Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 313 (2017) 512–534

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Approximate deconvolution reduced order modeling

Xuping Xie[a], David Wells[b], Zhu Wang[c], Traian Iliescu[a,*]

[a] *Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA*
[b] *Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA*
[c] *Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA*

Received 22 April 2016; received in revised form 17 August 2016; accepted 1 October 2016
Available online 12 October 2016

Highlights

- A new large eddy simulation reduced order model (LES-ROM) framework.
- A new approximate deconvolution ROM (AD-ROM) not of eddy viscosity type.
- Successful application of AD-ROM to 3D flow past cylinder at $Re = 1000$.

Abstract

This paper proposes a large eddy simulation reduced order model (LES-ROM) framework for the numerical simulation of convection-dominated flows. In this LES-ROM framework, the proper orthogonal decomposition (POD) is used to define the ROM basis and a ROM differential filter is used to define the large ROM structures. An approximate deconvolution (AD) approach is used to solve the ROM closure problem and develop a new AD-ROM. This AD-ROM is tested in the numerical simulation of the three-dimensional flow past a circular cylinder at a Reynolds number $Re = 1000$.

Keywords: Reduced order model; Proper orthogonal decomposition; Large eddy simulation; Approximate deconvolution; Spatial filter

1. Introduction

Reduced order models (ROMs) are an efficient approach to the numerical simulation of fluid flow, since they can reduce the computational time of a direct numerical simulation (DNS) by orders of magnitude while preserving key features of the flow. ROMs have been successfully used in numerous applications, such as fluid flow optimization and control, cardiovascular flows, or geophysical flows (see, e.g., the surveys in [1–6]). The proper orthogonal decomposition (POD) is one of the most successful methods for ROM development. An accurate numerical simulation is used in POD to extract the dominant structures, which are then used in a Galerkin approximation of the underlying equations [3,7]. In this paper, POD will be exclusively used to construct the ROMs.

Standard ROMs are extremely efficient and relatively accurate for laminar flows. They generally fail, however, in the numerical simulation of convection-dominated flows [8–15]. Indeed, to ensure a low computational cost, only the

[*] Corresponding author.
E-mail addresses: xupingxy@vt.edu (X. Xie), wellsd2@rpi.edu (D. Wells), wangzhu@math.sc.edu (Z. Wang), iliescu@vt.edu (T. Iliescu).
URLs: http://www.math.vt.edu/people/xupingxy (X. Xie), http://homepages.rpi.edu/~wellsd2 (D. Wells), http://people.math.sc.edu/wangzhu (Z. Wang), http://www.math.vt.edu/people/iliescu (T. Iliescu).

http://dx.doi.org/10.1016/j.cma.2016.10.005
0045-7825/© 2016 Elsevier B.V. All rights reserved.

first few POD modes are generally used in the ROM. The resulting low-dimensional ROM, however, generally yields poor results in the numerical simulation of convection-dominated flows, often in the form of numerical oscillations (see, e.g., [3,4,8,9,14,16–18]). In the ROM literature, a common explanation for this failure of standard ROMs is the violation of the concept of energy cascade [3,8,9,14,18,19]. Indeed, in [20] it was shown that the concept of energy cascade is also valid in a POD setting: The main role of the neglected POD modes is to drain energy from the ROM. Thus, since standard ROMs fail to account for this transfer of energy from resolved to unresolved POD modes, they generally yield inaccurate results.

Over the years, numerous models for the effect of the neglected POD modes have been devised (see, e.g., [8–10, 13,14,19,21,22]). Although a survey of these approaches is beyond the scope of this paper, some of the most recent developments can be found in, e.g., [9,13,14,23,24] and references therein. Many of these models are similar to the large eddy simulation (LES) closure models. Indeed, to model the effect of the neglected POD modes, some authors start with the filtered equations and propose closure models for ROMs, just as in LES (see, e.g., [3,8,20]). Other authors, however, do not employ filtering and the LES analogy to develop models for the neglected POD modes (see, e.g., [9,17–19]). For the sake of brevity, in what follows we will use *ROM closure model* as a generic term for models of neglected POD modes [3,9,17–20,23,24].

Since the analogy between ROM and LES closure modeling is relevant to the model that we propose in this paper, we include next a "time capsule" of LES closure modeling. Of course, we only include the LES developments that are relevant to our paper; for more details, we refer the reader to specialized texts, e.g., [25–28].

The idea in LES is straightforward. Instead of approximating the flow variables, the LES models aim at approximating the spatially filtered flow variables, which can be discretized on significantly coarser spatial meshes. However, since the underlying Navier–Stokes equations are nonlinear, filtering them yields a system of equations that is not closed. Closing the spatially filtered Navier–Stokes equations, which is commonly known as the *closure problem*, represents the main challenge faced by LES.

There are many closure models used in LES. Following Sagaut's terminology [28], these LES closure models could be divided into two categories: (i) functional closure models and (ii) structural closure models. Chronologically, the functional closure models were developed first and the structural closure models were put forth later.

The *functional* LES closure models (chapters 5 and 6 in [28]) follow from phenomenological arguments and aim at modeling the physical effect of the sub-filter scales (i.e., the scales below the spatial filter radius). The main tool used in developing these functional closure models is Kolmogorov's statistical theory of turbulence and the resulting energy cascade, which postulates that the main role of the sub-filter scales is to drain energy out of the LES model. In the functional closure models, this is generally achieved by adding an eddy viscosity to the molecular viscosity of the system. Probably the most popular example in this class is the Smagorinsky model [29].

