Comput. Methods Appl. Mech. Engrg. 237–240 (2012) 10–26

Contents lists available at SciVerse ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison

Zhu Wang [a], Imran Akhtar [b], Jeff Borggaard [a], Traian Iliescu [a,*]

[a] *Department of Mathematics, Virginia Tech, Blacksburg, VA 24061-0123, USA*
[b] *Department of Mechanical Engineering, NUST College of Electrical & Mechanical Engineering, National University of Sciences & Technology, Islamabad, Pakistan*

ARTICLE INFO

Article history:
Received 8 December 2011
Received in revised form 13 April 2012
Accepted 20 April 2012
Available online 3 May 2012

Keywords:
Proper orthogonal decomposition
Reduced-order modeling
Turbulence
Large eddy simulation
Dynamic subgrid-scale model
Variational multiscale model

ABSTRACT

This paper puts forth two new closure models for the proper orthogonal decomposition reduced-order modeling of structurally dominated turbulent flows: the dynamic subgrid-scale model and the variational multiscale model. These models, which are considered state-of-the-art in large eddy simulation, together with the mixing length and the Smagorinsky closure models, are tested in the numerical simulation of 3D turbulent flow past a circular cylinder at $Re = 1000$. Five criteria are used to judge the performance of the proper orthogonal decomposition reduced-order models: the kinetic energy spectrum, the mean velocity, the Reynolds stresses, the root mean square values of the velocity fluctuations, and the time evolution of the POD coefficients. All the numerical results are benchmarked against a direct numerical simulation. Based on these numerical results, we conclude that the dynamic subgrid-scale and the variational multiscale models are the most accurate.

1. Introduction

Reduced-order models (*ROMs*) of structurally dominated turbulent flows are central to many applications in science and engineering, such as fluid flow control (see for example, [1–10]) and data assimilation of atmospheric and oceanic flows [11–13]. Both computational efficiency and physical accuracy are needed for the success of these ROMs in practical applications. Striking a balance between efficiency and accuracy in ROMs of turbulent flows is, of course, challenging. Indeed, it is clear that retaining fewer modes leads to more efficient ROMs. Preserving the physical accuracy of the resulting ROM, however, becomes challenging, since the modes that are not included in the ROM representation of the underlying turbulent flow need to be modeled. Thus, the ROM must contain a *closure model* to represent the effect of the discarded POD modes. The quest of balancing the computational efficiency and physical accuracy represents one of the main challenges in ROMs for turbulent flows.

One of the most successful ROM strategies for structurally dominated turbulent flows has been the *proper orthogonal decomposition* (*POD*) [14,15]. POD starts with data from an accurate numerical simulation (or physical experiment), extracts the most energetic modes in the system, and utilizes a Galerkin procedure that yields a ROM of the underlying turbulent flow. The first *proper orthogonal decomposition reduced-order model (POD-ROM)* for the turbulent boundary layer was proposed in [16]. This model truncated the POD basis and used an eddy viscosity-based closure model. This POD-ROM yielded good qualitative results, considering the coarseness of the approximation. The criterion used to assess the accuracy of the model was the intermittency of bursting events in the turbulent boundary layer. This POD-ROM was further investigated numerically in two subsequent papers [17,18]. The model reproduced the qualitative physics of the turbulent boundary layer well. Furthermore, by adding new POD modes to the model, the accuracy of the model was increased.

Since the eddy viscosity POD closure model put forth in [16], several alternative closure modeling strategies have been used, such as calibration [19–21] and the use of the H^1 norm instead of the standard L^2 norm in the derivation of the POD basis [22–24]. For a survey of these alternative approaches, the reader is referred to [25] (also see [26]). The focus of this report is eddy viscosity closure models that aim to improve the physical accuracy of the ROM for general flow settings. These settings will, in general, be different from those used in the computation of the underlying POD basis (e.g., longer time intervals and different Reynolds numbers). Similar goals are shared by the promising finite-time thermodynamics formalism put forth in [27–29]. The alternative POD closure

* Corresponding author.
E-mail addresses: wangzhu@vt.edu (Z. Wang), imran.akhtar@ceme.nust.edu.pk (I. Akhtar), jborggaard@vt.edu (J. Borggaard), iliescu@vt.edu (T. Iliescu).
URLs: http://www.math.vt.edu/people/wangzhu (Z. Wang), http://www.math.vt.edu/people/akhtar (I. Akhtar), http://www.math.vt.edu/people/borggajt (J. Borggaard), http://www.math.vt.edu/people/iliescu (T. Iliescu).

0045-7825/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cma.2012.04.015

modeling strategies mentioned above (i.e., calibration and H^1 norm POD basis selection), however, optimize the ROM with respect to the same flow setting used in the computation of the POD basis. Thus, as shown in [21], their application to general flow settings is beset with difficulties.

Despite their initial success, POD-ROMs have generally been limited to laminar flows and relatively few reports on closure modeling strategies for turbulent flows have appeared in the literature [16–18,21,27,30–41]. This is in stark contrast to the amount of work done in traditional turbulence modeling, such as *large eddy simulation* (*LES*), where literally hundreds of closure models have been proposed and investigated (see for example, [42]) over the same time period. This disparity in closure modeling between POD reduced-order modeling and classical turbulence modeling seems even more dramatic considering that the concept of an energy cascade, which is a fundamental modeling principle in LES, is also valid in a POD setting. Indeed, the validity of the extension of the energy cascade concept to the POD setting was studied numerically in [43], where the authors have investigated the energy transfer among POD modes in a non-homogeneous computational setting. By monitoring the triad interactions due to the nonlinear term in the Navier–Stokes equations, they concluded that the transfer of energy among the POD modes is similar to the transfer of energy among Fourier modes. Specifically, they found that there is a net forward energy transfer from low index POD modes to higher index POD modes and that this transfer of energy is local in nature (that is, energy is mainly transferred among POD modes whose indices are close to one another). This study (see also [35]) clearly suggests that LES ideas based on the energy cascade concept could also be used in devising POD-ROMs.

One of the main reasons for the scarcity of closure models for POD-ROMs of turbulent flows is the impractical cost of implementing standard LES closure models employed in a POD-ROM setting. Indeed, most of the computational cost of a POD-ROM lies in assembling the vectors, matrices and tensors of the ROM. This, however, is hardly a problem for POD-ROM, since the vectors, matrices and tensors are assembled only *once*, at the beginning of the POD-ROM simulation, and reused at every time step. Standard (nonlinear) LES closure models, however, introduce new vectors and matrices that need to be recomputed at every time step. Thus, a straightforward numerical discretization of such closure models comes at a huge computational cost, rendering the resulting POD-ROMs impractical.

In the past few years, a number of strategies have been introduced to treat nonlinear terms in POD-ROMs. These include interpolatory methods such as the empirical interpolation method [44], the discrete empirical interpolation method [45,46], the closely related group finite element approach [49], the gappy POD [47,48] and a novel two-level discretization method [50]. The latter approach is best suited for this study since it does not constrain the nonlinear term to lie within a predefined set. This approach is based on a two-level discretization of the vectors, matrices and tensors of the POD-ROM, in which all the terms are computed on the fine grid, except for the nonlinear closure model terms, which are computed on a coarser grid. In [50], numerical simulations of a turbulent flow past a 3D cylinder at $Re = 1000$ with a standard LES closure model [51] demonstrated that the new two-level discretization is both computationally efficient and physically accurate. Indeed, the new two-level algorithm decreased the CPU time of the standard one-level algorithm by more than an order of magnitude, without compromising the physical accuracy.

In this report, we use the two-level algorithm proposed in [50] to discretize two new POD-ROMs, inspired from state-of-the-art LES closure modeling strategies: the *dynamic subgrid-scale* (*DS*) model [52–54] and the *variational multiscale* (*VMS*) model [55]. We also consider the standard mixing-length closure model proposed in [16] and the Smagorinsky model proposed in [35,38,50], both being standard LES closure models. All four POD-ROMs are tested in the numerical simulation of a 3D turbulent flow past a circular cylinder at $Re = 1000$. Five criteria are used to judge the performance of the POD-ROMs: the kinetic energy spectrum, the mean velocity, the Reynolds stresses, the root mean square values of the velocity fluctuations, and the time evolution of the POD coefficients. All the numerical results are benchmarked against a *direct numerical simulation* (*DNS*).

The rest of the paper is organized as follows. The general methodology used in the development of POD-ROMs is presented in Section 2. The four POD closure models are described in Section 3 and are investigated numerically in Section 4. Finally, conclusions and several research directions currently pursued by our group are provided in Section 5.

2. POD reduced-order modeling

We now present the general approach used in the development of POD-ROMs. We start by briefly describing the POD methodology; for more details, the reader is referred to [14,15]. To this end, we consider the numerical solution of the incompressible *Navier–Stokes equations* (*NSE*):

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

where $\mathbf{u}$ is the velocity, p is the pressure and Re is 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,

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

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

$$\mathbf{X}^{k-1} = \text{span}\{\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_{k-1}\}, \quad 2 \leqslant k \leqslant N, \quad \text{in} \quad \mathcal{H}, \tag{3}$$

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

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

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 a centering trajectory), e.g., $\mathbf{y}(\mathbf{x},t) = \mathbf{u}(\mathbf{x},t) - \mathbf{U}(\mathbf{x})$ [14]. Note that each POD basis vector $\boldsymbol{\varphi}_k$ 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 POD-ROM of the flow is constructed from the POD basis by writing

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

where $\mathbf{U}(\mathbf{x})$ is the centering trajectory, $\{\boldsymbol{\varphi}_j\}_{j=1}^r$ 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. We now replace the velocity $\mathbf{u}$ with $\mathbf{u}_r$ in the NSE (1), then project the resulting equations onto the subspace $\mathbf{X}^r$. Using the boundary conditions and the fact that all modes are solenoidal, one obtains the *POD Galerkin reduced-order model* (*POD-G-ROM*):

$$\left(\frac{\partial \mathbf{u}_r}{\partial t}, \boldsymbol{\phi}\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \boldsymbol{\phi}) + \left(\frac{2}{Re}\mathbb{D}(\mathbf{u}_r), \nabla\boldsymbol{\phi}\right) = 0 \quad \forall \boldsymbol{\phi} \in \mathbf{X}^r, \tag{7}$$

where $\mathbb{D}(\mathbf{u}_r) := (\nabla\mathbf{u}_r + (\nabla\mathbf{u}_r)^T)/2$ is the deformation tensor of $\mathbf{u}_r$. We note that, since the computational domain that we consider is large enough, the pressure terms in (7) can be neglected (for details, see [37,56]). The POD-G-ROM (7) yields the following autonomous dynamical system for the vector of time coefficients, $\mathbf{a}(t)$:

$$\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A}\mathbf{a} + \mathbf{a}^T\mathbf{B}\mathbf{a}, \tag{8}$$

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:

$$a_j(0) = \langle \boldsymbol{\varphi}_j, \mathbf{u}(\cdot,0) - \mathbf{U}(\cdot)\rangle_{\mathcal{H}}, \quad j = 1,\ldots,r. \tag{9}$$

The finite dimensional system (8) can be written componentwise as follows: For all $k = 1,\ldots,r$,

$$\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km}a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{kmn}a_n(t)a_m(t), \tag{10}$$

where

$$b_k = -(\boldsymbol{\varphi}_k, \mathbf{U}\cdot\nabla\mathbf{U}) - \frac{2}{Re}\left(\nabla\boldsymbol{\varphi}_k, \frac{\nabla\mathbf{U} + \nabla\mathbf{U}^T}{2}\right), \tag{11}$$

$$A_{km} = -(\boldsymbol{\varphi}_k, \mathbf{U}\cdot\nabla\boldsymbol{\varphi}_m) - (\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_m\cdot\nabla\mathbf{U}) - \frac{2}{Re}\left(\nabla\boldsymbol{\varphi}_k, \frac{\nabla\boldsymbol{\varphi}_m + \nabla\boldsymbol{\varphi}_m^T}{2}\right), \tag{12}$$

$$B_{kmn} = -(\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{\varphi}_n). \tag{13}$$

3. POD closure models

In this section, we present the four POD closure models investigated numerically in Section 4. To this end, we start by describing the filtering operation and the spatial lengthscale δ used in the POD closure models. Both are needed in order to define meaningful LES-inspired POD closure models.

3.1. POD filter

In LES, the filter is the central tool used to obtain simplified mathematical models that are computationally tractable. The filtering operation is effected by convolution of flow variables with a rapidly decaying *spatial* filter g_δ, where δ is the radius of the spatial filter. In POD, however, there is no explicit spatial filter used. Thus, in order to develop LES-type POD closure models, a POD filter needs to be introduced. Given the hierarchical nature of the POD basis, a natural such filter appears to be the Galerkin projection. For all $\mathbf{u} \in \mathcal{H}$, the Galerkin projection $\bar{\mathbf{u}} \in \mathbf{X}^r$ is the solution of the following equation:

$$(\mathbf{u} - \bar{\mathbf{u}}, \boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\varphi} \in \mathbf{X}^r. \tag{14}$$

The Galerkin projection defined in (14) will be the filter used in all POD closure models studied in this report.

3.2. POD lengthscale

Next, we introduce the lengthscale δ used in the POD closure models. We emphasize that this choice is one of the fundamental issues in making a connection with LES. Indeed, we need such a lengthscale (δ) in order to define dimensionally sound POD models of LES flavor. To simplify the notation, in this section we utilize subscripts for the three spatial components: x_1, x_2, x_3 and u_1, u_2, u_3. In the other sections, we use the standard notation x, y, z and u, v, w.

To derive the lengthscale δ, we use dimensional analysis. A dimensionally sound lengthscale $l_>$ for a turbulent pipe flow was defined in [16]. In fact, this lengthscale was only defined implicitly, through the turbulent eddy viscosity $v_T := u_> l_>$. Indeed, Eq. (22) in [16] reads

$$v_T := u_> l_> = \frac{\int_0^{X_2}\langle u_{i>}u_{i>}\rangle dx_2}{\left(X_2 \int_0^{X_2}\langle u_{i>,j}u_{i>,j}\rangle dx_2\right)^{1/2}}, \tag{15}$$

where repeated indices denote summation, the subscript $_>$ denotes unresolved POD modes,

$$\langle f\rangle = \frac{1}{L_1L_3}\int_0^{L_1}\int_0^{L_3} f(\mathbf{x},t)dx_1dx_3 \tag{16}$$

denotes the spatial average of f in the homogeneous directions (here x_1 and x_3), and L_1, L_3 and X_2 are the streamwise, spanwise, and wall-normal dimensions of the computational domain, respectively. Note that the authors in [16] only considered the wall region, not the entire pipe flow. In (15), the following notation was used: $u_{i>} = \sum_{j=r+1}^{N} a_j^i\varphi_j, u_{i>}u_{i>} = \sum_{i=1}^{3} u_{i>}u_{i>}$, and $u_{i>,j} = \frac{\partial u_{i>}}{\partial x_j}$. Note that a quick dimensional analysis shows that the quantity defined in (15) has the units of a viscosity. Indeed,

$$[v_T] = \frac{\frac{m}{s}\frac{m}{s}m}{\left[m\left(\frac{1}{s}\frac{1}{s}m\right)\right]^{1/2}} = \frac{\frac{m^3}{s^2}}{\frac{m}{s}} = \frac{m^2}{s}. \tag{17}$$

In Appendix B of [16], the authors have further simplified (15) and expressed v_T in terms of the first neglected POD modes:

$$v_T := u_> l_> = \frac{\sum_{(\mathbf{k},n)}\lambda_{\mathbf{k}}^{(n)}}{\left(X_2L_1L_3\sum_{(\mathbf{k},n)}\lambda_{\mathbf{k}}^{(n)}\left(\int_0^{X_2} D\Phi_{i_{\mathbf{k}}}^{(n)} D\Phi_{i_{\mathbf{k}}}^{(n)*}dx_2 - k_1^2 - k_3^2\right)\right)^{1/2}}, \tag{18}$$

where the triplets $(\mathbf{k}, n)$ are the first neglected POD modes.

