, Nonlinear Dynamics 48 (2007) 331–337.
[28] M. Ilak, C.W. Rowley, Modeling of transitional channel flow using balanced proper orthogonal decomposition, Physics of Fluids 20 (034103) (2008).
[29] Z. Lou, J. Chen, I.M. Navon, X. Yang, Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier–Stokes equations, SIAM Journal on Numerical Analysis 47 (1) (2008) 1–19.
[30] M. Bergmann, C.H. Bruneau, A. Iollo, Enablers for robust POD models, Journal of Computational Physics 228 (2) (2009) 516–538.
[31] A. Hay, J. Borggaard, D. Pelletier, Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition, Journal of Fluid Mechanics 629 (2009) 41–72.
[32] J.A. Burns, B.B. King, A reduced basis approach to the design of low-order feedback controllers for nonlinear continuous systems, Journal of Vibration and Control 4 (1998) 297–323.
[33] S.S. Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition, International Journal for Numerical Methods in Fluids 34 (5) (2000) 425–448.
[34] K. Ito, J.D. Schroeter, Reduced order feedback synthesis for viscous incompressible flows, Mathematical and Computer Modelling 33 (2001) 173–192.
[35] K. Kunisch, S. Volkwein, L. Xie, HJB–POD-based feedback design for the optimal control of evolution problems, SIAM Journal on Applied Dynamical Systems 3 (4) (2004) 701–722.
[36] M. Bergmann, L. Cordier, J.-P. Brancher, Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model, Physics of Fluids 17 (097101) (2005).
[37] I. Akhtar, Parallel simulations, reduced-order modeling, and feedback control of vortex shedding using fluidic actuators, PhD Thesis, Virginia Tech, Blacksburg, VA, 2008.
[38] S. Bagheri, L. Brandt, D. Henningson, Input-output analysis, model reduction and control of the flat-plate boundary layer, Journal of Fluid Mechanics 620 (2009) 263–298.
[39] A. Barbagallo, D. Sipp, P. Schmid, Closed-loop control of an open cavity flow using reduced-order models, Journal of Fluid Mechanics 641 (2009) 1–50.
[40] S. Ahuja, C. Rowley, Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators, Journal of Fluid Mechanics 645 (2010) 447–478.
[41] J. Borggaard, M. Stoyanov, L. Zietsman, Linear feedback control of a von Kármán street by cylinder rotation, in: Proceedings of the 2010 American Control Conference, FrB06.3, 2010, pp. 5674–5681.
[42] D.T. Crommelin, A.J. Majda, Strategies for model reduction: comparing different optimal bases, Journal of the Atmospheric Sciences 61 (2004) 2206–2217.
[43] Z. Luo, J. Zhu, R. Wang, I.M. Navon, Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical Pacific Ocean reduced gravity model, Computer Methods in Applied Mechanics and Engineering 196 (41–44) (2007) 4184–4195.
[44] D.N. Daescu, I.M. Navon, A dual-weighted approach to order reduction in 4DVAR data assimilation, Monthly Weather Review 136 (3) (2008) 1026–1041.
[45] F. Fang, C.C. Pain, I.M. Navon, M.D. Piggott, G.J. Gorman, P.E. Farrell, P.A. Allison, A.J.H. Goddard, A POD reduced-order 4 D-Var adaptive mesh ocean modelling approach, International Journal for Numerical Methods in Fluids 60 (7) (2009) 709–732.
[46] A. Doostan, R.G. Ghanem, J. Red-Horse, Stochastic model reduction for chaos representations, Computer Methods in Applied Mechanics and Engineering 196 (2007) 3951–3966.
[47] J. Burkardt, M.D. Gunzburger, C. Webster, Reduced order modeling of some nonlinear stochastic partial differential equations, International Journal of Numerical Analysis and Modeling 4 (3–4) (2007) 368–391.
[48] E. Arian, M. Fahl, E.W. Sachs, Managing POD models by optimization methods, in: Proceedings of the 41st IEEE Conference on Decision and Control, Number ThP04-6, 2002, pp. 3300–3305.
[49] T. Bui-Thanh, M. Damodaran, K. Willcox, Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, AIAA Journal 42 (8) (2004) 1505–1516.