The second type of closure models is *structural LES* closure models (chapter 7 in [28]). These closure models are generally derived through mathematical rather than phenomenological arguments, e.g., formal series expansions. One of the most popular models in this class is the approximate deconvolution (AD) model [30–32], which uses the deconvolution approaches developed in the image processing and inverse problems communities to recover the original signal from a blurred filtered signal.

The ROM closure model development was similar to the LES closure modeling development. Just as in LES, the first ROM closure models have been of the functional type. These functional ROM closure models have generally used some sort of stabilization procedure to model the effect of the discarded POD modes. Probably the first functional ROM closure model has been the mixing length model pioneered by Lumley and his group [8]. This model was followed by more accurate ROM closure models, such as the Smagorinsky model [33–35], the variational multiscale model [14,22] and the dynamic subgrid-scale model [14]. As already mentioned, a physical motivation for the approach used to develop these functional ROM closure models is the concept of energy cascade [20]. We emphasize that, just as in the LES case, these functional ROM closure models do not employ an explicit spatial filter; instead, they use phenomenological arguments to model the effect of the discarded POD modes.

To our knowledge, structural ROM closure models have not been proposed yet. Again, this is not surprising, since in LES the structural closure models were developed much later than the functional closure models. We believe that the main reason why structural ROM closure models have not been proposed is that the concept of *explicit spatial filtering* (which is central in the development of structural LES closure models) has not been exploited yet in the ROM field.

In this paper, we take a first step in this direction and propose a structural ROM closure model. To this end, we first outline an LES-ROM framework that employs explicit ROM spatial filtering to develop new ROMs in which the term that needs to be modeled (i.e., the ROM stress tensor) is explicitly defined. We emphasize that, in this LES-ROM framework, we carry out explicit spatial filtering at a ROM level, not at the input data level. Thus, the new LES-ROMs can use either DNS input data (as we do in this paper, see Sections 2 and 5) or LES input data (see discussion in Section 6). Within the LES-ROM framework, we use the AD method to put forth a new structural ROM closure model, the AD-ROM. Specifically, given the approximation of the filtered ROM variables, we use AD to obtain an approximation of the original unfiltered ROM variables and solve the ROM closure problem. Since the AD problem is notoriously ill-posed, we use a regularization method from image processing and inverse problems to obtain stable AD approximations.

Previous Relevant Work. The LES-ROM framework that we use in this paper has been also utilized in, e.g., [3,8,14]. We emphasize that, to our knowledge, the LES-ROM framework has only been used *implicitly*, to motivate various functional ROM closure models [3,8]. In this paper, we take a completely different approach and *explicitly* use the LES-ROM framework (specifically, the ROM stress tensor) to develop a structural ROM closure model, the AD-ROM.

In ROMs, spatial filtering has been used as a preprocessing step, to filter out the noise in the snapshot data, i.e., in the generation of the POD modes (see, e.g., Section 5 in [36] for a survey of relevant work). We emphasize, however, that our approach is *fundamentally different*, since we *explicitly* use spatial filters in the *actual ROMs*, i.e., we modify the mathematical model, not the input data. To our knowledge, explicit ROM spatial filtering has only be used in [14,37,38].

In this paper, we exclusively use ROM spatial filtering to develop the LES-ROM framework and the AD-ROM. A different filtering approach was used in [23]: Instead of ROM spatial filtering, the authors used *ROM ensemble averaging* to develop the ensemble averaged ROM, called the Galerkin–Reynolds equation (see Eq. (7) in [23]). The ensemble averaged ROM proposed in [23] has, in our opinion, great potential. We emphasize, however, that the LES-ROM that we propose in this paper and the ensemble averaged ROM proposed in [23] are *fundamentally different*: The LES-ROM employs the LES methodology to develop a dynamical system for the spatial averages of the ROM velocity. The ensemble averaged ROM, on the other hand, is reminiscent of RANS formalism and consists of a dynamical system for the ensemble average of the ROM coefficients.

The rest of the paper is organized as follows: In Section 2, we briefly discuss the POD and the standard ROM. In Section 3, we outline the LES-ROM framework. In Section 4, we use this LES-ROM framework to develop the new AD-ROM. In Section 5, we present numerical results for the new AD-ROM for a 3D flow past a circular cylinder at Reynolds number $Re = 1000$. Finally, in Section 6 we draw conclusions and present an outlook of future work.

2. Reduced order modeling

In this section, the POD, the standard Galerkin ROM and the centering trajectory are briefly presented. The *Navier–Stokes equations (NSE)* are used as mathematical model:

$$\boldsymbol{u}_t - \frac{2}{Re} \nabla \cdot \mathbb{D}(\boldsymbol{u}) + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \boldsymbol{0}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where $\boldsymbol{u}$ is the velocity, p the pressure, Re the Reynolds number and $\mathbb{D}(\boldsymbol{u}) := (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top)/2$ is the deformation tensor of $\boldsymbol{u}$. In this paper, the NSE (1)–(2) are supplemented with the initial condition $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$ and steady Dirichlet boundary conditions.