In Eq. (9.90) of [14], the authors define another dimensionally sound turbulent viscosity

$$v_T := u_> l_> = \frac{1}{X_2}\int_0^{X_2}\frac{\langle u_{i>}u_{i>}\rangle}{\langle u_{i>,j}u_{i>,j}\rangle^{1/2}}dx_2. \tag{19}$$

A quick dimensional analysis shows that the quantity defined in (19) also has the units of a viscosity.

We can use the definition of v_T in (15) and dimensional analysis to define the following lengthscale $l_>$:

$$l_> := \frac{\int_0^{X_2}\langle u_{i>}u_{i>}\rangle dx_2}{X_2\int_0^{X_2}\langle u_{i>,j}u_{i>,j}\rangle dx_2}. \tag{20}$$

A quick dimensional analysis reveals that the quantity defined in (20) is dimensionally sound, i.e., has the units of a lengthscale. Similarly, we can use the definition of v_T in (19) and dimensional analysis to define the following alternative lengthscale $l_>$:

$$l_> := \left(\frac{1}{X_2}\int_0^{X_2}\frac{\langle u_{i>}u_{i>}\rangle}{\langle u_{i>,j}u_{i>,j}\rangle}dx_2\right)^{1/2}. \tag{21}$$

Again, a quick dimensional analysis shows that the quantity defined in (21) is also dimensionally sound, i.e., has the units of a lengthscale. We note that one could also use the two definitions of v_T in (15) and (19) to define two dimensionally sound velocity scales

$u_>$. Since these velocity scales are not used in this report, we do not carry out their derivation.

In the 3D flow past a cylinder example that we consider in Section 4, both (20) and (21) are valid candidates for the definition of the lengthscale δ. The only modification we need to make (due to our computational domain) is to replace the horizontal averaging by spanwise averaging and take double integrals in the remaining directions. Respectively, we have

$$\delta := \left(\frac{\int_0^{L_1} \int_0^{L_2} \langle u_{i>} u_{i>} \rangle dx_1 dx_2}{\int_0^{L_1} \int_0^{L_2} \langle u_{i>,j} u_{i>,j} \rangle dx_1 dx_2} \right)^{1/2} \tag{22}$$

and

$$\delta := \left(\frac{1}{L_1 L_2} \int_0^{L_1} \int_0^{L_2} \frac{\langle u_{i>} u_{i>} \rangle}{\langle u_{i>,j} u_{i>,j} \rangle} dx_1 dx_2 \right)^{1/2}. \tag{23}$$

3.3. POD closure models

We are now ready to present the four POD closure models that are investigated numerically in Section 4.

The POD-G-ROM (7) has been successfully used for laminar flows. For structurally dominated turbulent flows, however, the POD-G-ROM simply fails (see for example, [50]). The reason is that the effect of the discarded POD modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_N\}$ needs to be included in the model. For turbulent flows, the most natural way to tackle this POD *closure problem* is by using the *eddy viscosity* (*EV*) concept, which states that the role of the discarded modes is to extract energy from the system. The concept of energy cascade, which is well established in a Fourier setting, has been recently confirmed in a POD setting in the numerical investigations in [43]. Thus, using LES inspired EV closure models in POD-ROM is a natural step.

In this section, we propose two new POD closure models: the dynamic subgrid-scale model and the variational multiscale model. We emphasize that, although these models were announced in [57], this study represents their *first* careful derivation and thorough numerical investigation. We also numerically test the mixing-length [16] and Smagorinsky [35,38,50] POD closure models.

Since all four POD closure models are of EV type, we first present a general EV POD-ROM framework. Then, for each closure model, we specify its implementation in this general framework. The general EV POD-ROM framework can be written as:

$$\dot{\mathbf{a}} = \left(\mathbf{b} + \widetilde{\mathbf{b}}(\mathbf{a}) \right) + \left(\mathbf{A} + \widetilde{\mathbf{A}}(\mathbf{a}) \right) \mathbf{a} + \mathbf{a}^T \mathbf{B} \mathbf{a}, \tag{24}$$

which is just a slight modification of the POD-G-ROM (8). The new terms in (24) (the vector $\widetilde{\mathbf{b}}(\mathbf{a})$ and the matrix $\widetilde{\mathbf{A}}(\mathbf{a})$) correspond to the numerical discretization of the POD closure model. In componentwise form, Eq. (24) can be written as

$$\dot{a}_k(t) = \left(b_k + \widetilde{b}_k(\mathbf{a}) \right) + \sum_{m=1}^{r} \left(A_{km} + \widetilde{A}_{km}(\mathbf{a}) \right) a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t), \tag{25}$$

where b_k, A_{km}, and B_{kmn} are the same as those in Eq. (8) and $\widetilde{b}_k(\mathbf{a})$ and $\widetilde{A}_{km}(\mathbf{a})$ depend on the specific closure model used.

3.3.1. The mixing-length POD reduced-order model (ML-POD-ROM)

The first POD closure model was the *mixing-length* model proposed in [16]. This closure model is of EV type and amounts to increasing the viscosity coefficient ν by

$$\nu_{ML} = \alpha \nu_T = \alpha U_{ML} L_{ML}, \tag{26}$$

where U_{ML} and L_{ML} are characteristic velocity and length scales for the unresolved scales, and α is an $\mathcal{O}(1)$ nondimensional parameter that characterizes the energy being dissipated. Using the EV ansatz in (26), the *mixing-length POD reduced-order model* (*ML-POD-ROM*) has the form (24), where

$$\widetilde{b}_k(\mathbf{a}) = -\nu_{ML} \left(\nabla \boldsymbol{\varphi}_k, \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \tag{27}$$

$$\widetilde{A}_{km}(\mathbf{a}) = -\nu_{ML} \left(\nabla \boldsymbol{\varphi}_k, \frac{\nabla \boldsymbol{\varphi}_m + \nabla \boldsymbol{\varphi}_m^T}{2} \right). \tag{28}$$

The parameter α is expected to vary in a real turbulent flow, and different values of α may result in different dynamics of the flow [14,16–18]. There are also different ways to define ν_T in (26): relation (15) was used in [16], whereas relation (19) was used in [14]. We also mention that several other authors have used the ML-POD-ROM (26) (see for example, [50,57]). Improvements to the mixing-length model (26) in which the EV coefficient is mode dependent were proposed in [30,32,58].

3.3.2. The Smagorinsky POD reduced-order model (S-POD-ROM)

A potential improvement over the simplistic mixing-length hypothesis is to replace the constant ν_{ML} in (27) and (28) (which is computed only once, at the beginning of the simulation) with a variable turbulent viscosity (which is recomputed at every time step), such as that proposed in [51]. This yields a POD closure model in which the viscosity coefficient is increased by

$$\nu_S := (C_S \delta)^2 \| \mathbb{D}(\mathbf{u}_r) \|, \tag{29}$$

where C_S is the Smagorinsky constant, δ is the lengthscale defined in Section 3.2 and $\| \mathbb{D}(\mathbf{u}_r) \|$ is the Frobenius norm of the deformation tensor $\mathbb{D}(\mathbf{u}_r)$. Using the EV ansatz in (29), the *Smagorinsky POD reduced-order model* (*S-POD-ROM*) has the form (24), where

$$\widetilde{b}_k(\mathbf{a}) = -2 (C_S \delta)^2 \left(\nabla \boldsymbol{\varphi}_k, \| \mathbb{D}(\mathbf{u}_r) \| \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \tag{30}$$

$$\widetilde{A}_{km}(\mathbf{a}) = -2 (C_S \delta)^2 \left(\nabla \boldsymbol{\varphi}_k, \| \mathbb{D}(\mathbf{u}_r) \| \frac{\nabla \boldsymbol{\varphi}_m + \nabla \boldsymbol{\varphi}_m^T}{2} \right). \tag{31}$$

The S-POD-ROM (30) and (31) was proposed in [57] (see also [35]) and was used in the reduced-order modeling of structurally dominated 3D turbulent flows in [38,50]. Its advantage over the ML-POD-ROM (27) and (28) is obvious: the latter utilizes a constant EV coefficient at every time step, whereas the former recomputes the EV coefficient (which depends on $\| \mathbb{D}(\mathbf{u}_r) \|$) at every time step. To address the significant computational burden posed by the recalculation of the Smagorinsky EV coefficient at every time step, a novel two-level discretization algorithm proposed in [50] is employed in Section 4.

3.3.3. The variational multiscale POD reduced-order model (VMS-POD-ROM)

The VMS method, a state-of-the-art LES closure modeling strategy, was introduced in [55,59,60]. The VMS method is based on the principle of *locality* of energy transfer, i.e., it uses the ansatz that energy is transferred mainly between neighboring scales. In [43], the transfer of energy among POD modes for turbulent flow past a backward-facing step (a non-homogeneous separated flow) was investigated numerically. In their report, it was shown that the Fourier-decomposition based concepts of energy cascade and locality of energy transfer are also valid in the POD context (see Figs. 3 and 4 in [43]). Thus, VMS closure models represent a natural choice for POD-ROM.

To develop the VMS POD closure model, we start by decomposing the finite set of POD modes $\mathbf{X}^r$ into the direct sum of *large resolved* POD modes $\mathbf{X}_L^r$ and *small resolved* POD modes $\mathbf{X}_S^r$:

$$\mathbf{X}^r = \mathbf{X}^r_L \oplus \mathbf{X}^r_S, \quad \text{where} \tag{32}$$

$$\mathbf{X}^r_L := \operatorname{span}\left\{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{r_L}\right\} \quad \text{and} \tag{33}$$

$$\mathbf{X}^r_S := \operatorname{span}\left\{\boldsymbol{\varphi}_{r_L+1}, \boldsymbol{\varphi}_{r_L+2}, \ldots, \boldsymbol{\varphi}_{r}\right\}. \tag{34}$$

Accordingly, we decompose $\mathbf{u}_r$ into two components: $\mathbf{u}_r^L$ representing the *large resolved* scales, and $\mathbf{u}_r^S$ representing the *small resolved* scales:

$$\mathbf{u}_r = \mathbf{u}_r^L + \mathbf{u}_r^S, \tag{35}$$

where

$$\mathbf{u}_r^L = \mathbf{U} + \sum_{j=1}^{r_L} a_j \boldsymbol{\varphi}_j, \tag{36}$$

$$\mathbf{u}_r^S = \sum_{j=r_L+1}^{r} a_j \boldsymbol{\varphi}_j. \tag{37}$$

The two components $\mathbf{u}_r^L$ and $\mathbf{u}_r^S$ represent the projections of $\mathbf{u}_r$ onto the two spaces $\mathbf{X}^r_L$ and $\mathbf{X}^r_S$, respectively. The general POD-ROM framework (24) can now be separated into two equations - one for $\mathbf{a}^L$ (the vector of POD coefficients of $\mathbf{u}_r^L$) and one for $\mathbf{a}^S$ (the vector of POD coefficients of $\mathbf{u}_r^S$). The *variational multiscale POD reduced-order model* (VMS-POD-ROM) applies an eddy viscosity term to the small resolved scales only, following the principle of locality of energy transfer. The VMS-POD-ROM reads:

$$\begin{bmatrix} \dot{\mathbf{a}}^L \\ \dot{\mathbf{a}}^S \end{bmatrix} = \begin{bmatrix} \mathbf{b}^L \\ \mathbf{b}^S \end{bmatrix} + \mathbf{A}^r \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix} + \begin{bmatrix} \mathbf{A}^L & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^S + \widetilde{\mathbf{A}}^S(\mathbf{a}^S) \end{bmatrix} \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix} + \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix}^T \mathbf{B} \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix}. \tag{38}$$

The finite dimensional system (38) can be written componentwise as follows:

$$\dot{a}_k^L(t) = b_k^L + \sum_{m=1}^{r} A^r_{km} a_m(t) + \sum_{j=1}^{r_L} A^L_{kj} a_j(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t), \quad \forall k = 1, \ldots, r_L, \tag{39}$$

$$\dot{a}_k^S(t) = b_k^S + \sum_{m=1}^{r} A^r_{km} a_m(t) + \sum_{j=r_L+1}^{r} \left(A^S_{kj} + \widetilde{A}^S_{kj}\right) a_j(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t) \quad \forall k = r_L + 1, \ldots, r, \tag{40}$$

where

$$b_k^L = -(\boldsymbol{\varphi}_k, \mathbf{U}\cdot\nabla\mathbf{U}) - \frac{2}{Re}\left(\nabla\boldsymbol{\varphi}_k, \frac{\nabla\mathbf{U} + \nabla\mathbf{U}^T}{2}\right), \tag{41}$$

$$A^r_{km} = -(\boldsymbol{\varphi}_k, \mathbf{U}\cdot\nabla\boldsymbol{\varphi}_m) - (\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_m\cdot\nabla\mathbf{U}), \tag{42}$$

$$A^L_{kj} = -\frac{2}{Re}\left(\nabla\boldsymbol{\varphi}_k, \frac{\nabla\boldsymbol{\varphi}_j + \nabla\boldsymbol{\varphi}_j^T}{2}\right), \tag{43}$$

$$B_{kmn} = -(\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{\varphi}_n), \tag{44}$$

$$b_k^S = -(\boldsymbol{\varphi}_k, \mathbf{U}\cdot\nabla\mathbf{U}), \tag{45}$$

$$A^S_{kj} = -\frac{2}{Re}\left(\nabla\boldsymbol{\varphi}_k, \frac{\nabla\boldsymbol{\varphi}_j + \nabla\boldsymbol{\varphi}_j^T}{2}\right), \tag{46}$$

$$\widetilde{A}^S_{kj}(\mathbf{a}) = -2(C_S\delta)^2\left(\nabla\boldsymbol{\varphi}_k, \|\mathbb{D}(\mathbf{u}_r^S + \mathbf{U})\|\frac{\nabla\boldsymbol{\varphi}_j + \nabla\boldsymbol{\varphi}_j^T}{2}\right). \tag{47}$$

We emphasize that the system of equations (38) is *coupled* through two terms: (i) $\mathbf{a}^T\mathbf{B}\mathbf{a}$, which represents the nonlinearity $(\mathbf{u}^r\cdot\nabla)\mathbf{u}^r$; and (ii) $\mathbf{A}^r\mathbf{a}$, which represents the term $(\mathbf{u}^r\cdot\nabla)\mathbf{u}^r$ linearized around the centering trajectory $\mathbf{U}$. The difference between the VMS-POD-ROM (38)–(47) and the S-POD-ROM (30) and (31) is that the former acts only on the small resolved scales (since the Smagorinsky EV term $(C_S\delta)^2\|\mathbb{D}(\mathbf{u}_r^S + \mathbf{U})\|$ is included only in the equation corresponding to $\mathbf{a}^S$), whereas the latter acts on all (both large and small) resolved scales.

The VMS-POD-ROM (38)–(47) was announced in [57]. This study, however, represents its first careful derivation and thorough investigation in the numerical simulation of a 3D turbulent flow. We note that a fundamentally different VMS LES closure model that utilizes the NSE residual was proposed in [61]; this model was used in a POD setting in [26]. Yet another VMS-POD-ROM, inspired from the numerical stabilization methods developed in [62–65], was proposed, analyzed and tested in [66]. We emphasize that the VMS-POD-ROM (38)–(47) is different from both the model used in [26] and that used in [66].