[50] K. Carlberg, C. Farhat, A compact proper orthogonal decomposition basis for optimization-oriented reduced-order models, in: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2008, AIAA Paper 2008-5964.
[51] J.D. Rodriguez, L. Sirovich, Low-dimensional dynamics for the complex Ginzburg–Landau equation, Physica D (1990) 77–86.
[52] M.A. Cardoso, L.J. Durlofsky, Linearized reduced-order models for subsurface flow simulation, Journal of Computational Physics 229 (2010) 681–700.
[53] S. Acharjee, N. Zabaras, A proper orthogonal decomposition approach to microstructure model reduction in Rodrigues space with applications to optimal control of microstructure-sensitive properties, Acta Materialia 51 (2003) 5627–5646.
[54] M. Bozkurttas, R. Mittal, H. Dong, G.V. Lauder, P. Madden, Low-dimensional models and performance scaling of a highly deformable fish pectoral fin, Journal of Fluid Mechanics 631 (2009) 311–342.
[55] G. Kerschen, J.C. Golinval, A model updating strategy of non-linear vibrating structures, International Journal for Numerical Methods in Engineering 60 (2004) 2147–2164.
[56] S. Lall, J.E. Marsden, S. Glavaški, A subspace approach to balanced truncation for model reduction of nonlinear control systems, International Journal of Robust and Nonlinear Control 12 (6) (2002) 519–535.
[57] A. Sawant, A. Acharya, Model reduction via parametrized invariant manifolds: some examples. <arXiv:math-ph/0412022>, 2004.
[58] B.F. Feeny, On the proper orthogonal modes and normal modes of continuous vibration systems, Journal of Vibration and Acoustics 124 (1) (2002) 157–160.
[59] V. Lenaerts, G. Kerschen, J.-C. Golinval, Identification of a continuous structure with a geometrical non-linearity part II: proper orthogonal decomposition, Journal of Sound and Vibration 262 (2003) 907–919.
[60] B.F. Feeny, A complex orthogonal decomposition for wave motion analysis, Journal of Sound and Vibration 310 (2008) 77–90.
[61] P.B. Goncalves, F.M.A. Silva, Z.J.G.N. Del Prado, Low-dimensional models for the nonlinear vibration analysis of cylindrical shells based on a perturbation procedure and proper orthogonal decomposition, Smart Material and Structures 315 (3) (2008) 641–663.
[62] S. Volkwein, A. Hepberger, Impedance identification by POD model reduction techniques, Automatisierungstechnik 8 (2008) 437–446.
[63] F. Lanata, A. Del Grosso, Damage detection and localization for continuous static monitoring of structures using a proper orthogonal decomposition of signals, Smart Material and Structures 15 (2006) 811–1829.
[64] J. Lawrie, W. Hearnie, Using modified proper orthogonal decomposition (MPOD) for reducing ecosystem models, The ANZIAM Journal 48 (2007) 461–473.
[65] Y. Zhang, M.A. Henson, Y.G. Kevrekedis, Nonlinear order reduction of discretized cell population models, in: Proceedings of the American Control Conference, 2003, pp. 2383–2388.
[66] P. Sagaut, Large Eddy Simulation for Incompressible Flows, third ed., Scientific Computation, Springer-Verlag, Berlin, 2006.
[67] L.C. Berselli, T. Iliescu, W.J. Layton, Mathematics of Large Eddy Simulation of Turbulent Flows, Scientific Computation, Springer-Verlag, Berlin, 2006.
[68] D.E. Goldstein, O.V. Vasilyev, Stochastic coherent adaptive large eddy simulation method, Physics of Fluids 16 (2004) 2497.
[69] M. Farge, K. Schneider, N. Kevlahan, Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis, Physics of Fluids 11 (1999) 2187–2201.

[70] M. Farge, K. Schneider, Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets, Flow, Turbulence and Combustion 66 (4) (2001) 393–426.
[71] G. De Stefano, O.V. Vasilyev, Stochastic coherent adaptive large eddy simulation of forced isotropic turbulence, Journal of Fluid Mechanics 646 (1) (2010) 453–470.
[72] M. Morzynski, W. Stankiewicz, B.R. Noack, F. Thiele, R. King, G. Tadmor, Generalized mean-field model for flow control using a continuous mode interpolation, in: Proceedings of the Third AIAA Flow Control Conference, 2006, AIAA Paper 2006-3488.