Remark 2.1 (*Viscous Term Formulation*)**.** In the NSE (1), we used the viscous term formulation

$$\frac{2}{Re} \nabla \cdot \mathbb{D}(\boldsymbol{u}) = \frac{1}{Re} \nabla \cdot \left[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top \right]. \tag{3}$$

When $\nabla \cdot \boldsymbol{u} = 0$, the viscous term formulation (3) is identical to the standard formulation

$$\frac{1}{Re} \nabla \cdot \nabla \boldsymbol{u} = \frac{1}{Re} \Delta \boldsymbol{u}. \tag{4}$$

As noted in Section 3.3.1 of [39], however, the finite element (FE) discretizations of the two formulations are usually not identical.

Both the continuous version and the FE discretization of the alternative viscous term formulation (3) are carefully discussed in Section 3.11.3, Section 3.12.1 and Section 3.13.1 of [39]. We only note that, since $\mathbb{D}(\boldsymbol{u}) = \mathbb{D}(\boldsymbol{u})^{\top}$, the following inequality (which is equation 3.11-11 of [39]) holds:

$$\left(2\,\mathbb{D}(\boldsymbol{u}),\, \nabla\boldsymbol{u}\right) = \left(\mathbb{D}(\boldsymbol{u}),\, \nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\top}\right) = 2\left(\mathbb{D}(\boldsymbol{u}),\, \mathbb{D}(\boldsymbol{u})\right) \geq 0. \tag{5}$$

Inequality (5) and Korn's inequality (see Theorem 3.77 in [40]) imply that the matrix corresponding to the FE discretization of (3) in $\boldsymbol{H}_0^1$ is positive definite.

In the remainder of the paper, we use the alternative viscous term formulation (3).

2.1. Proper orthogonal decomposition (POD)

One of the most popular reduced order modeling techniques is the POD, which we exclusively use to generate all the ROMs in this paper. We briefly describe POD next. For more details, the reader is referred to, e.g., [3,4,7]. The POD starts with the snapshots $\{\boldsymbol{u}_h^1, \ldots, \boldsymbol{u}_h^{N_s}\}$, which are numerical approximations of the NSE at N_s different time instances. For all the ROMs in this paper, we use DNS data to build the snapshots. Specifically, we utilize the FE approximations of the NSE as snapshots (see Section 5 for details). We emphasize, however, that other numerical methods can be used instead. The POD seeks a low-dimensional basis that approximates the snapshots optimally with respect to a certain norm. In this paper, we choose the commonly used L^2-norm. The solution of the minimization problem is equivalent to the solution of the eigenvalue problem

$$\boldsymbol{Y}\,\boldsymbol{Y}^{\top}\boldsymbol{M}_h\,\boldsymbol{\varphi}_j = \lambda_j\,\boldsymbol{\varphi}_j, \quad j = 1, \ldots, N, \tag{6}$$

where $\boldsymbol{\varphi}_j$ and λ_j denote the vector of the FE coefficients of the POD basis functions and the POD eigenvalues, respectively, $\boldsymbol{Y}$ denotes the snapshot matrix, whose columns correspond to the FE coefficients of the snapshots, $\boldsymbol{M}_h$ denotes the FE mass matrix, and N is the dimension of the FE space $\boldsymbol{X}^h$ [41]. The eigenvalues are real and non-negative, so they can be ordered as follows:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0. \tag{7}$$

The POD basis consists of the normalized functions $\{\boldsymbol{\varphi}_j\}_{j=1}^r$, which correspond to the first $r \ll N$ largest eigenvalues. Thus, the POD space is defined as $\boldsymbol{X}^r := \mathrm{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$.

The centering trajectory method is popular in ROM development [3]. In this approach, the snapshots $\{\boldsymbol{u}_h^1, \ldots, \boldsymbol{u}_h^{N_s}\}$ are replaced by $\{\boldsymbol{u}_h^1 - \boldsymbol{U}, \ldots, \boldsymbol{u}_h^{N_s} - \boldsymbol{U}\}$, where $\boldsymbol{U} = \frac{1}{N_s}\sum_{i=1}^{N_s}\boldsymbol{u}_h^i$ is the centering trajectory and $\boldsymbol{u}_h^i - \boldsymbol{U}$ are the snapshot fluctuations. Thus, the POD basis functions are computed from the snapshot fluctuations $\boldsymbol{u}_h^i - \boldsymbol{U},\ i = 1, \ldots, N_s$. The centering trajectory method is especially useful for problems that have steady nonhomogeneous Dirichlet boundary conditions, such as the boundary conditions in the 3D flow past a cylinder investigated in Section 5. Since in this case the snapshot fluctuations $\boldsymbol{u}_h^i - \boldsymbol{U},\ i = 1, \ldots, N_s$ satisfy homogeneous Dirichlet boundary conditions, the centering trajectory method avoids the challenges posed by the nonhomogeneous boundary conditions in ROMs (see, however, [42] for alternative approaches). In what follows, we will use the centering trajectory approach. We emphasize, however, that the new AD-ROM that we develop in what follows works equally well when the centering trajectory is not used.