3.3.4. Dynamic subgrid-scale POD reduced-order model (DS-POD-ROM)

For all three POD-ROM closure models defined above (i.e., ML-POD-ROM (27) and (28), S-POD-ROM (30) and (31), and VMS-POD-ROM (38)–(47)), the definition has been entirely phenomenological. Indeed, arguing that the role of the discarded POD modes is to extract energy from the system, we used an EV ansatz to derive closure models of increasing complexity and physical accuracy. The *dynamic subgrid-scale* (*DS*) POD-ROM closure model is also of EV type. Its derivation, however, requires a precise definition of the filtering operation. The DS closure model has its origins in LES, where it is considered state-of-the-art (see for example, [42]). In LES, the filtering operation is effected by convolving the flow variables with a rapidly decaying spatial filter. In POD, the filtering operation is effected by using the POD Galerkin projection described in Section 3.1 (see (14)). To derive the precise POD filtered equations, we start with the NSE (1) in which the velocity $\mathbf{u}$ is replaced by its POD approximation $\mathbf{u}(\mathbf{x},t) \approx \mathbf{u}_r(\mathbf{x},t) \equiv \mathbf{U}(\mathbf{x}) + \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\mathbf{x})$ in (6), and obtain

$$\frac{\partial\mathbf{u}_r}{\partial t} - Re^{-1}\Delta\mathbf{u}_r + (\mathbf{u}_r\cdot\nabla)\mathbf{u}_r + \nabla p = 0. \tag{48}$$

Using the fact that $\nabla\cdot\mathbf{u}_r = 0$ in (48), we get $(\mathbf{u}_r\cdot\nabla)\mathbf{u}_r = \nabla\cdot(\mathbf{u}_r\mathbf{u}_r)$. Thus, (48) can be rewritten as

$$\frac{\partial\mathbf{u}_r}{\partial t} - Re^{-1}\Delta\mathbf{u}_r + \nabla\cdot(\mathbf{u}_r\mathbf{u}_r) + \nabla p = 0. \tag{49}$$

Applying the POD filtering operation (14) to (49), using the fact that the POD Galerkin projection is a linear operator, and *assuming* that differentiation and POD filtering commute, we obtain

$$\frac{\partial\bar{\mathbf{u}}_r}{\partial t} - Re^{-1}\Delta\bar{\mathbf{u}}_r + \nabla\cdot(\overline{\mathbf{u}_r\mathbf{u}_r}) + \nabla\bar{p} = 0. \tag{50}$$

We note that, if filtering and differentiation do not commute, one has to estimate the commutation error (see for example, [67–69]). We also note that, since the POD filtering operation is the Galerkin projection (14), $\bar{\mathbf{u}}_r = \mathbf{u}_r$. To maintain a notation consistent with the nonlinear term, we continue to use the $\bar{\mathbf{u}}_r$ notation in the following development.

The POD filtered Eq. (50) can be rewritten as

$$\frac{\partial\bar{\mathbf{u}}_r}{\partial t} - Re^{-1}\Delta\bar{\mathbf{u}}_r + \nabla\cdot(\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r) + \nabla\cdot(\boldsymbol{\tau}_r) + \nabla\bar{p} = 0, \tag{51}$$

where

$$\boldsymbol{\tau}_r = \overline{\mathbf{u}_r\mathbf{u}_r} - \bar{\mathbf{u}}_r\bar{\mathbf{u}}_r \tag{52}$$

is the POD subfilter-scale stress tensor. Thus, the POD-G-ROM (7) amounts to setting $\boldsymbol{\tau}_r = 0$. For turbulent flows, as we have already mentioned, this approximation is flawed. Thus, one needs to address the POD closure problem, i.e., to model the POD subfilter-scale stress tensor $\boldsymbol{\tau}_r$ in terms of the POD filtered velocity $\bar{\mathbf{u}}_r$. We note that the POD closure problem is exactly the LES closure problem, in which the spatial filtering is replaced by POD Galerkin projection. For all three POD-ROM closure models defined so far in this section (i.e., ML-POD-ROM (27) and (28), S-POD-ROM (30) and (31), and VMS-POD-ROM (38)–(47)), the closure problem has been addressed by assuming an EV ansatz for $\boldsymbol{\tau}_r$. The DS-POD-ROM employs an EV ansatz as well; specifically, the Smagorinsky model is utilized:

$$\boldsymbol{\tau}_r \approx -2(C_S\delta)^2\|\mathbb{D}(\bar{\mathbf{u}}_r)\|\mathbb{D}(\bar{\mathbf{u}}_r), \tag{53}$$

in which C_S is not a constant (as in the Smagorinsky model), but a function of space and time, i.e., $C_S = C_S(\mathbf{x}, t)$. To compute $C_S(\mathbf{x}, t)$, we follow the LES derivation in [42] and replace the LES spatial filtering with the POD Galerkin projection. Since there are two spatial filters in the LES derivation of the DS model, we define a second POD Galerkin projection (in addition to that defined in (14)): For all $\mathbf{u} \in \mathbf{X}$, the second (test) Galerkin projection $\tilde{\mathbf{u}} \in \mathbf{X}^R$ (where $R < r$) is the solution of the following equation:

$$(\mathbf{u} - \tilde{\mathbf{u}}, \boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\varphi} \in \mathbf{X}^R. \tag{54}$$

Applying the second POD filtering operation (54) to (50), we obtain:

$$\frac{\partial \widetilde{\bar{\mathbf{u}}}_r}{\partial t} - Re^{-1}\Delta\widetilde{\bar{\mathbf{u}}}_r + \nabla\cdot(\widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r) + \nabla\cdot(\mathbf{T}_r) + \nabla\widetilde{\bar{p}} = 0, \tag{55}$$

where

$$\mathbf{T}_r = \widetilde{\overline{\mathbf{u}_r\mathbf{u}_r}} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r \tag{56}$$

is the second POD subfilter-scale stress tensor. We note that the following identity (known as the "Germano identity" in LES) holds:

$$\mathbf{T}_r = \widetilde{\overline{\mathbf{u}_r\mathbf{u}_r}} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r = \left(\widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r\right) + \left(\widetilde{\overline{\mathbf{u}_r\mathbf{u}_r}} - \widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r}\right) = \mathbf{L}_r + \widetilde{\boldsymbol{\tau}}_r, \tag{57}$$

where $\mathbf{L}_r = \widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r$ and $\widetilde{\boldsymbol{\tau}}_r = \widetilde{\overline{\mathbf{u}_r\mathbf{u}_r}} - \widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r}$. We *assume the same EV ansatz* for the two POD subfilter-scale stress tensors, $\boldsymbol{\tau}_r$ and $\mathbf{T}_r$:

$$\mathbf{T}_r \approx -2(C_S\widetilde{\delta})^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r), \tag{58}$$

$$\boldsymbol{\tau}_r \approx -2(C_S\delta)^2\|\mathbb{D}(\bar{\mathbf{u}}_r)\|\mathbb{D}(\bar{\mathbf{u}}_r), \tag{59}$$

where $\widetilde{\delta}$ is the filter radius used in the second POD filtering operation (54). *Assuming* that C_S remains constant under the second POD filtering (54), we obtain:

$$\widetilde{\boldsymbol{\tau}}_r \approx -2(C_S\delta)^2\|\mathbb{D}\widetilde{(\bar{\mathbf{u}}_r)\|\mathbb{D}}(\bar{\mathbf{u}}_r) \approx -2(C_S\delta)^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r). \tag{60}$$

Substituting (58) and (60) into (57) we obtain:

$$-2(C_S\widetilde{\delta})^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r) = \left(\widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r\right) - 2(C_S\delta)^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r). \tag{61}$$

We note that C_S is the only unknown in (61), all other terms being computable quantities. Since all the terms in (61) are tensors, the unknown C_S cannot satisfy all six equations. Thus, the following least squares approach is considered instead:

$$\min_{C_S^2}\left[\left(\widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r\right) - 2(C_S\delta)^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r) + 2(C_S\widetilde{\delta})^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\right] : \left[\left(\widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r\right) - 2(C_S\delta)^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r) + 2(C_S\widetilde{\delta})^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\right]. \tag{62}$$

The solution $C_S(\mathbf{x}, t)$ of (62) is:

$$C_S^2(\mathbf{x},t) = \frac{\left[\widetilde{\bar{\mathbf{u}}_r\bar{\mathbf{u}}_r} - \widetilde{\bar{\mathbf{u}}}_r\widetilde{\bar{\mathbf{u}}}_r\right] : \left[2\delta^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r) - 2\widetilde{\delta}^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\right]}{\left[2\delta^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r) - 2\widetilde{\delta}^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\right] : \left[2\delta^2\|\mathbb{D}(\bar{\mathbf{u}}_r\widetilde{)\|\mathbb{D}}(\bar{\mathbf{u}}_r) - 2\widetilde{\delta}^2\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\|\mathbb{D}(\widetilde{\bar{\mathbf{u}}}_r)\right]}. \tag{63}$$

Since the stress tensors can be computed directly from the resolved field, (63) yields a time- and space-dependent formula for $C_S(\mathbf{x}, t)$.

Thus, the DS-POD-ROM increases the viscosity coefficient by

$$\nu_{DS} := (C_S(\mathbf{x}, t)\delta)^2\|\mathbb{D}(\mathbf{u}_r)\|, \tag{64}$$

where $C_S(\mathbf{x}, t)$ is the coefficient in (63), δ is the lengthscale defined in Section 3.2 and $\|\mathbb{D}(\mathbf{u}_r)\|$ the Frobenius norm of the deformation tensor $\mathbb{D}(\mathbf{u}_r)$. Thus, the *dynamic subgrid-scale POD reduced-order model* (*DS-POD-ROM*) has the form (24), where

$$\widetilde{b}_k(\mathbf{a}) = -2\delta^2\left(\nabla\boldsymbol{\varphi}_k, C_S^2(\mathbf{x}, t)\|\mathbb{D}(\mathbf{u}_r)\|\frac{\nabla\mathbf{U} + \nabla\mathbf{U}^T}{2}\right), \tag{65}$$

$$\widetilde{A}_{km}(\mathbf{a}) = -2\delta^2\left(\nabla\boldsymbol{\varphi}_k, C_S^2(\mathbf{x}, t)\|\mathbb{D}(\mathbf{u}_r)\|\frac{\nabla\boldsymbol{\varphi}_m + \nabla\boldsymbol{\varphi}_m^T}{2}\right). \tag{66}$$

Note that ν_{DS} defined in (64) can take negative values. This can be interpreted as *backscatter*, the inverse transfer of energy from high index POD modes to low index modes. The notion of backscatter, well-established in LES (see for example, [42]), was also found in a POD setting in the numerical investigation in [43].

4. Numerical tests

In this section, we use a structurally dominated 3D turbulent flow problem to test the four POD-ROMs described in Section 3: (i) the ML-POD-ROM (27) and (28); (ii) the S-POD-ROM (30) and (31); (iii) the new VMS-POD-ROM (38)–(47); and (iv) the new DS-POD-ROM (65) and (66). We also include results for the POD-G-ROM (7) (i.e., a POD-ROM without any closure model). A successful POD closure model should at least perform better than the POD-G-ROM (7). Finally, a DNS projection of the evolution of the POD modes served as benchmark for our numerical simulations: The closeness to the DNS data was used as a criterion for the success of the POD closure model. The qualitative behavior of all POD-ROMs is judged according to the following five criteria: (i) the kinetic energy spectrum; (ii) the mean velocity; (iii) the Reynolds stresses; (iv) the *root mean square* (*rms*) values of the velocity fluctuations; and (v) the time evolution of the POD coefficients. The first four criteria measure the temporal and spatial average behavior of the POD-ROMs, whereas the last criterion measures the instantaneous behavior of the POD-ROMs. We also include a computational efficiency assessment for all four POD-ROMs as well as a sensitivity study to measure the robustness of the numerical results with respect to changes in r, the number of POD modes retained in the POD-ROMs. In Section 4.1, details of the numerical methods and parameter choices are given. In Section 4.2, numerical results are presented and discussed.

4.1. Numerical methods and parameter choices

We investigate all four POD-ROMs in the numerical simulation of 3D flow past a circular cylinder at $Re = 1000$. The Reynolds number is computed using the diameter (D) of the cylinder as the lengthscale and the freestream velocity (U_∞) as the velocity scale. The wake of the flow is fully turbulent. The cylinder is parallel to the z-axis and the freestream flow is in the positive x-direction (see Fig. 2). In this section, u denotes the streamwise velocity component (associated with the x-axis), v denotes the normal velocity component (associated with the y-axis), and w denotes the spanwise velocity component (associated with the z-axis). A parallel CFD solver is employed on the time interval $[0, 300]$ to generate the DNS data [70]. Details on the numerical discretization are presented in Appendix A.

Collecting 1000 snapshots of the velocity field (u, v, w) over the time interval $[0, 75]$ and applying the method of snapshots developed in [15], we obtain the POD basis $\{\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_N\}$. These POD modes are then interpolated onto a structured quadratic finite element triangulation with nodes coinciding with the nodes used in the original DNS finite difference discretization. The first $r = 6$ POD modes capture 84% the system's kinetic energy. These modes are used in all POD-ROMs that we investigate next. For all the POD-ROMs, the time discretization was effected by using the explicit Euler method with $\Delta t = 7.5 \times 10^{-4}$ and the spatial discretization used piecewise quadratic Lagrange finite elements ($\mathbb{P}_2$).

POD-ROMs for the pressure field have also been developed [36,56]. The advantage of these POD-ROMs is that the computation of hydrodynamic forces on structures becomes possible. A pressure-Poisson based POD-ROM that requires the POD modes of the pressure field in addition to those for the velocity field was proposed in [56]. This model predicted the lift and drag forces on a cylinder at $Re = 100$. In this report, however, we only develop POD-ROMs of the velocity field. Thus, lift and drag forces cannot be computed and compared with those from DNS.

It is important to note that the quadratic nonlinearity in the NSE (1) allows for easy precomputation of the vector $\mathbf{b}$, the matrix $\mathbf{A}$ and the tensor $\mathbf{B}$ in the POD-G-ROM (8). For the general nonlinear EV POD closure model (24), however, the vector $\widetilde{\mathbf{b}}(\mathbf{a})$ and the matrix $\widetilde{\mathbf{A}}(\mathbf{a})$ that correspond to the additional closure terms have to be recomputed (reassembled) at each time step. Since the POD basis functions are global, although only a few are used in POD-ROMs ($r \ll N$), reassembling $\widetilde{\mathbf{b}}(\mathbf{a})$ and $\widetilde{\mathbf{A}}(\mathbf{a})$ at each time step would dramatically increase the CPU time of the corresponding POD-ROM. Thus, a major advantage of POD-ROMs (the dramatic decrease of computational time), would be completely lost.

To ensure a high computational efficiency of the POD-ROMs, we utilize two approaches: (i) Instead of updating the closure terms in the POD-ROMs every time step, we recompute them every 1.5 time units (i.e., every 20,000 time steps). The previous numerical investigations in [50] showed that this approach does not compromise the numerical accuracy of the S-POD-ROM (30) and (31). (ii) We employ the two-level algorithm introduced in [50] to discretize the nonlinear closure models. We emphasize that, in order to maintain a fair numerical comparison of the four POD-ROMs, we used both algorithmic choices (i) and (ii) listed above in *all* four POD-ROMs. Therefore, the success or failure of the POD-ROM can solely be attributed to the closure term, which is the only distinguishing feature among all POD-ROMs, and not to the specific algorithmic choices, which are the same for all POD-ROMs.

The two-level algorithm used in all four POD-ROMs is summarized below.