[73] A. Iollo, S. Lanteri, J.A. Desideri, Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations, Theoretical and Computational Fluid Dynamics 13 (6) (2000) 377–396.
[74] M.A. Grepl, Y. Maday, N.C. Nguyen, A.T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM: Mathematical Modelling and Numerical Analysis 41 (3) (2007) 575–605.
[75] H. Park, L. Sirovich, Turbulent thermal convection in a finite domain. II. Numerical results, Physics of Fluids A 2 (9) (1990) 1659–1668.
[76] B.R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzynski, P. Comte, G. Tadmor, A finite-time thermodynamics of unsteady fluid flows, Journal of Non-Equilibrium Thermodynamics 33 (2) (2008) 103–148.
[77] S.B. Pope, Turbulent Flows, Cambridge University Press, Cambridge, 2000.
[78] B.R. Noack, P. Papas, P.A. Monkewitz, Low-dimensional Galerkin model of a laminar shear-layer, Technical Report, École Polytechnique Fédérale de Lausanne, Number 2002-01, 2002.
[79] J. Borggaard, A. Duggleby, A. Hay, T. Iliescu, Z. Wang, Reduced-order modeling of turbulent flows, in: Proceedings of MTNS 2008, 2008.
[80] J.S. Smagorinsky, General circulation experiments with the primitive equations, Monthly Weather Review 91 (1963) 99–164.
[81] K. Kunisch, S. Volkwein, Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition, Journal of Optimization Theory and Applications 102 (2) (1999) 345–371.
[82] J. Borggaard, T. Iliescu, Z. Wang, Artificial viscosity proper orthogonal decomposition, Mathematical and Computer Modelling, in press, doi:10.1016/j.mcm.2010.08.015.
[83] A.E. Deane, C. Mavriplis, Low-dimensional description of the dynamics in separated flow past thick airfoils, AIAA Journal 6 (1994) 1222–1234.
[84] B.R. Noack, K. Afanasiev, M. Morzynski, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, Journal of Fluid Mechanics 497 (2003) 335–363.
[85] C.H.K. Williamson, Vortex dynamics in the cylinder wake, Annual Review of Fluid Mechanics 28 (1996) 477–539.
[86] I. Akhtar, O.A. Marzouk, A.H. Nayfeh, A van der Pol – Duffing oscillator model of hydrodynamic forces on canonical structures, Journal of Computational and Nonlinear Dynamics 4 (4) (2009) 041006.
[87] I. Akhtar, A.H. Nayfeh, C.J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows: A numerical study of vortex shedding, Theoretical and Computational Fluid Dynamics 23 (3) (2009) 213–237.
[88] B.R. Noack, P. Papas, P.A. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, Journal of Fluid Mechanics 523 (2005) 339–365.
[89] C. Norberg, An experimental investigation of the flow around a circular cylinder: influence of aspect ratio, Journal of Fluid Mechanics 258 (1994) 287–316.
[90] C. Evangelinos, G.E. Karniadakis, Dynamics and flow structures in the turbulent wake of rigid and flexible cylinders subject to vortex-induced vibrations, Journal of Fluid Mechanics 400 (1999) 91–124.
[91] I. Akhtar, J. Borggaard, A. Hay, Shape sensitivity analysis in flow models using a finite-difference approach, Mathematical Problems in Engineering (2010).
[92] B.P. Leonard, A stable and accurate convective modeling procedure based on quadratic upstream interpolation, Computational Methods in Applied Mechanical Engineering 19 (1979) 59–98.
[93] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, Journal of Computational Physics 2 (1) (1967) 12–26.
[94] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes, Journal of Computational Physics 59 (1985) 308–323.
[95] S.A. Orszag, L.C. Kells, Transition to turbulence in plane Poiseuille and plane Couette flow, Journal of Fluid Mechanics 96 (1) (1985) 159–205.
[96] C.L. Street, M.Y. Hussaini, A numerical solution of the appearance of chaos in finite length Taylor–Couette flow, Applied Numerical Mathematics 6 (1991) 123–139.
[97] Y. Zang, R.L. Street, J.R. Koseff, A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates, Journal of Computational Physics 114 (1994) 18–33.