2.2. The Galerkin ROM (G-ROM)

To develop the standard Galerkin ROM, we start by considering the ROM velocity approximation, $\boldsymbol{U}_r$, defined as follows:

$$\boldsymbol{U}_r(\boldsymbol{x}, t) = \boldsymbol{U}(\boldsymbol{x}) + \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x}) + \boldsymbol{u}_r(\boldsymbol{x}, t), \tag{8}$$

where $\{a_j(t)\}_{j=1}^{r}$ are the sought time-varying POD coefficients, $\boldsymbol{U}$ is the centering trajectory defined in Section 2.1, and $\boldsymbol{u}_r(\boldsymbol{x}, t) = \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x})$ is the ROM velocity fluctuation. We use the POD basis, the fact that the POD modes are weakly divergence-free (just as the snapshots), and a Galerkin approximation of the NSE to obtain the standard *Galerkin ROM (G-ROM)* [3]: $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \boldsymbol{U}_r}{\partial t}, \boldsymbol{\varphi}_i\right) + \frac{2}{Re}\left(\mathbb{D}(\boldsymbol{U}_r), \nabla\boldsymbol{\varphi}_i\right) + \left(\boldsymbol{U}_r \cdot \nabla\boldsymbol{U}_r, \boldsymbol{\varphi}_i\right) = 0. \tag{9}$$

The G-ROM (9) yields the following autonomous dynamical system for the vector of time coefficients, $\mathbf{a}(t)$:

$$\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A}\,\mathbf{a} + \mathbf{a}^{\top}\,\mathbf{B}\,\mathbf{a}, \tag{10}$$

where $\mathbf{b}$, $\mathbf{A}$, and $\mathbf{B}$ correspond to the constant, linear, and quadratic terms in the numerical discretization of the NSE (1)–(2), respectively. The initial conditions are obtained by projection:

$$a_j(0) = \left(\boldsymbol{u}_0 - \mathbf{U}, \boldsymbol{\varphi}_j\right), \quad j = 1, \ldots, r. \tag{11}$$

The finite dimensional system (10) can be written componentwise as follows: For all $i = 1, \ldots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} A_{im} a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}\, a_m(t)\, a_n(t), \tag{12}$$

where

$$b_i = -\left(\mathbf{U} \cdot \nabla\mathbf{U}, \boldsymbol{\varphi}_i\right) - \frac{2}{Re}\left(\frac{\nabla\mathbf{U} + \nabla\mathbf{U}^{\top}}{2}, \nabla\boldsymbol{\varphi}_i\right), \tag{13}$$

$$A_{im} = -\left(\mathbf{U} \cdot \nabla\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_i\right) - \left(\boldsymbol{\varphi}_m \cdot \nabla\mathbf{U}, \boldsymbol{\varphi}_i\right) - \frac{2}{Re}\left(\frac{\nabla\boldsymbol{\varphi}_m + \nabla\boldsymbol{\varphi}_m{}^{\top}}{2}, \nabla\boldsymbol{\varphi}_i\right), \tag{14}$$

$$B_{imn} = -\left(\boldsymbol{\varphi}_m \cdot \nabla\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_i\right). \tag{15}$$

3. Large eddy simulation ROM (LES-ROM) framework

Numerous ROM closure modeling approaches have been proposed to address the inaccuracy (and numerical instability) of the standard G-ROM (see, e.g., [4,13,14] and references therein). In this paper, we take a different approach and develop an LES-ROM framework that is centered around explicit ROM spatial filtering.

In Section 3.1, we present a continuous ROM spatial filter and two discrete ROM spatial filters. In Section 3.2, we use this explicit ROM spatial filtering to develop the LES-ROM framework.

3.1. Explicit ROM spatial filtering

To develop the new LES-ROM framework, we use the explicit *differential filter (DF)*. We present the continuous formulation of the DF in Section 3.1.1, and two of its discrete formulations in Section 3.1.2 and Section 3.1.3.

The DF has been pioneered in LES by Germano [43,44]. Later, it was also used in a ROM context to develop regularized ROMs: The ROM-DF was first used in [37] for the 1D Kuramoto–Sivashinsky equation in a periodic setting. The ROM-DF was subsequently used in [38] for the 3D NSE in a general non-periodic setting.

3.1.1. Continuous differential filter (C-DF)

The *continuous differential filter (C-DF)* has been extensively used to develop LES models [45]. It is defined as follows (see, e.g., Section 2.2.6 in [45]): Let δ be the radius of the C-DF. For all $\boldsymbol{\Phi} \in \boldsymbol{L}^2(\Omega)$, find $\overline{\boldsymbol{\Phi}} \in \boldsymbol{L}^2(\Omega)$ such that

$$-\delta^2\, \Delta\overline{\boldsymbol{\Phi}} + \overline{\boldsymbol{\Phi}} = \boldsymbol{\Phi} \qquad \text{in } \Omega \tag{16}$$

$$\overline{\boldsymbol{\Phi}} = \mathbf{0} \qquad \text{on } \partial\Omega. \tag{17}$$

The C-DF (16)–(17) is a linear spatial filter, which employs an elliptic operator to smooth the input variables and eliminate the small scales (i.e., high frequencies) [25].

The following lemma was proved in [46]:

Lemma 3.1. *The C-DF (16)–(17) is a self-adjoint operator on* $\boldsymbol{L}^2(\Omega)$.