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

(67)

In (67), M represents the total number of time steps. The idea in the two-level algorithm is straightforward: Instead of computing the closure terms $\widetilde{\mathbf{b}}(\mathbf{a}^{\ell}), \widetilde{\mathbf{A}}(\mathbf{a}^{\ell})$ directly on the fine mesh (as done in the standard one-level algorithm), the two-level algorithm discretizes them on a *coarser mesh*. Thus, the two-level algorithm is much more efficient than the standard one-level algorithm. Indeed, in [50] it was shown that the two-level algorithm (67) achieves the same level of accuracy as the one-level algorithm while decreasing the computational cost by an order of magnitude. In all four POD-ROMs, we apply the two-level algorithm with a coarsening factor $R_c = 4$ in both radial and azimuthal directions. Thus, the vectors and matrices related to the nonlinear closure terms are computed on a coarse finite element mesh with $37 \times 49 \times 17$ grid points.

In Section 3.2, we proposed two definitions for the POD lengthscale δ. In the finite element discretization that we employ, definition (23) is harder to implement than (22), therefore we use the latter. Using definition (22) with $r = 6$, we obtain $\delta = 0.1179$, which is the POD lengthscale that we will use in all four POD-ROMs. For the DS-POD-ROM (65) and (66), we need to define the second (test) Galerkin projection (54) and the corresponding filter radius $\widetilde{\delta}$. Choosing $R = 1$ in (54) and using (22), we obtain $\widetilde{\delta} = 0.1769$.

The constants in EV LES models are determined in a straightforward fashion, utilizing scaling laws satisfied by general 3D turbulent flows (see for example, [42]). Although the energy cascade concept in a POD context was verified numerically in [43], there are no general scaling laws available in this setting. Thus, to our knowledge, the correct values for the EV constants α in the ML-POD-ROM (27) and (28) and C_S in the S-POD-ROM (30) and (31) and the new VMS-POD-ROM (38)–(47) are still not known. To determine these EV constants, we run the corresponding POD-ROM on the short time interval $[0, 15]$ with several different values for the EV constants and choose the value that yields the results that are closest to the DNS results. This approach yields the following values for the EV constants: $\alpha = 3 \times 10^{-3}$ for the ML-POD-ROM, $C_S = 0.1426$ for the S-POD-ROM, and $C_S = 0.1897$ for the VMS-POD-ROM. We emphasize that these EV constant values are optimal only on the short time interval tested, and they might actually be non-optimal on the entire time interval $[0, 300]$ on which the POD-ROMs are tested. Thus, this heuristic procedure ensures some fairness in the numerical comparison of the four POD-ROMs.

In the VMS-POD-ROM, only the first POD mode is considered as the large resolved POD mode, that is, $r_L = 1$ in (33). In the DS-POD-ROM, since v_{DS} can be negative, we use a standard "clipping" procedure to ensure the numerical stability of the discretization (see for example, [42]). Specifically, we let $C_S(\mathbf{x}, t) = \max\{C_S(\mathbf{x}, t), -0.2\}$. The value -0.2 is determined as follows. We run the DS-POD-ROM without "clipping" for the time interval $[0,15]$ and record $C^{-}_{S,ave}$, the average negative value of $C_S(\mathbf{x}, t)$. We then evaluate the DS-POD-ROM on the entire time interval $[0,300]$ with the "clipping" value $C^{-}_{S,ave}/2 = -0.2$. We note that there are alternative procedures to deal with the same issue in LES, such as VDSMwc [71]. We utilized the standard "clipping" procedure described above as a first step in the numerical investigation of the DS-POD-ROM.

4.2. Numerical results

Before presenting the quantitative comparison of the POD-ROMs, we give a flavor of the topology of the resulting flow fields. Fig. 1 presents the first streamwise POD mode (top left), the first normal POD mode (top right), the third streamwise POD mode (bottom left), and the third normal POD mode (bottom right). Fig. 2 presents snapshots of horizontal velocity at $t = 142.4$ s for DNS, POD-G-ROM, ML-POD-ROM, S-POD-ROM, VMS-POD-ROM, and DS-POD-ROM. For clarity, only five isosurfaces are drawn. Taking the DNS results as a benchmark, the POD-G-ROM seems to add unphysical structures. The ML-POD-ROM, on the other hand, appears to add too much numerical dissipation to the system and thus eliminates some of the vortical structures in the wake. The S-POD-ROM, VMS-POD-ROM, and DS-POD-ROM perform well, capturing a similar amount of structure as the DNS. It also seems that there is some phase shift for all these POD-ROMs. Due to space limitations, only one time instance snapshot is shown for the POD-ROMs. The general behavior over the entire time interval is similar; it can be found at <http://www.math.vt.edu/people/wangzhu/POD_3DNumComp.html>.

Fig. 3 presents the energy spectra of the four POD-ROMs and, for comparison purposes, of the POD-G-ROM. The five energy spectra

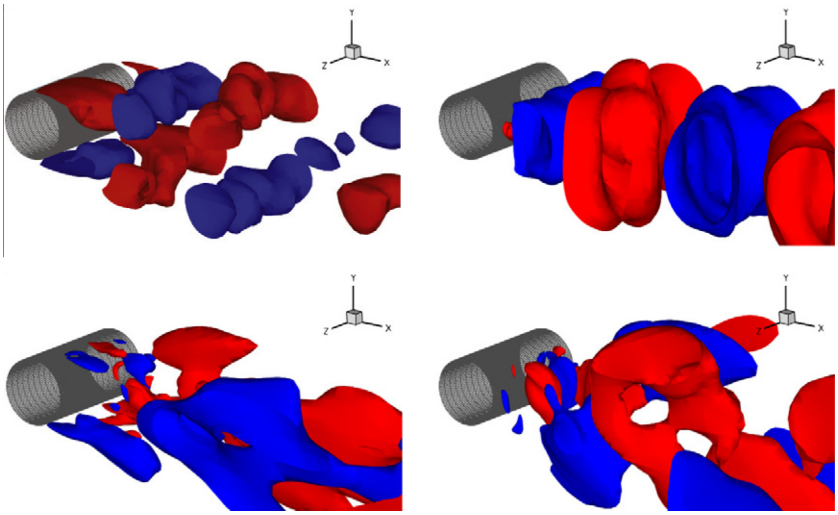

Fig. 1. First streamwise POD mode (top left), first normal POD mode (top right), third streamwise POD mode (bottom left), and third normal POD mode (bottom right).

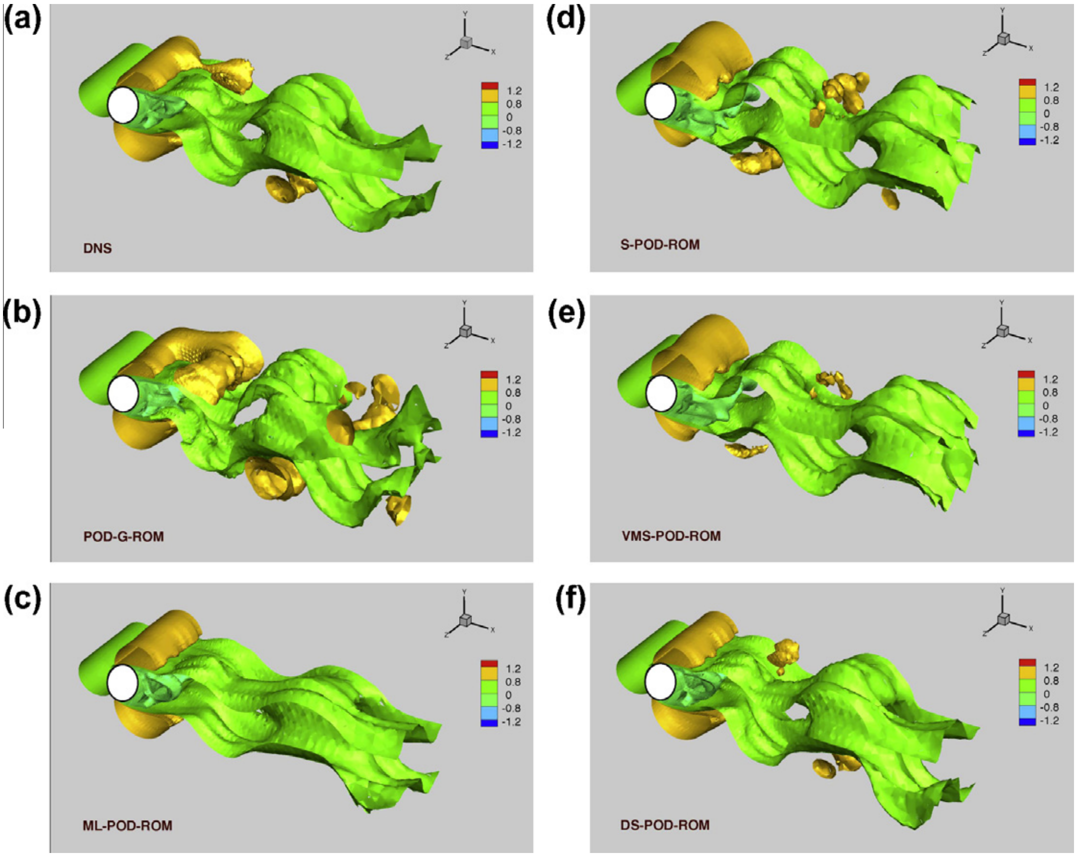

Fig. 2. Snapshots of horizontal velocity at $t = 142.5$ s for: (a) DNS; (b) the POD-G-ROM (7); (c) the ML-POD-ROM (27) and (28); (d) the S-POD-ROM (30) and (31); (e) the new VMS-POD-ROM (38)–(47); and (f) the new DS-POD-ROM (65) and (66). Five isosurfaces are plotted.

are compared with the DNS energy spectrum. All energy spectra are calculated from the average kinetic energy of the nodes in the cube with side 0.1 centered at the probe (0.9992, 0.3575, 1.0625). It is clear that the energy spectrum of the POD-G-ROM overestimates the energy spectrum of the DNS. The energy spectrum of the ML-POD-ROM, on the other hand, underestimates the energy spectrum of the DNS, especially at the higher frequencies. The S-POD-ROM has a more accurate spectrum than the ML-POD-ROM, but displays high oscillations at the higher frequencies. The VMS-POD-ROM is a clear improvement over the S-POD-ROM, with smaller oscillations at the higher frequencies. The energy spectrum of the DS-POD-ROM is qualitatively similar to that of the VMS-POD-ROM. The DS-POD-ROM spectrum decreases the amplitude of the high frequency oscillations of the VMS-POD-ROM even further, although it introduces some sporadic large amplitude oscillations at high frequencies. To summarize, the DS-POD-ROM and the VMS-POD-ROM yield the most accurate energy spectra, i.e., the closest to the DNS energy spectrum. On the

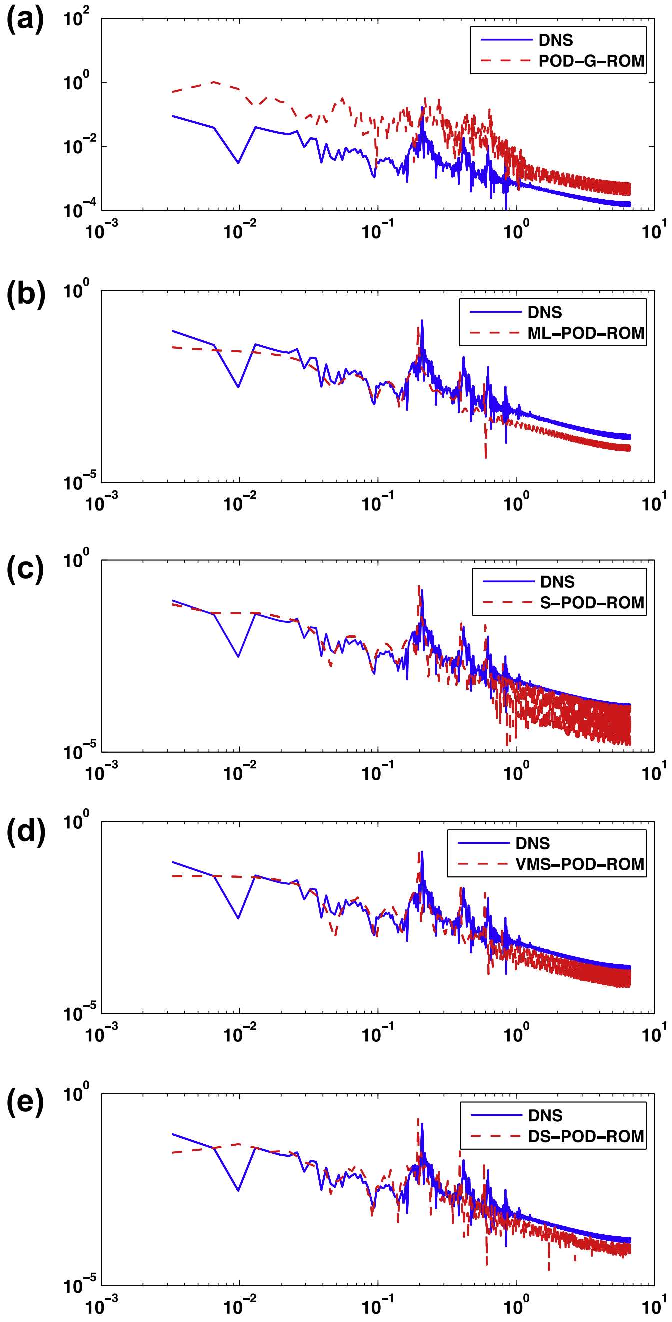

Fig. 3. Kinetic energy spectrum of the DNS (blue) and the POD-ROMs (red): (a) the POD-G-ROM (7); (b) the ML-POD-ROM (27) and (28); (c) the S-POD-ROM (30) and (31); (d) the new VMS-POD-ROM (38)–(47); and (e) the new DS-POD-ROM (65) and (66). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

average, the DS-POD-ROM performs slightly better than the VMS-POD-ROM.

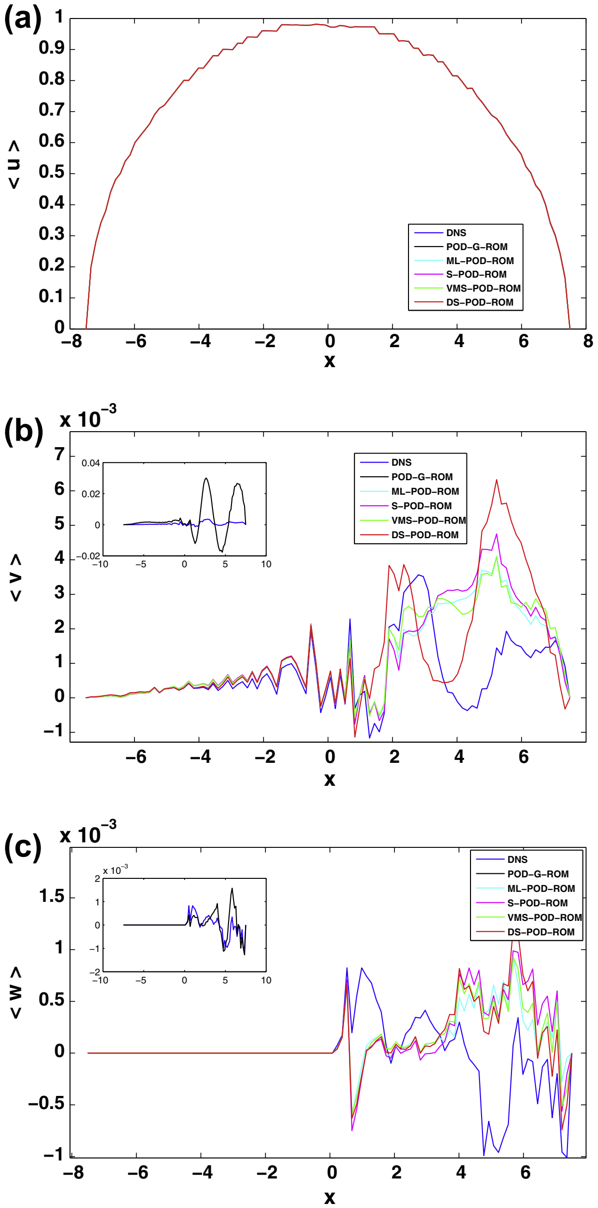

Fig. 4. Mean velocity components for DNS and POD-ROMs: (a) $\langle u \rangle$ (the mean streamwise velocity), (b) $\langle v \rangle$ (the mean normal velocity), and (c) $\langle w \rangle$ (the mean spanwise velocity), where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

The second criterion in the comparison of the four POD-ROMs is the mean velocity components: $\langle u \rangle$ (the mean streamwise velocity), $\langle v \rangle$ (the mean normal velocity), and $\langle w \rangle$ (the mean spanwise velocity). The time averaging is effected on the interval $[0, 300]$ and the spatial averaging is performed in the *yz*-direction ($\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$). Since the topology of the velocity field is markedly different in the *x*-, *y*-, and *z*-directions (see Figs. 1 and 2), one could also consider spatial averaging in the *xz*-direction ($\langle \cdot \rangle = \langle \cdot \rangle_{txz}$). We note that, since the numerical results with spatial averaging in the *xz*-direction were qualitatively similar to those with spatial averaging in the *yz*-direction, they were not included in this report. These numerical results, however, can be found in [25]. In Fig. 4, we plot the mean velocity components for different values of *x*. Fig. 4 yields the following conclusions: First, the mean streamwise velocity is computed accurately by all POD-ROMs. Second, the POD-G-ROM yields inaccurate results for the mean normal velocity; all the other POD-ROMs perform significantly better than the POD-G-ROM. Third, the mean spanwise velocity results are similar for the ML-POD-ROM, the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM; the POD-G-ROM performs better than all these POD-ROMs over certain regions, and worse over other regions.