Proof. Consider $\boldsymbol{\Phi}, \boldsymbol{\Psi} \in \boldsymbol{L}^2(\Omega)$. By the definition of the C-DF, we have

$$-\delta^2 \, \Delta \overline{\boldsymbol{\Psi}} + \overline{\boldsymbol{\Psi}} = \boldsymbol{\Psi} \qquad \text{in } \Omega \tag{18}$$

$$\overline{\boldsymbol{\Psi}} = \boldsymbol{0} \qquad \text{on } \partial\Omega. \tag{19}$$

Multiplying (18) by $\overline{\boldsymbol{\Phi}}$, integrating by parts, and using (17), we obtain

$$\delta^2 \left(\nabla \overline{\boldsymbol{\Psi}}, \nabla \overline{\boldsymbol{\Phi}} \right) + \left(\overline{\boldsymbol{\Psi}}, \overline{\boldsymbol{\Phi}} \right) = \left(\boldsymbol{\Psi}, \overline{\boldsymbol{\Phi}} \right) \qquad \text{in } \Omega. \tag{20}$$

Multiplying (16) by $\overline{\boldsymbol{\Psi}}$, integrating by parts, and using (19), we obtain

$$\delta^2 \left(\nabla \overline{\boldsymbol{\Phi}}, \nabla \overline{\boldsymbol{\Psi}} \right) + \left(\overline{\boldsymbol{\Phi}}, \overline{\boldsymbol{\Psi}} \right) = \left(\boldsymbol{\Phi}, \overline{\boldsymbol{\Psi}} \right) \qquad \text{in } \Omega. \tag{21}$$

Eqs. (20) and (21) prove the lemma. □

3.1.2. FE differential filter (FE-DF)

The *FE differential filter (FE-DF)* is defined as follows [25,45]: Let δ be the radius of the FE-DF. For all $\boldsymbol{u}^h \in \boldsymbol{X}^h$ (where $\boldsymbol{X}^h$ is the FE space), find $\overline{\boldsymbol{u}^h}^h \in \boldsymbol{X}^h$ such that

$$-\delta^2 \, \Delta \overline{\boldsymbol{u}^h}^h + \overline{\boldsymbol{u}^h}^h = \boldsymbol{u}^h \qquad \text{in } \Omega \tag{22}$$

$$\overline{\boldsymbol{u}^h}^h = \boldsymbol{0} \qquad \text{on } \partial\Omega, \tag{23}$$

which yields the following linear system:

$$\left(\boldsymbol{M}_h + \delta^2 \, \boldsymbol{S}_h \right) \overline{\boldsymbol{U}^h}^h = \boldsymbol{M}_h \, \boldsymbol{U}^h, \tag{24}$$

where $\boldsymbol{M}_h \in \mathbb{R}^{N \times N}$ is the FE mass matrix, $\boldsymbol{S}_h \in \mathbb{R}^{N \times N}$ is the FE stiffness matrix, $\overline{\boldsymbol{U}^h}^h \in \mathbb{R}^N$ is the vector of FE coefficients of the output filtered variable $\overline{\boldsymbol{u}^h}^h$, and $\boldsymbol{U}^h \in \mathbb{R}^N$ is the vector of FE coefficients of the input variable $\boldsymbol{u}^h$.

3.1.3. ROM differential filter (ROM-DF)

The *ROM differential filter (ROM-DF)* is defined as follows: Let δ be the radius of the ROM-DF. For all $\boldsymbol{u}^r \in \mathbf{X}^r$, find $\overline{\boldsymbol{u}^r}^r \in \boldsymbol{X}^r$ such that

$$-\delta^2 \, \Delta \overline{\boldsymbol{u}^r}^r + \overline{\boldsymbol{u}^r}^r = \boldsymbol{u}^r \qquad \text{in } \Omega \tag{25}$$

$$\overline{\boldsymbol{u}^r}^r = \boldsymbol{0} \qquad \text{on } \partial\Omega, \tag{26}$$

which yields the following linear system:

$$\left(\boldsymbol{M}_r + \delta^2 \, \boldsymbol{S}_r \right) \overline{\boldsymbol{a}^r}^r = \boldsymbol{M}_r \boldsymbol{a}^r, \tag{27}$$

where $\boldsymbol{M}_r \in \mathbb{R}^{r \times r}$ is the POD mass matrix with entries $(\boldsymbol{M}_r)_{ij} = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i)$, $\boldsymbol{S}_r \in \mathbb{R}^{r \times r}$ is the POD stiffness matrix with entries $(\boldsymbol{S}_r)_{ij} = (\nabla \boldsymbol{\varphi}_j, \nabla \boldsymbol{\varphi}_i)$, $\overline{\boldsymbol{a}^r}^r \in \mathbb{R}^r$ is the vector of POD coefficients of the output filtered variable $\overline{\boldsymbol{u}^r}^r$, and $\boldsymbol{a}^r \in \mathbb{R}^r$ is the vector of POD coefficients of the input variable $\boldsymbol{u}^r$.

We emphasize that the computational cost of the ROM-DF is much lower than the computational cost of the FE-DF, since the former yields a very small $r \times r$ linear system, whereas the latter yields a much larger, $N \times N$ linear system.

3.2. Large eddy simulation ROM (LES-ROM)

To develop the LES-ROM, we use the standard LES approach [25–28,47]:

1. Filter the NSE to obtain the *spatially filtered NSE (SF-NSE)*.
2. Use the SF-NSE and the ROM approximation to obtain the LES-ROM.

The motivation for using the above strategy in the LES field is straightforward: The SF-NSE are a system of equations for the filtered flow variables, which can be approximated on the given coarse mesh more accurately than the original, unfiltered flow variables. In this paper, we use a similar argument to develop the LES-ROM framework: Since the spatial structures in the SF-NSE are larger than the spatial structures in the NSE, we expect that, for a fixed target numerical accuracy of the ROM, the SF-NSE require fewer POD modes than the NSE. This, of course, is advantageous from a computational point of view in challenging, realistic applications.

In the remainder of this section, we first present the derivation of the SF-NSE and the associated closure problem. Then, we outline the general LES-ROM framework. In Section 4, we use this LES-ROM framework to develop the new AD-ROM.

We filter the NSE (1) with the continuous ROM differential filter, C-DF, and assume that differentiation and filtering commute (see, e.g., Chapter 9 in [25]):

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} - \frac{2}{Re} \nabla \cdot \mathbb{D}(\overline{\boldsymbol{u}}) + \overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}} + \nabla \overline{p} = \boldsymbol{0}. \tag{28}$$

Next, we project (28) onto a space of weakly divergence-free functions $\boldsymbol{\phi}$ to obtain the weak formulation of the *spatially filtered NSE (SF-NSE)*:

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi} \right) + \frac{2}{Re} \left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi} \right) + \left(\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}, \boldsymbol{\phi} \right) = \boldsymbol{0}. \tag{29}$$

Remark 3.1. The standard SF-NSE generally include a pressure term. Since we used weakly divergence-free test functions $\boldsymbol{\phi}$, the pressure term from (28) dropped out of (29). We used this approach since the POD basis functions employed in our ROMs are weakly divergence-free. Thus, the SF-NSE (29), although nonstandard, are better suited to our ROM framework.

Of course, in LES one cannot simply discretize the SF-NSE (29), since these equations are not closed. Thus, one first needs to address the *closure problem*, i.e., to model the filtered nonlinearity $\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}$ in terms of the filtered velocity $\overline{\boldsymbol{u}}$:

$$\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}} = \text{function}(\overline{\boldsymbol{u}}). \tag{30}$$

The *closure model* is the explicit dependence on $\overline{\boldsymbol{u}}$ of the filtered nonlinearity $\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}$ in Eq. (30). Once the closure problem is addressed, the SF-NSE (29) yield the *LES model*:

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi} \right) + \frac{2}{Re} \left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi} \right) + \left(\text{function}(\overline{\boldsymbol{u}}), \boldsymbol{\phi} \right) = \boldsymbol{0}. \tag{31}$$

Of course, the LES model (31) needs to be discretized. We use a ROM approximation and obtain the *LES-ROM*: $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \boldsymbol{W}_r}{\partial t}, \boldsymbol{\varphi}_i \right) + \frac{2}{Re} \left(\mathbb{D}(\boldsymbol{W}_r), \nabla \boldsymbol{\varphi}_i \right) + \left(\text{function}(\boldsymbol{W}_r), \boldsymbol{\varphi}_i \right) = \boldsymbol{0}, \tag{32}$$

where $\boldsymbol{W}_r$ is the ROM approximation of the filtered velocity $\overline{\boldsymbol{u}}$ in (31).

Remark 3.2. In the LES literature, the ROM closure problem generally takes a form that is different from (30). In this paper, however, we use the equivalent form in (30), since just as in LES [25,28,45], it makes the derivation of the AD-ROM in Section 4 straightforward.

For completeness, however, we outline the derivation of the standard formulation of the LES closure problem. First, in the NSE, we use the fact that $\nabla \cdot \boldsymbol{u} = 0$ to rewrite the nonlinear term as follows: $\boldsymbol{u} \cdot \nabla \boldsymbol{u} = \nabla \cdot (\boldsymbol{u}\,\boldsymbol{u})$. Thus, the SF-NSE (29) can be written as

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi} \right) + \frac{2}{Re} \left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi} \right) + \left(\overline{\nabla \cdot (\boldsymbol{u}\,\boldsymbol{u})}, \boldsymbol{\phi} \right) = \mathbf{0}. \tag{33}$$

Next, we decompose the nonlinearity in (33) as follows:

$$\overline{\boldsymbol{u}\boldsymbol{u}} = \overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}} + \left(\overline{\boldsymbol{u}\,\boldsymbol{u}} - \overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}} \right) = \overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}} + \boldsymbol{\tau}, \tag{34}$$

where, following the LES terminology [25,28], the stress tensor

$$\boldsymbol{\tau} = \overline{\boldsymbol{u}\boldsymbol{u}} - \overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}} \tag{35}$$

is the *subfilter-scale stress tensor*. Thus, the SF-NSE (33) take the form

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi} \right) + \frac{2}{Re} \left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi} \right) + \left(\nabla \cdot (\overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}}), \boldsymbol{\phi} \right) + \left(\nabla \cdot \boldsymbol{\tau}, \boldsymbol{\phi} \right) = \mathbf{0}. \tag{36}$$

We emphasize that Eq. (36) can be regarded as the standard weak formulation for the NSE for $\overline{\boldsymbol{u}}$ (not $\boldsymbol{u}$), plus the extra term $\left(\nabla \cdot \boldsymbol{\tau}, \boldsymbol{\phi} \right)$. The closure problem for the SF-NSE (36) is

$$\boldsymbol{\tau} = \text{function}(\overline{\boldsymbol{u}}). \tag{37}$$

Eq. (37) is the form generally used in the LES literature for the closure problem.