As a third criterion in the comparison of the POD-ROMs, we utilize the Reynolds stresses: $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$ (the *xy*-component of the Reynolds stress), $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$ (the *xz*-component of the Reynolds stress), and $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$ (the *yz*-component of the Reynolds stress), where $\langle \cdot \rangle$ represents the temporal and spatial averaging operator. The time averaging is effected on the interval

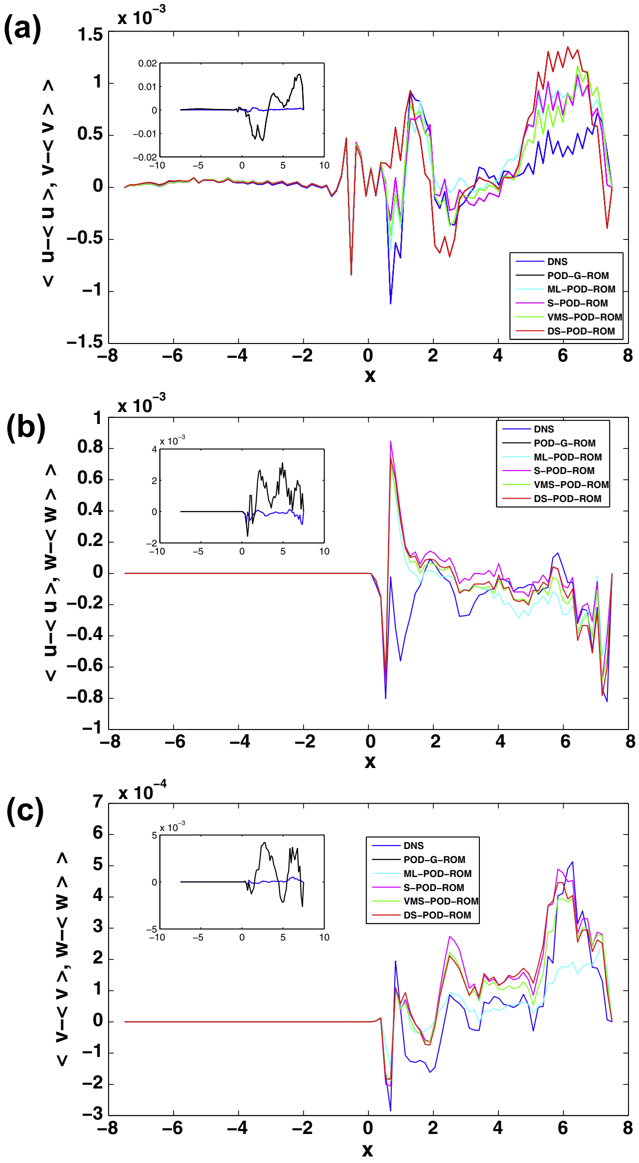

Fig. 5. Reynolds stresses for DNS and POD-ROMs: (a) $\langle u-\langle u\rangle, v-\langle v\rangle\rangle$ (the *xy*-component of the Reynolds stress), (b) $\langle u-\langle u\rangle, w-\langle w\rangle\rangle$ (the *xz*-component of the Reynolds stress), and (c) $\langle v-\langle v\rangle, w-\langle w\rangle\rangle$ (the *yz*-component of the Reynolds stress), where $\langle\cdot\rangle=\langle\cdot\rangle_{tyz}$.

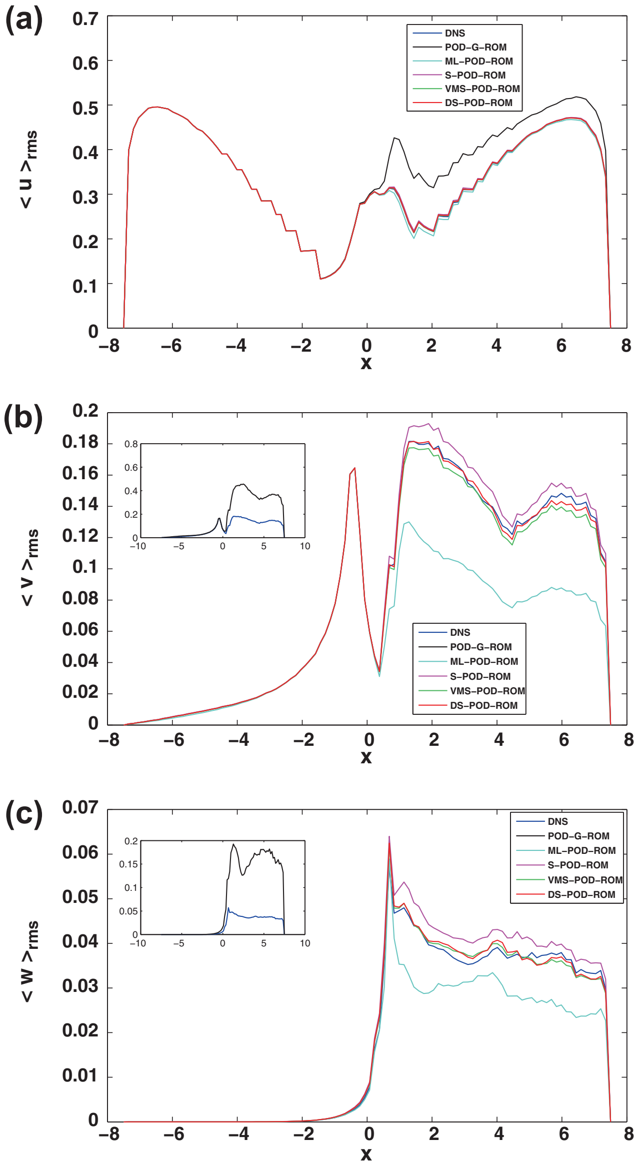

Fig. 6. Rms values of the velocity fluctuations for DNS and POD-ROMs: (a) $\langle u\rangle_{rms}=\langle u-\langle u\rangle, u-\langle u\rangle\rangle$ (the rms value of the streamwise velocity fluctuations), (b) $\langle v\rangle_{rms}=\langle v-\langle v\rangle, v-\langle v\rangle\rangle$ (the rms value of the normal velocity fluctuations), and (c) $\langle w\rangle_{rms}=\langle w-\langle w\rangle, w-\langle w\rangle\rangle$ (the rms value of the spanwise velocity fluctuations), where $\langle\cdot\rangle=\langle\cdot\rangle_{tyz}$.

$[0,300]$ and the spatial averaging is performed in the *yz*-direction ($\langle\cdot\rangle=\langle\cdot\rangle_{tyz}$). In Fig. 5, we plot the Reynolds stresses for different values of *x*. Fig. 5 yields the following conclusions. First, the POD-G-ROM Reynolds stresses are consistently the most inaccurate (i.e., the farthest from the DNS Reynolds stresses). The second conclusion is that the ML-POD-ROM, the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM all have similar behaviors. Indeed, different POD-ROMs might outperform the others over different spatial regions, but there is no clear "winner" over the entire spatial interval for any of the Reynolds stresses and spatial averaging used.

The fourth criterion in the POD-ROM comparison is the rms values of the velocity fluctuations: $\langle u\rangle_{rms}=\langle u-\langle u\rangle, u-\langle u\rangle\rangle$ (the rms of the streamwise velocity fluctuations), $\langle v\rangle_{rms}=\langle v-\langle v\rangle, v-\langle v\rangle\rangle$ (the rms of the normal velocity fluctuations), and $\langle w\rangle_{rms}=\langle w-\langle w\rangle, w-\langle w\rangle\rangle$ (the rms of the spanwise velocity fluctuations). The time averaging is effected on the interval $[0,300]$ and the spatial averaging is performed in the *yz*-direction ($\langle\cdot\rangle=\langle\cdot\rangle_{tyz}$). In Fig. 6, we plot the rms values for different values of *x*. Fig. 6 yields the following conclusions. Similar to the Reynolds stresses case, the POD-G-ROM rms values of the velocity fluctuations are consistently the most inaccurate (i.e., the farthest from the DNS rms values). The rms plots corresponding to the four POD-ROMs, however, display a clear, consistent ordering this time. Indeed, the DS-POD-ROM and the VMS-POD-ROM consistently outperformed the other two POD-ROMs (the S-POD-ROM and the ML-POD-ROM), especially when the $\langle v\rangle_{rms}$ and the $\langle w\rangle_{rms}$ plots are considered. The S-POD-ROM consistently performs worse than the DS-POD-ROM and the VMS-POD-ROM, but is clearly more accurate than the ML-POD-ROM.

As the fifth criterion in judging the performance of the POD-ROMs, the time evolutions of the POD basis coefficients $a_1(\cdot)$ and $a_4(\cdot)$ on the entire time interval $[0,300]$ are shown in Figs. 7 and 8. We note that the other POD coefficients have similar behavior. Thus, for clarity of exposition, we include only $a_1(\cdot)$ and $a_4(\cdot)$. The POD-G-ROM's time evolutions of a_1 and a_4 are clearly inaccurate. Indeed, the magnitude of a_4 is nine times larger than that of the

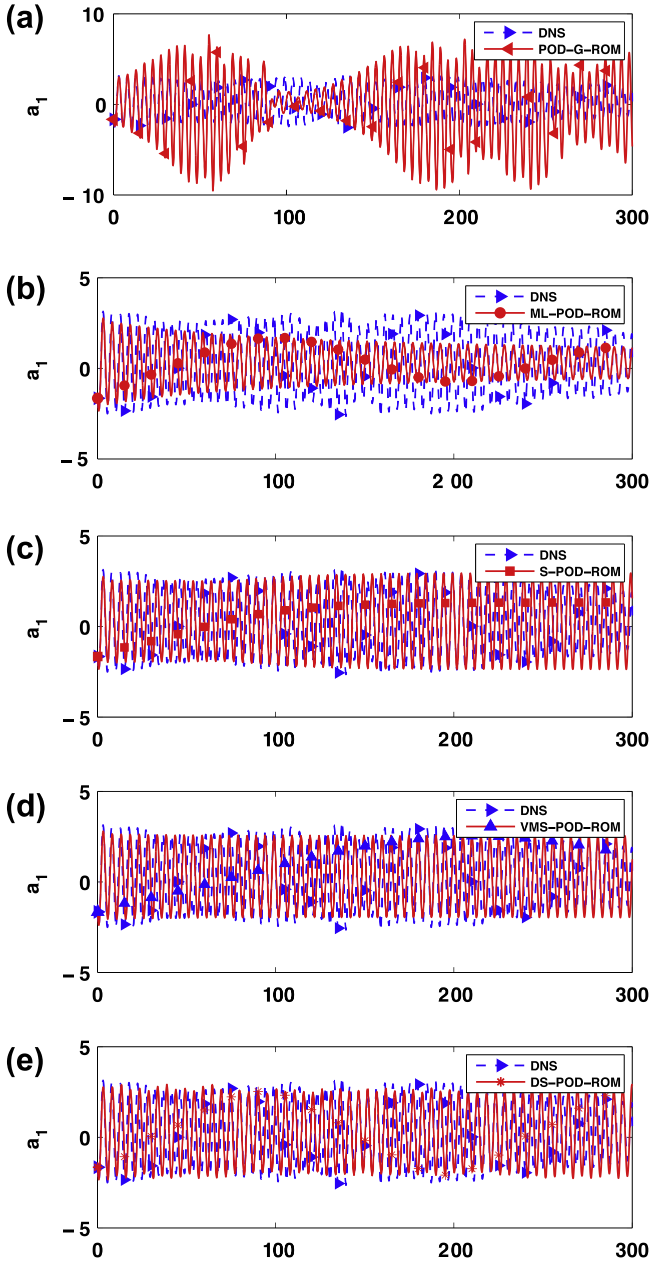

Fig. 7. Time evolution of the POD basis coefficient a_1 of the DNS (blue) and the POD-ROMs (red): (a) the POD-G-ROM (7); (b) the ML-POD-ROM (27) and (28); (c) the S-POD-ROM (30) and (31); (d) the new VMS-POD-ROM (38)–(47); and (e) the new DS-POD-ROM (65) and (66). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

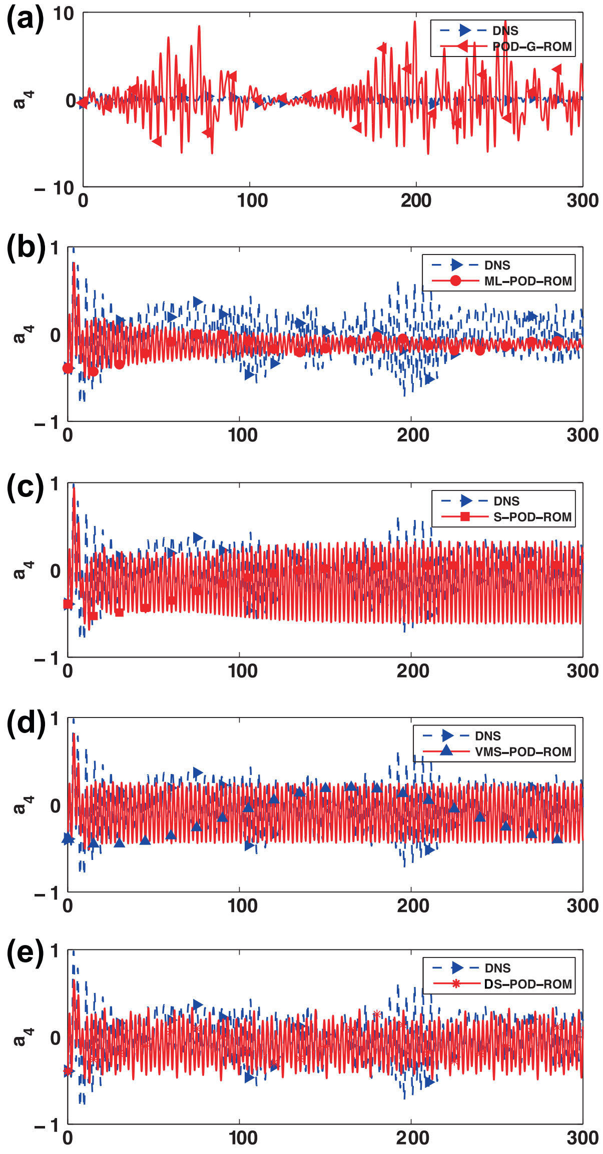

Fig. 8. Time evolution of the POD basis coefficient a_4 of the DNS (blue) and the POD-ROMs (red): (a) the POD-G-ROM (7); (b) the ML-POD-ROM (27) and (28); (c) the S-POD-ROM (30) and (31); (d) the new VMS-POD-ROM (38)–(47); and (e) the new DS-POD-ROM (65) and (66). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

DNS projection, which indicates the need for closure modeling. The ML-POD-ROM's time evolutions of a_1 and a_4 are also inaccurate. Specifically, although the time evolution at the beginning of the simulation (where the EV constant α was chosen) is relatively accurate, the accuracy significantly degrades toward the end of the simulation. For example, the magnitude of a_4 at the end of the simulation is only one eighth of that of the DNS. The S-POD-ROM yields more accurate time evolutions than the ML-POD-ROM for both a_1 and a_4, although the magnitude of the POD coefficients stays almost constant at a high level. The VMS-POD-ROM's time evolutions of a_1 and a_4 are better than those of the S-POD-ROM, since the magnitudes of the POD coefficients are closer to those of the DNS. Finally, the DS-POD-ROM also yields accurate results. We note that the DS-POD-ROM's a_1 and a_4 coefficients have significantly more variability than the corresponding coefficients of the VMS-POD-ROM. This is a consequence of the fact that the EV coefficient C_S varies in time and space for the DS-POD-ROM, whereas it is constant for the VMS-POD-ROM. To summarize, the DS-POD-ROM and the VMS-POD-ROM perform the best. On the average, the DS-POD-ROM performs slightly better than the VMS-POD-ROM.

Based on the overall results (the kinetic energy spectrum, the mean velocity, the Reynolds stresses, the rms values of the velocity fluctuations, and the time evolutions of the POD basis coefficients $a_1(\cdot)$ and $a_4(\cdot)$), the DS-POD-ROM and the VMS-POD-ROM outperform the ML-POD-ROM and the S-POD-ROM. To determine which one of the DS-POD-ROM and the VMS-POD-ROM performs best, we collected the results in Figs. 8(d) and 8(e) (corresponding to the time evolution of the POD basis coefficient $a_4(\cdot)$ for the DNS projection, the VMS-POD-ROM and the DS-POD-ROM) and we displayed them in the same plot in Fig. 9. Since it is difficult to distinguish between the results from the VMS-POD-ROM and the DS-

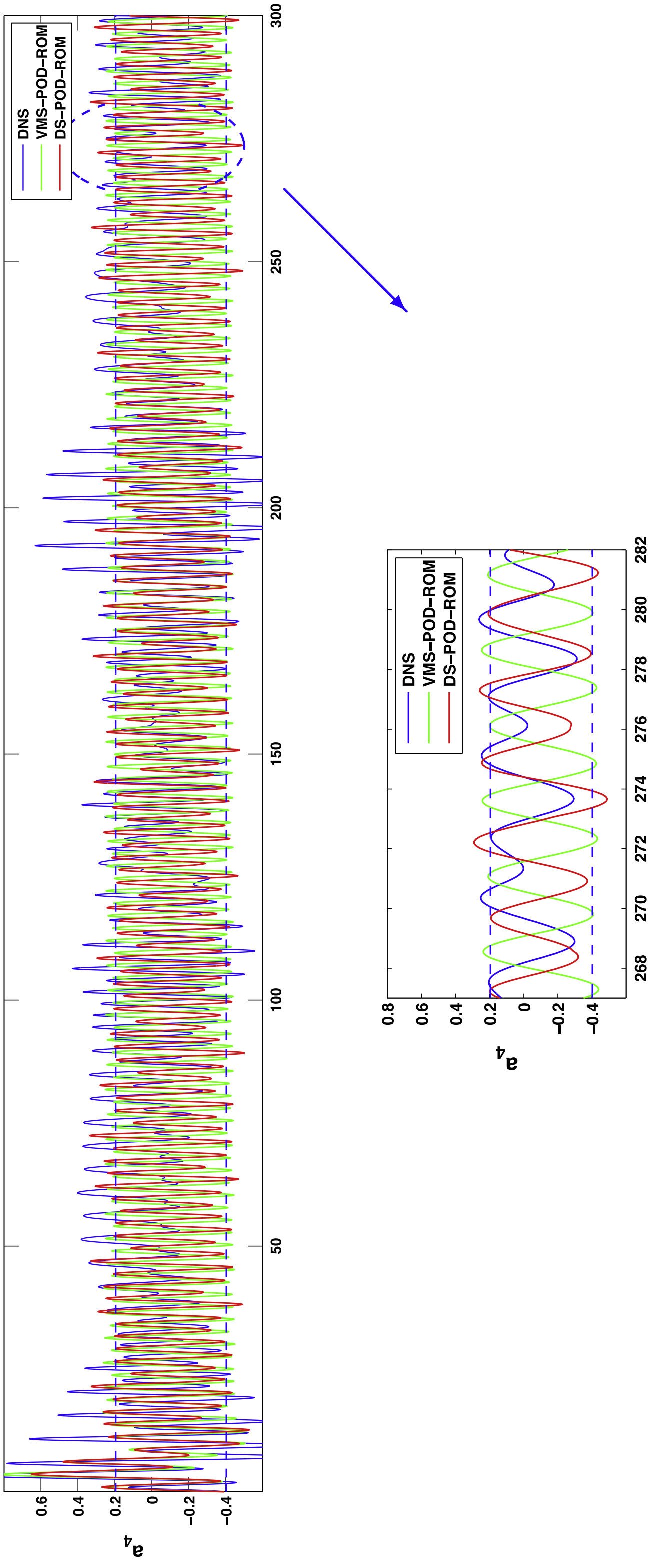

Fig. 9. Time evolution of the POD basis coefficient a_4 of the DNS (blue), the new VMS-POD-ROM (38)–(47) (green), and the new DS-POD-ROM (65) and (66) (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Speed-up factors of POD-ROMs.

	POD-G-ROM	ML-POD-ROM	S-POD-ROM	VMS-POD-ROM	DS-POD-ROM
S_f	665	659	36	41	23

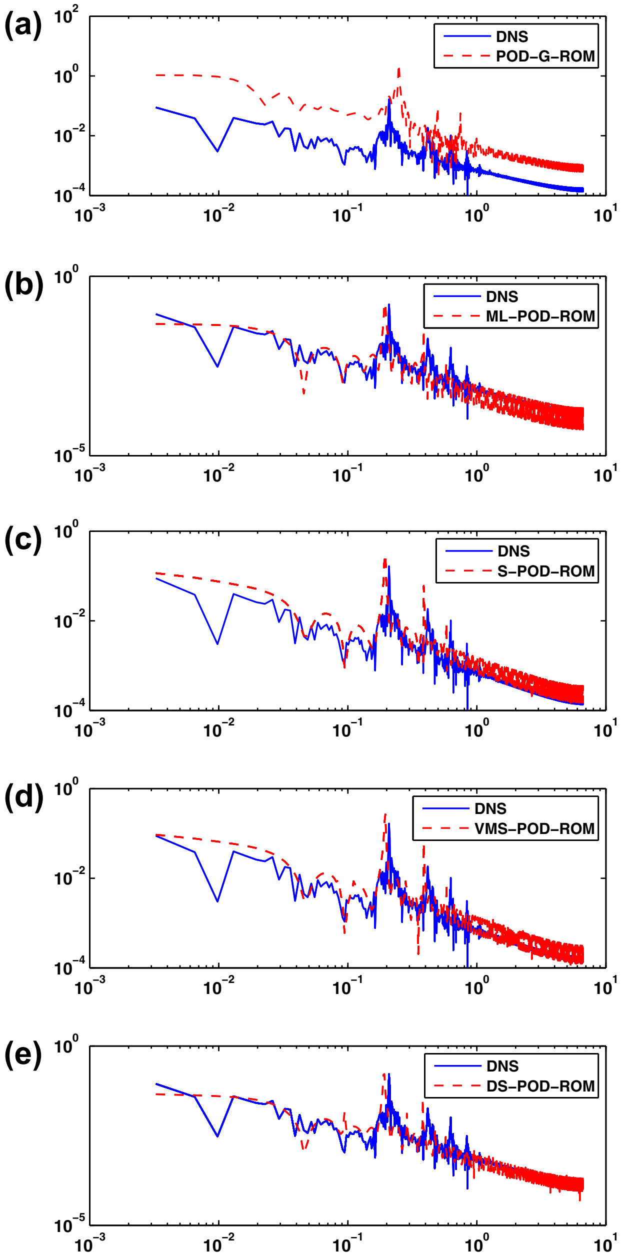

Fig. 10. Kinetic energy spectrum of the DNS (blue) and the POD-ROMs (red) for $r = 4$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

POD-ROM, we zoomed in on the POD basis coefficient a_4 over the time interval [266,282]. Based on the plot in the inset, it is clear that, for this time interval, the DS-POD-ROM performs better than the VMS-POD-ROM. More importantly, it appears that the magnitude of a_4 in the DS-POD-ROM displays some of the variability displayed by the DNS; the magnitude of the VMS-POD-ROM's a_4 coefficient, on the other hand, displays an almost periodic behavior. We believe that the variation of the DS-POD-ROM's a_4 coefficient is due to the dynamic computation of the EV coefficient, which changes in both space and time; the EV coefficient of the

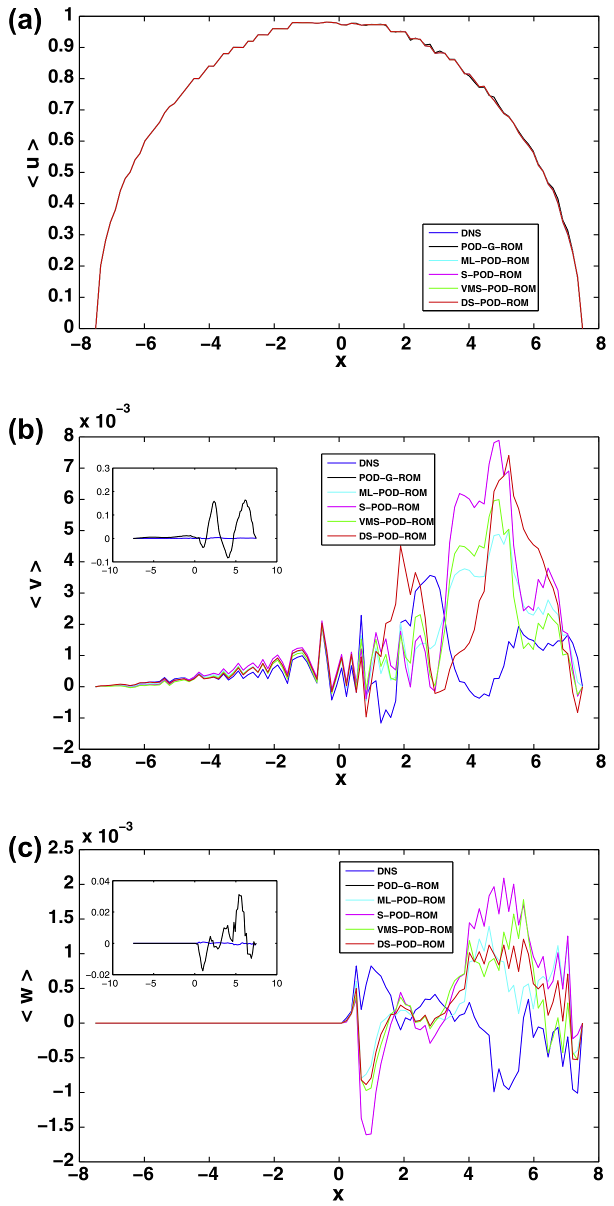

Fig. 11. Mean velocity components for DNS and POD-ROMs for $r = 4$: (a) $\langle u \rangle$ (the mean streamwise velocity), (b) $\langle v \rangle$ (the mean normal velocity), and (c) $\langle w \rangle$ (the mean spanwise velocity), where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

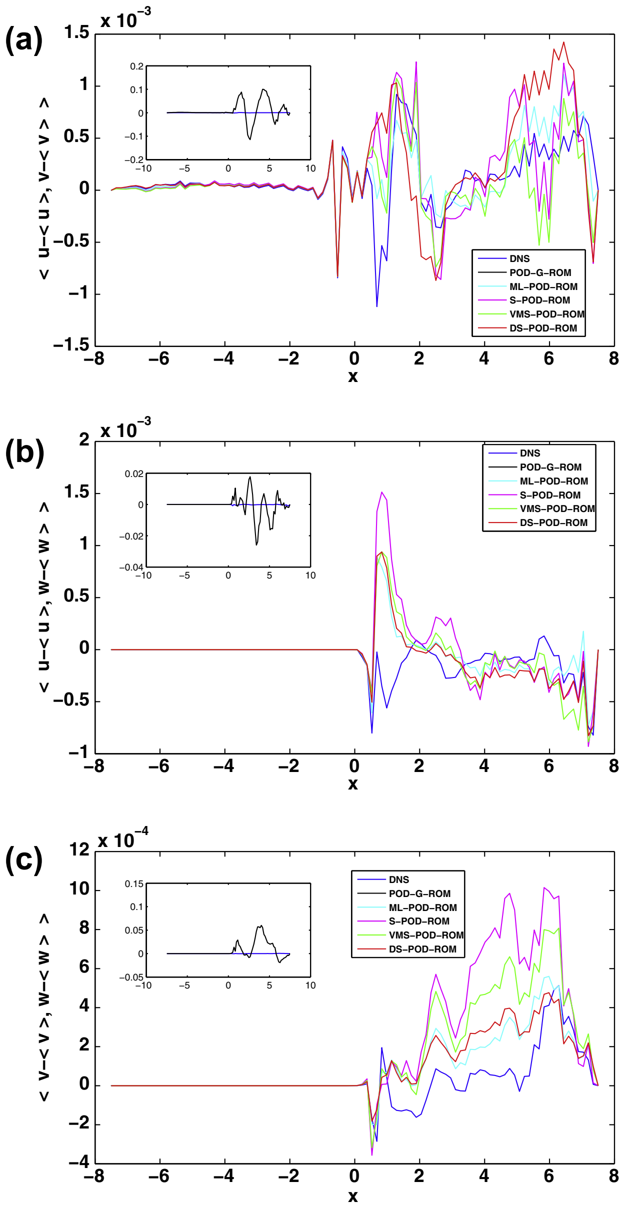

Fig. 12. Reynolds stresses for DNS and POD-ROMs for $r = 4$: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$ (the *xy*-component of the Reynolds stress), (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$ (the *xz*-component of the Reynolds stress), and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$ (the *yz*-component of the Reynolds stress), where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

VMS-POD-ROM, however, is constant and computed at the beginning of the simulation.

To summarize, the VMS and DS approaches, which are state-of-the-art closure modeling strategies in LES, yield the most accurate POD closure models for the 3D turbulent flow that we investigated. Indeed, the DS-POD-ROM and the VMS-POD-ROM clearly yield: (i) the best energy spectra (with a plus for the DS-POD-ROM, see Fig. 3); (ii) the best rms values (see Fig. 6); and (iii) the best time evolutions of the POD coefficients a_1 and a_4 (with an advantage for the DS-POD-ROM, see Figs. 7–9). Furthermore, with respect to the other two criteria (the mean velocity components in Fig. 4 and the Reynolds stresses in Fig. 5), the DS-POD-ROM and the VMS-POD-ROM perform at least as well as the other POD-ROMs. Thus, we conclude that the DS-POD-ROM and the VMS-POD-ROM yield the most accurate average and instantaneous numerical results.