4. Approximate deconvolution ROM (AD-ROM)

Most ROM closure models have generally used some sort of stabilization procedure, e.g., [8–10,13,14,18,19,21–23,48]. A physical motivation for this popular approach is given in [20], where it is shown that the concept of energy cascade is also valid in a POD setting. Thus, the role of the ROM closure model is to drain energy out of the system.

In this paper, we use a fundamentally different, novel ROM closure modeling strategy that centers around the concept of *approximate deconvolution (AD)*. In LES, the AD models have been pioneered in a series of papers by Adams and Stolz [30–32,49]. These AD models are based on the *deconvolution* idea, which is central in *image processing* and *inverse problems* [50–52]: Given an approximation of the filtered input signal, find an approximation of the input itself. In LES, the deconvolution is used as follows: Given the available approximation of the filtered velocity, the deconvolution is used to obtain an approximation of the unfiltered velocity, which is then used to solve the LES closure problem.

In this paper, we use the LES-ROM framework developed in Section 3.2 to construct a novel AD-ROM. In Section 4.1, we present the deconvolution procedure. In Section 4.2, we use this deconvolution to develop the new AD-ROM.

4.1. Approximate deconvolution

For clarity of presentation, let us denote with G the explicit ROM spatial filter used to develop the LES-ROM framework in Section 3.2. We assume that an approximation of the filtered velocity

$$\overline{\boldsymbol{u}} := G\,\boldsymbol{u} \tag{38}$$

is available. The goal in the ROM deconvolution problem is to find an approximation of the original flow variable, $\boldsymbol{u}$, so that the ROM closure problem (30) can be solved.

Since G is invertible, at first glance, one just has to use an *exact deconvolution (ED)*, i.e., employ the inverse of the filter G in (38) to solve the deconvolution problem:

$$\boldsymbol{u}^{ED} = G^{-1}\,\overline{\boldsymbol{u}} = \boldsymbol{u}. \tag{39}$$

It is well-known in the inverse problem community that the ED is generally a bad idea because the inverse problem (38) is *ill-posed*: small changes in the data can lead to large changes in the solution [45,50,52]. Indeed, inverting the operator G in (39) results in division by small coefficients of the high-frequency components of the operator G. Thus, any changes in the input (38) translate into large unphysical oscillations in the output (39) [50,52]. The ED (39) was carefully investigated in Section 5.2 of [53], where it was shown that, as expected, for input signals ($\boldsymbol{u}$) of different types, adding noise to the filtered signal ($\overline{\boldsymbol{u}}$) and then using the ED (39) results in inaccurate approximations ($\boldsymbol{u}^{ED}$), usually in the form of numerical oscillations.

To avoid the numerical oscillations in the ED (39), we use one of the *regularized deconvolution* methods developed in the image processing and inverse problems communities [50–52]. Following the LES terminology, we call this method *approximate deconvolution (AD)*. Since this paper is the first investigation of the new AD-ROM, we employ one of the simplest regularized deconvolutions used in the image processing and inverse problems communities, the *Lavrentiev regularization*:

$$\boldsymbol{u}^{AD} = \Big(G + \mu\, I \Big)^{-1} \overline{\boldsymbol{u}}, \tag{40}$$

where I is the identity operator and μ is a regularization parameter.

The AD (40) was carefully investigated in Section 5.4 in [53], where it was shown that it performs much better than the ED (39): For input signals of different types, noise was added to the filtered signal ($\overline{\boldsymbol{u}}$), and the ED and AD were compared. The AD (40) produced much better results than the ED (39): The latter increased the amplitude of the noise added to the filtered input signal, which, in turn, yielded spurious numerical oscillations. The former, on the other hand, significantly decreased the amplitude of the noise and yielded relatively accurate results.

Remark 4.1 (*AD vs ED*)**.** We emphasize that these results are not surprising at all. Indeed, they reflect the reality in the inverse problems and image processing communities, where it is well known that the ED produces inaccurate results, and the AD should be used instead. However, both the AD and ED were carefully investigated in [53], since, to our knowledge, this was the first investigation of deconvolution in the ROM field.

4.2. *Approximate deconvolution ROM (AD-ROM)*

The AD approximation (40) is used to close the LES model (31): $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\varphi}_i \right) + \frac{2}{Re} \Big(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\varphi}_i \Big) + \Big(\overline{\boldsymbol{u}^{AD} \cdot \nabla \boldsymbol{u}^{AD}}, \boldsymbol{\varphi}_i \Big) = \boldsymbol{0}. \tag{41}$$

At first glance, to obtain an LES-ROM, one would simply replace the filtered velocity $\overline{\boldsymbol{u}}$ in (41) with a ROM approximation

$$\boldsymbol{W}_r = \boldsymbol{U} + \sum_{j=1}^{r} a_j\, \boldsymbol{\varphi}_j = \boldsymbol{U} + \boldsymbol{w}_r. \tag{42}$$

Remark 4.2. Note that the coefficients a_j in the $\boldsymbol{w}_r$ expansion (42) are different from the coefficients a_j used in the $\boldsymbol{u}_r$ expansion (8). We use the same notation, however, to ensure consistency with the traditional ROM notation.