A natural question, however, is whether the new POD closure modeling strategies that we proposed display a high level of computational efficiency, which is one of the trademarks of a successful POD-ROM. To answer this question, we computed the CPU times for all four POD-ROMs and compared them with those of the DNS and the POD-G-ROM.

To make such a comparison, however, we first need to address the numerical differences between the DNS and the POD-ROMs. First, the discretizations used in the two approaches are completely different. Indeed, the spatial discretization used in the DNS was the finite difference method, whereas for the POD-ROMs we used a finite element method. Furthermore, the time-discretization used in the DNS was second-order (Crank–Nicolson and Adams–Bashforth), whereas in the POD-ROMs 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. Most importantly, the DNS was performed on a parallel machine (on 16 processors), whereas all the POD-ROM runs were carried out on a single-processor machine. Thus, to en-

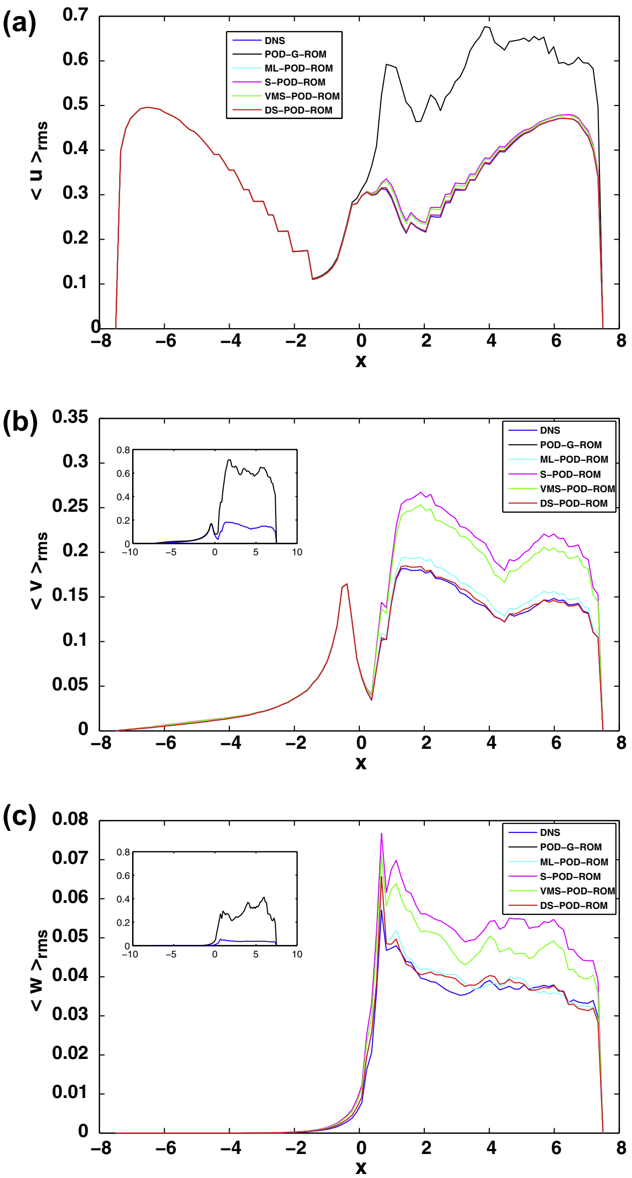

Fig. 13. Rms values of the velocity fluctuations for DNS and POD-ROMs for $r = 4$: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$ (the rms value of the streamwise velocity fluctuations), (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$ (the rms value of the normal velocity fluctuations), and (c) $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$ (the rms value of the spanwise velocity fluctuations), where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

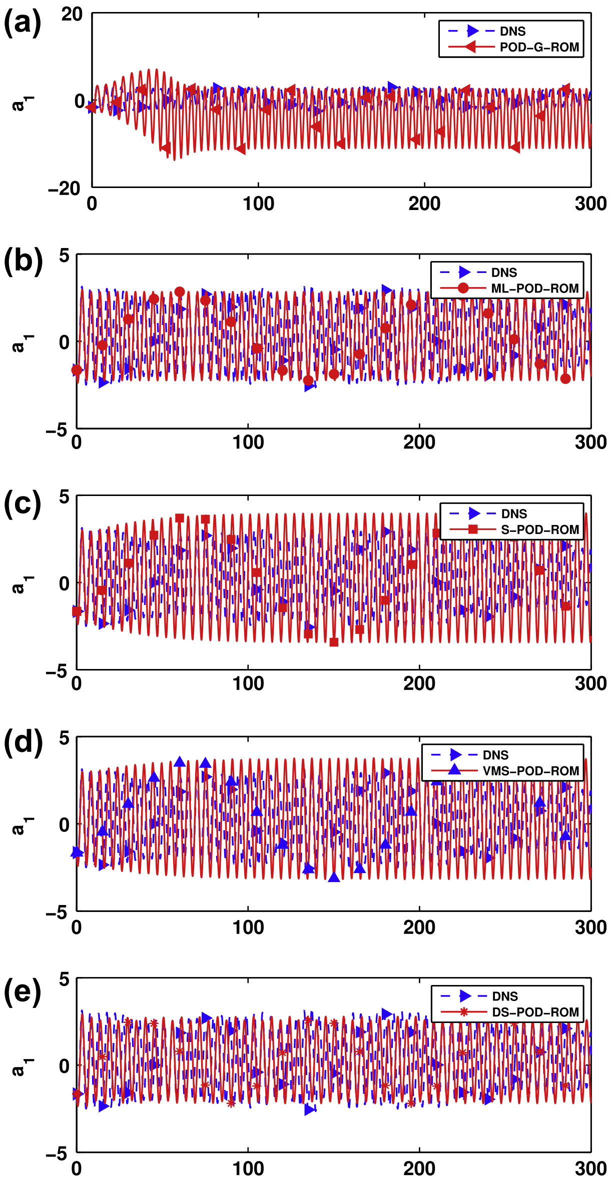

Fig. 14. Time evolutions of the POD basis coefficient a_1 of the DNS (blue) and the POD-ROMs (red) for $r = 4$: (a) the POD-G-ROM (7); (b) the ML-POD-ROM (27) and (28); (c) the S-POD-ROM (30) and (31); (d) the new VMS-POD-ROM (38)–(47); and (e) the new DS-POD-ROM (65) and (66). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sure a more realistic comparison between the CPU times of the DNS and the POD-ROMs, we multiplied the CPU time of the DNS by a factor of 16.

To measure the computational efficiency of the four POD-ROMs, we define the speed-up factor

$$S_f = \frac{\text{CPU time of DNS}}{\text{CPU time of POD-ROM}} \tag{68}$$

and list results in Table 1. The most efficient model is the POD-G-ROM. This is not surprising, since no closure model is used in POD-G-ROM and thus no CPU time is spent computing an additional nonlinear term at each time step. The second most efficient model is the ML-POD-ROM. This is again natural, since only a linear closure model is employed in the ML-POD-ROM and thus the computational overhead is minimal. The speed-up factors for the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM are one order of magnitude lower than those for the ML-POD-ROM and the POD-G-ROM. The reason is that the former use nonlinear closure models, which increase significantly the computational time. Note, however, that the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM are still significantly more efficient than the DNS.

Finally, to assess the robustness of the POD-ROMs, we conduct a sensitivity study with respect to r, the number of POD modes retained in the models. To this end, we repeat the above numerical investigation of the POD-ROMs, this time, however, with *fewer* POD modes: We use $r = 4$ instead of $r = 6$, as we have used so far. The question we are trying to address is whether the conclusions drawn from the previous numerical results remain valid when r is reduced from 6 to 4. We note that, since the numerical results with spatial averaging in the *xz*-direction were qualitatively similar to those with spatial averaging in the *yz*-direction, they were not included in this report. These numerical results, however, can be found in [25].

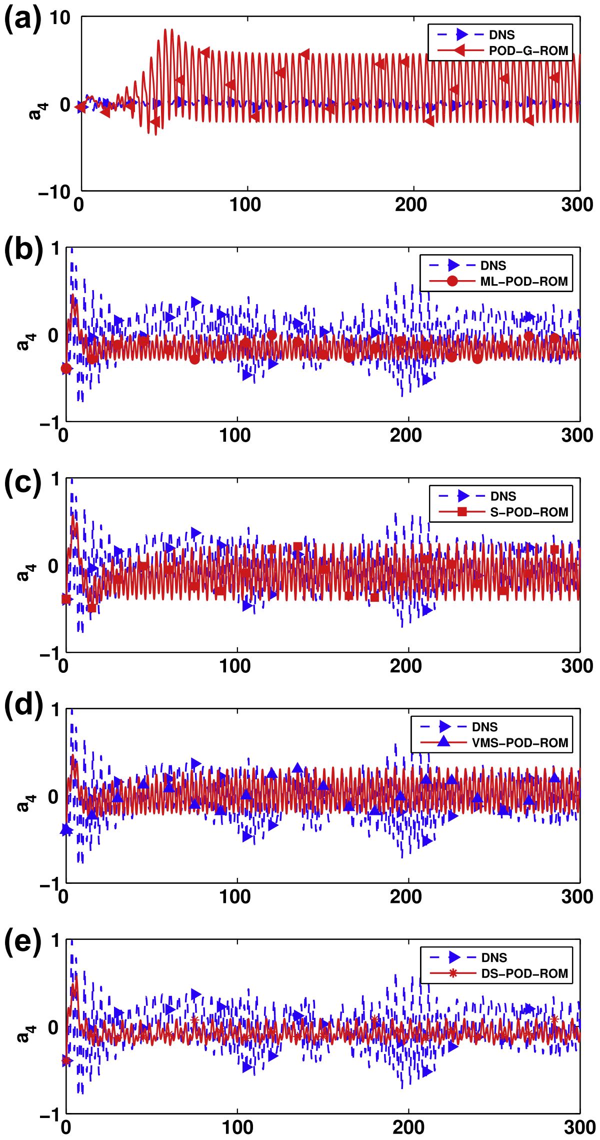

Fig. 15. Time evolutions of the POD basis coefficient a_4 of the DNS (blue) and the POD-ROMs (red) for $r = 4$: (a) the POD-G-ROM (7); (b) the ML-POD-ROM (27) and (28); (c) the S-POD-ROM (30) and (31); (d) the new VMS-POD-ROM (38)–(47); and (e) the new DS-POD-ROM (65) and (66). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 10 presents the energy spectra of the POD-ROMs and the DNS. The DS-POD-ROM is clearly the most accurate model, as was the case for $r = 6$ (see Fig. 3).

Fig. 11 presents the mean velocity components of the POD-ROMs and the DNS. The same conclusions as in the $r = 6$ (see Fig. 4) can be drawn: (i) all the POD-ROMs predict the mean streamwise velocity accurately; and (ii) all the POD-ROMs (except the POD-G-ROM) predict the normal velocity component well. In contrast with the $r = 6$ case, however, the POD-G-ROM is much worse than the other POD-ROMs in predicting the spanwise velocity component.

Fig. 12 presents the Reynolds stresses of the POD-ROMs and the DNS. The same conclusions as for $r = 6$ study (see Fig. 5) can be drawn: (i) the POD-G-ROM yields inaccurate results; and (ii) all other POD-ROMs perform similarly.

Fig. 13 presents the rms values of the velocity fluctuations of the POD-ROMs and the DNS. As in the $r = 6$ case (see Fig. 6), the POD-G-ROM yields very inaccurate results. There is, however, a difference between the results for $r = 4$ and those for $r = 6$. Indeed, the most accurate results in Fig. 13 are those for the DS-POD-ROM (as in the $r = 6$ case), but the ML-POD-ROM performs better than the VMS-POD-ROM. We do not have an explanation for these relatively inaccurate results yielded by the VMS-POD-ROM. The S-POD-ROM performs badly, just as in the $r = 6$ case.

Figs. 14 and 15 present the time evolutions of the POD basis coefficients $a_1(\cdot)$ and $a_4(\cdot)$ on the entire time interval $[0, 300]$. As in the $r = 6$ case (see Fig. 7), the DS-POD-ROM is the most accurate model for predicting the time evolution of $a_1(\cdot)$, followed (in this order) by the ML-POD-ROM, the VMS-POD-ROM, the S-POD-ROM, and the POD-G-ROM (which is the most inaccurate model). The ordering for the time evolution of $a_4(\cdot)$, however, is different: the most accurate models seem to be the VMS-POD-ROM and the S-POD-ROM, followed by the DS-POD-ROM, the ML-POD-ROM, and the POD-G-ROM (which is again the most inaccurate model).

To summarize, the same general conclusions as in the $r = 6$ case can be drawn in the $r = 4$ case: Overall, the DS-POD-ROM and the VMS-POD-ROM yield the most accurate average and instantaneous numerical results. Furthermore, with a few exceptions, the POD-ROMs results display a relatively low sensitivity with respect to changes in r, the number of POD modes employed in the models.

5. Conclusions

This paper put forth two new POD-ROMs (the DS-POD-ROM and the VMS-POD-ROM), which are inspired by state-of-the-art LES closure modeling strategies. These two new POD-ROMs together with the ML-POD-ROM and the S-POD-ROM were tested in the numerical simulation of a 3D turbulent flow past a cylinder at $Re = 1000$. For completeness, we also included results with the POD-G-ROM (i.e., a POD-ROM without any closure model), as well as the DNS projection of the evolution of the POD modes, which served as benchmark for our numerical simulations.

To assess the performance of the POD-ROMs, five criteria were considered in this paper: (i) the kinetic energy spectrum; (ii) the mean velocity; (iii) the Reynolds stresses; (iv) the root mean square values of the velocity fluctuations; and (v) the time evolution of the POD coefficients. The first four criteria measure the temporal and spatial average behavior of the POD-ROMs, whereas the last criterion measures the instantaneous behavior of the POD-ROMs. Based on the numerical results, the following general conclusions were drawn: Both the POD-G-ROM and the ML-POD-ROM yielded inaccurate results. The DS-POD-ROM and the VMS-POD-ROM clearly outperformed these two models, yielding more accurate results. The DS-POD-ROM generally performed slightly better than the VMS-POD-ROM and seemed to display more adaptivity in terms of adjusting the magnitude of the POD basis coefficients. Overall, however, the two models yielded similar qualitative results. This seems to reflect the LES setting, where both DS and VMS closure modeling strategies are considered state-of-the-art [59,60]. The DS-POD-ROM and the VMS-POD-ROM, although not as computationally efficient as the POD-G-ROM or the ML-POD-ROM, significantly decreased the CPU time of the DNS. Finally, to assess the robustness of the POD-ROMs, we conducted a sensitivity study with respect to r, the number of POD modes retained in the models. To this end, we repeated the above numerical investigation of the POD-ROMs, this time, however, using $r = 4$ instead of $r = 6$. The same general conclusions were drawn in the $r = 4$ case: Overall, the DS-POD-ROM and the VMS-POD-ROM yielded the most accurate average and instantaneous numerical results. Furthermore, with a few exceptions, the POD-ROMs results displayed a relatively low sensitivity with respect to changes in r. To summarize, for the 3D turbulent flow that we investigated, the DS-POD-ROM and the VMS-POD-ROM were found to perform the best among the POD-ROMs investigated, combining a relative high numerical accuracy with a high level of computational efficiency.

We plan to further investigate several other research avenues. First, we plan to study more efficient time-discretization approaches and take advantage of parallel computing in order to further decrease the computational time and, at the same time, increase the dimension (and thus the physical accuracy) of the POD-ROMs. Second, since the linear closure model (ML-POD-ROM) is computationally efficient, but only works on a relative short time interval if the appropriate EV coefficient α is chosen, we will investigate a hybrid approach: We will use the DS approach to calculate α only when the flow displays a high level of variability, and then use this value in the ML-POD-ROM as long as the flow does not experience sudden transitions. Third, using these computational developments, we will investigate the new POD-ROMs in higher Reynolds number structurally dominated turbulent flows. These challenging flows will represent the ultimate test for the new POD-ROM closure models: they will reveal whether POD can fulfill its original promise of enabling accurate and efficient modeling of realistic structurally dominated turbulent flows. Due to the physical complexity of the underlying flows, increased numbers of snapshots and POD modes need to be considered to ensure the accuracy of the POD-ROM. To tackle the computational complexity of the POD-ROMs (and especially their nonlinear terms), we plan to use the new two-level algorithms [50] in conjunction with the empirical interpolation method [44] and the discrete interpolation method [45,46]. Finally, we plan to employ the new POD-ROMs in other scientific and engineering applications in which accurate POD closure modeling is needed, such as optimal control, optimization, and data assimilation problems.

Acknowledgments

We thank the referees for their comments and suggestions, which significantly improved the manuscript. We also greatly appreciate the financial support of the Air Force Office of Scientific Research through grant FA9550-08-1-0136 and of the National Science Foundation through grant DMS-1016450. A significant part of the computations were carried out on SystemX at Virginia Tech's Advanced Research Computing center (http://wwfw.arc.vt.edu). The allocation grant and support provided by the staff are gratefully acknowledged.

Appendix A

We briefly describe the numerical method used to generate the DNS data for the test problem in Section 4. For more details on the discretization, validation, verification, and parallel implementation of the numerical scheme, the reader is referred to [56,70].

A parallel CFD algorithm is employed to solve the incompressible NSE. Utilizing a strong conservative form, the NSE are written in curvilinear coordinates and are nondimensionalized using the diameter (D) of the cylinder as the lengthscale and the freestream velocity (U_∞) as the velocity scale.

A body conformal "O"-type grid and curvilinear coordinates in an Eulerian reference frame are used to simulate the 3D flow past a cylinder at Re = 1000. The NSE are approximated numerically using an approach similar to that employed in [72]. A 2D domain decomposition technique ensuring that each processor gets a "slice" of the grid is used to implement the algorithm on a distributed-memory, message-passing parallel computer. The transformed NSE are approximated numerically on a non-staggered-grid layout. The Cartesian velocity components and pressure are defined at the center of the control volume in the computational space and the volume fluxes 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 [73]. 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.

Table 2
3D flow past a cylinder at Re = 1000. Validation of DNS results. Notice that the mean drag coefficient $\overline{C_D}$ and Strouhal number St match those in [78,79].

Data source	$\overline{C_D}$	St
Experiment [78]	1.0	0.21
3D DNS [79]	1.02	0.202
3D DNS (present)	1.11	0.205

A semi-implicit scheme is utilized 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.

A fractional-step method [72,74–77] is used to advance the solution in time. This 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.

A DNS of the flow past a circular cylinder is performed on a $144 \times 192 \times 32$ grid distributed over 16 processors, in circumferential direction. Thus, the load per processor is $144 \times 12 \times 32$ 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 $\approx$0.2, which corresponds to a nondimensional time step size $\Delta t = 2 \times 10^{-3}$ for this grid.

To validate our DNS results, we compute the mean drag coefficient $\overline{C_D}$ and the Strouhal number St. In Table 2, these quantities are compared with those obtained by Norberg [78] and Evangelinos and Karniadakis [79]. Note that the values obtained by using our DNS results match those in [78,79].

The 1D energy spectrum at $(x/D, y/D, z/D) = (1.0, 0.5, 1.0)$ exhibits a $-\frac{5}{3}$ law in the inertial range, which extends about half a decade in wave number [70].

References

[1] K. Ito, S.S. Ravindran, A reduced-order method for simulation and control of fluid flows, J. Comput. Phys. 143 (1998) 403–425.

[2] W.R. Graham, J. Peraire, K.Y. Tang, Optimal control of vortex shedding using low-order models. Part II – model-based control, Int. J. Numer. Methods Engrg. 44 (1999) 973–990.

[3] K. Cohen, S. Siegel, T. McLaughlin, E. Gillies, Feedback control of a cylinder wake low-dimensional model, AIAA J. 41 (2003) 1389–1391.

[4] M. Bergmann, L. Cordier, J. Brancher, Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model, Phys. Fluids 17 (2005) 097101.

[5] O. Lehmann, M. Luchtenburg, B.R. Noack, R. King, M. Morzyński, G. Tadmor, Wake stabilization using POD Galerkin models with interpolated modes, in: Proceedings of the 44th IEEE Conference on Decision and Control, 2005.

[6] J. Hoepffner, E. Akervik, U. Ehrenstein, D.S. Henningson, Control of cavity-driven separated boundary layer, in: Proceedings of the Conference on Active Flow Control, Berlin, 2006.

[7] S. Bagheri, L. Brandt, D. Henningson, Input–output analysis model reduction and control of the flat-plate boundary layer, J. Fluid Mech. 620 (2009) 263–298.

[8] A. Barbagallo, D. Sipp, P. Schmid, Closed-loop control of an open cavity flow using reduced-order models, J. Fluid Mech. 641 (2009) 1–50.

[9] S. Ahuja, C.W. Rowley, Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators, J. Fluid Mech. 645 (2010) 447–478.

[10] I. Akhtar, A.H. Nayfeh, Model based control of laminar wake using fluidic actuation, J. Comput. Nonlinear Dyn. 5 (2010) 041015.

[11] 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, Comput. Methods Appl. Mech. Engrg. 196 (2007) 4184–4195.

[12] D.N. Daescu, I.M. Navon, A dual-weighted approach to order reduction in 4DVAR data assimilation, Mon. Weather Rev. 136 (2008) 1026–1041.

[13] 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 4D-Var adaptive mesh ocean modelling approach, Int. J. Numer. Methods Fluids 60 (2009) 709–732.

[14] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge, 1996.

[15] L. Sirovich, Turbulence and the dynamics of coherent structures. Parts I–III, Quart. Appl. Math. 45 (1987) 561–590.

[16] N. Aubry, P. Holmes, J.L. Lumley, E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, J. Fluid Mech. 192 (1988) 115–173.

[17] B. Podvin, J. Lumley, A low-dimensional approach for the minimal flow unit, J. Fluid Mech. 362 (1998) 121–155.

[18] B. Podvin, On the adequacy of the ten-dimensional model for the wall layer, Phys. Fluids 13 (2001) 210–224.

[19] B. Galletti, C.H. Bruneau, L. Zannetti, A. Iollo, Low-order modelling of laminar flow regimes past a confined square cylinder, J. Fluid Mech. 503 (2004) 161–170.

[20] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, J. Comput. Phys. 207 (2005) 192–220.

[21] M. Buffoni, S. Camarri, A. Iollo, M.V. Salvetti, Low-dimensional modelling of a confined three-dimensional wake flow, J. Fluid Mech. 569 (2006) 141–150.

[22] A. Iollo, A. Dervieux, J.A. Désidéri, S. Lanteri, Two stable POD-based approximations to the Navier–Stokes equations, Comput. Visual. Science 3 (2000) 61–66.

[23] A. Iollo, S. Lanteri, J.A. Désidéri, Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations, Theoret. Comput. Fluid Dyn. 13 (2000) 377–396.

[24] F. Fang, C.C. Pain, I.M. Navon, G.J. Gorman, M.D. Piggott, P.A. Allison, P.E. Farrell, A.J.H. Goddard, A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows, Ocean Model. 28 (2009) 127–136.

[25] Z. Wang, Reduced-order modeling of complex engineering and geophysical flows: analysis and computations, Ph.D. thesis, Virginia Tech, 2012.

[26] M. Bergmann, C.H. Bruneau, A. Iollo, Enablers for robust POD models, J. Comput. Phys. 228 (2009) 516–538.

[27] B.R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzynski, P. Comte, G. Tadmor, A finite-time thermodynamics of unsteady fluid flows, J. Non-Equil. Thermodyn. 33 (2008) 103–148.

[28] B.R. Noack, M. Schlegel, M. Morzyński, G. Tadmor, System reduction strategy for Galerkin models of fluid flows, Int. J. Numer. Methods Fluids 63 (2010) 231–248.

[29] B.R. Noack, M. Morzynski, G. Tadmor, Reduced-Order Modelling for Flow Control, vol. 528, Springer Verlag, 2011.

[30] D. Rempfer, H.F. Fasel, Dynamics of three-dimensional coherent structures in a flat-plate boundary layer, J. Fluid Mech. 275 (1994) 257–283.

[31] D. Rempfer, Investigations of boundary layer transition via Galerkin projections on empirical eigenfunctions, Phys. Fluids 8 (1996) 175.

[32] W. Cazemier, R. Verstappen, A. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Phys. Fluids 10 (1998) 1685.

[33] X. Ma, G.E. Karniadakis, A low-dimensional model for simulating three-dimensional cylinder flow, J. Fluid Mech. 458 (2002) 181–190.

[34] S. Sirisup, G.E. Karniadakis, A spectral viscosity method for correcting the long-term behavior of POD models, J. Comput. Phys. 194 (2004) 92–116.

[35] B.R. Noack, P. Papas, P.A. Monkewitz, Low-dimensional Galerkin model of a laminar shear-layer, Technical Report 2002-01, École Polytechnique Fédérale de Lausanne, 2002.

[36] B.R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, J. Fluid Mech. 497 (2003) 335–363.

[37] B.R. Noack, P. Papas, P.A. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, J. Fluid Mech. 523 (2005) 339–365.

[38] S. Ullmann, J. Lang, A POD-Galerkin reduced model with updated coefficients for Smagorinsky LES, in: J.C.F. Pereira, A. Sequeira (Eds.), V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010, Lisbon, Portugal.

[39] A. Hay, J. Borggaard, D. Pelletier, Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition, J. Fluid Mech. 629 (2009) 41–72.

[40] A. Hay, J. Borggaard, I. Akhtar, D. Pelletier, Reduced-order models for parameter dependent geometries based on shape sensitivity analysis, J. Comput. Phys. 229 (2010) 1327–1352.

[41] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Int. J. Numer. Methods Eng. 86 (2011) 155–181.

[42] P. Sagaut, Large Eddy Simulation for Incompressible Flows, Scientific Computation, third edition., Springer-Verlag, Berlin, 2006.

[43] M. Couplet, P. Sagaut, C. Basdevant, Intermodal energy transfers in a proper orthogonal decomposition – Galerkin representation of a turbulent separated flow, J. Fluid Mech. 491 (2003) 275–284.

[44] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C.R. Acad. Sci. Paris, Ser. I 339 (2004) 667–672.

[45] S. Chaturantabut, D.C. Sorensen, J.C. Steven, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (2010) 2737–2764.

[46] S. Chaturantabut, D.C. Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction, SIAM J. Numer. Anal. 50 (2012) 46–63.

[47] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, Comput. Fluids 35 (2006) 208–226.

[48] D. Galbally, K. Fidkowski, K. Willcox, O. Ghattas, Non-linear model reduction for uncertainty quantification in large-scale inverse problems, Int. J. Numer. Methods Eng. 81 (2010) 1581–1608.

[49] B.T. Dickinson, J.R. Singler, Nonlinear model reduction using group proper orthogonal decomposition, Int. J. Numer. Anal. Model. 7 (2010) 356–372.

[50] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Two-level discretizations of nonlinear closure models for proper orthogonal decomposition, J. Comput. Phys. 230 (2011) 126–146.

[51] J.S. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–164.

[52] M. Germano, U. Piomelli, P. Moin, W. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A 3 (1991) 1760–1765.

[53] C. Meneveau, T. Lund, W. Cabot, A Lagrangian dynamic subgrid-scale model of turbulence, J. Fluid Mech. 319 (1996) 353–385.

[54] F. Porté-Agel, C. Meneveau, M.B. Parlange, A scale-dependent dynamic model for large-eddy simulation: application to a neutral atmospheric boundary layer, J. Fluid Mech. 415 (2000) 261–284.

[55] T.J.R. Hughes, L. Mazzei, K.E. Jansen, Large eddy simulation and the variational multiscale method, Comput. Vis. Sci. 3 (2000) 47–59.

[56] I. Akhtar, A.H. Nayfeh, C.J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows, Theoret. Comput. Fluid Dyn. 23 (2009) 213–237.

[57] J. Borggaard, A. Duggleby, A. Hay, T. Iliescu, Z. Wang, Reduced-order modeling of turbulent flows, in: Proceedings of MTNS 2008.

[58] B. Podvin, A proper-orthogonal-decomposition–based model for the wall layer of a turbulent channel flow, Phys. Fluids 21 (2009) 015111.

[59] T.J.R. Hughes, L. Mazzei, A. Oberai, A. Wray, The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence, Phys. Fluids 13 (2001) 505–512.

[60] T.J.R. Hughes, A. Oberai, L. Mazzei, Large eddy simulation of turbulent channel flows by the variational multiscale method, Phys. Fluids 13 (2001) 1784–1799.

[61] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (2007) 173–201.

[62] W.J. Layton, A connection between subgrid scale eddy viscosity and mixed methods, Appl. Math. Comput. 133 (2002) 147–157.

[63] J.-L. Guermond, Stabilization of Galerkin approximations of transport equations by subgrid modeling, M2AN Math, Model. Numer. Anal. 33 (1999) 1293–1316.

[64] V. John, S. Kaya, A finite element variational multiscale method for the Navier–Stokes equations, SIAM J. Sci. Comput. 26 (2005) 1485.

[65] V. John, A. Kindl, A variational multiscale method for turbulent flow simulation with adaptive large scale space, J. Comput. Phys. 229 (2010) 301–312.

[66] T. Iliescu, Z. Wang, Variational multiscale proper orthogonal decomposition: convection-dominated convection-diffusion-reaction equations, Math. Comput., in press.

[67] O. Vasilyev, D. Goldstein, Local spectrum of commutation error in large eddy simulation, Phys. Fluids 16 (2004) 470–473.

[68] O. Vasilyev, T. Lund, P. Moin, A general class of commutative filters for LES in complex geometries, J. Comput. Phys. 146 (1998) 82–104.

[69] L.C. Berselli, T. Iliescu, W.J. Layton, Mathematics of large eddy simulation of turbulent flows, Scientific Computation, Springer-Verlag, Berlin, 2006.

[70] I. Akhtar, Parallel simulations, reduced-order modeling and feedback control of vortex shedding using fluidic actuators, Ph.D. thesis, Virginia Tech, 2008.

[71] Y. Morinishi, O. Vasilyev, Vector level identity for dynamic subgrid scale modeling in large eddy simulation, Phys. Fluids 14 (2002) 3616.

[72] Y. Zang, R. Street, J. Koseff, A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates, J. Comput. Phys. 114 (1994) 18–33.

[73] B.P. Leonard, A stable and accurate convective modeling procedure based on quadratic upstream interpolation, Comput. Methods Appl. Mech. Engrg. 19 (1979) 59–98.

[74] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 135 (1997) 118–125.

[75] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes, J. Comput. Phys. 59 (1985) 308–323.

[76] S.A. Orszag, L.C. Kells, Transition to turbulence in plane Poiseuille and plane Couette flow, J. Fluid Mech. 96 (1985) 159–205.

[77] C.L. Street, M.Y. Hussaini, A numerical solution of the appearance of chaos in finite length Taylor–Couette flow, Appl. Numer. Math. 6 (1991) 123–139.

[78] C. Norberg, An experimental investigation of the flow around a circular cylinder: influence of aspect ratio, J. Fluid Mech. 258 (1994) 287–316.

[79] C. Evangelinos, G.E. Karniadakis, Dynamics and flow structures in the turbulent wake of rigid and flexible cylinders subject to vortex-induced vibrations, J. Fluid Mech. 400 (1999) 91–124.