We emphasize that approximation (42) is not appropriate. Indeed, since the POD basis functions were obtained from the numerical discretization of the *unfiltered* NSE, they can only approximate the *unfiltered* flow velocity, $\boldsymbol{u}$:

$$\boldsymbol{u} \approx \boldsymbol{U}_r = \boldsymbol{U} + \boldsymbol{u}_r. \tag{43}$$

Thus, the *filtered* flow velocity in (41) should be approximated as

$$\overline{\boldsymbol{u}} \approx \overline{\boldsymbol{U}_r} \overset{(43)}{=} \overline{\boldsymbol{U}} + \overline{\boldsymbol{u}_r}. \tag{44}$$

Furthermore, to be able to approximate $\overline{\boldsymbol{u}}$ in the ROM space $\mathbf{X}^r$, we make the following ansatz:

$$\overline{\boldsymbol{u}_r} \approx \boldsymbol{w}_r = \sum_{j=1}^{r} a_j \, \boldsymbol{\varphi}_j . \tag{45}$$

Thus, the filtered velocity $\overline{\boldsymbol{u}}$ in (41) should be replaced with

$$\boldsymbol{W}_r = \overline{\boldsymbol{U}} + \overline{\boldsymbol{u}_r} \overset{(45)}{\approx} \overline{\boldsymbol{U}} + \boldsymbol{w}_r . \tag{46}$$

We emphasize that, although approximations (42) and (46) are similar in that they both use ansatz (45), they are different in one important aspect: Approximation (42) *assumes* that the *filtered* centering trajectory $\overline{\boldsymbol{U}}$ can be approximated by the unfiltered centering trajectory $\boldsymbol{U}$ and the POD basis, whereas approximation (46) does not make this assumption.

Next, we present the ROM approximation of $\boldsymbol{u}^{AD}$ in (41):

$$\begin{aligned} \boldsymbol{u}^{AD} &\approx G^{-1}\,\overline{\boldsymbol{u}} \overset{(43)}{\approx} G^{-1}\,\overline{\boldsymbol{U}_r} \overset{(45)}{\approx} G^{-1}\left(\overline{\boldsymbol{U}} + \boldsymbol{w}_r\right) = \boldsymbol{U} + G^{-1}\,\boldsymbol{w}_r \\ &\overset{(40)}{\approx} \boldsymbol{U} + \left(G + \mu\, I\right)^{-1} \boldsymbol{w}_r . \\ &= \boldsymbol{U} + \boldsymbol{w}_r^{AD}, \end{aligned} \tag{47}$$

where $\boldsymbol{w}_r^{AD}$ is a ROM approximation of the unfiltered flow velocity.

Using the ROM approximations (46) and (47) in the LES model (41), we obtain the *approximate deconvolution ROM (AD-ROM)*:

$$\left(\frac{\partial(\overline{\boldsymbol{U}} + \boldsymbol{w}_r)}{\partial t}, \boldsymbol{\varphi}_i\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{U}} + \boldsymbol{w}_r), \nabla\boldsymbol{\varphi}_i\right) + \left(\overline{(\boldsymbol{U} + \boldsymbol{w}_r^{AD}) \cdot \nabla(\boldsymbol{U} + \boldsymbol{w}_r^{AD})}, \boldsymbol{\varphi}_i\right) = \mathbf{0}. \tag{48}$$

Furthermore, since the C-DF is self-adjoint (see Lemma 3.1), the AD-ROM (48) can be written in the following equivalent form:

$$\begin{aligned} &\left(\cancelto{0}{\frac{\partial\overline{\boldsymbol{U}}}{\partial t}} + \frac{\partial \boldsymbol{w}_r}{\partial t}, \boldsymbol{\varphi}_i\right) + \frac{2}{Re}\left(\mathbb{D}(\boldsymbol{U}), \nabla\overline{\boldsymbol{\varphi}_i}\right) + \frac{2}{Re}\left(\mathbb{D}(\boldsymbol{w}_r), \nabla\boldsymbol{\varphi}_i\right) \\ &+ \left(\boldsymbol{U}\cdot\nabla\boldsymbol{U}, \overline{\boldsymbol{\varphi}_i}\right) + \left(\boldsymbol{w}_r^{AD}\cdot\nabla\boldsymbol{U}, \overline{\boldsymbol{\varphi}_i}\right) + \left(\boldsymbol{U}\cdot\nabla\boldsymbol{w}_r^{AD}, \overline{\boldsymbol{\varphi}_i}\right) + \left(\boldsymbol{w}_r^{AD}\cdot\nabla\boldsymbol{w}_r^{AD}, \overline{\boldsymbol{\varphi}_i}\right) = \mathbf{0}. \end{aligned} \tag{49}$$

We note that the alternative AD-ROM formulation (49) is better suited for discretization than the standard AD-ROM formulation (48), since it avoids the spatial filtering of the vectors and tensors involving the centering trajectory ($\boldsymbol{U}$), which can be challenging in a ROM setting. Thus, in the numerical investigation in Section 5, we will use the alternative AD-ROM formulation (49).

Remark 4.3. At first glance, the approach used in the derivation of the alternative formulation of the AD-ROM in (49) (i.e., using the self-adjointness of the C-DF) is reminiscent of other approaches used in LES. For example, using test functions that are multiplied by the inverse of the C-DF and then using the self-adjointness of the C-DF was essential in the proof of the well-posedness of the Rational LES model (see discussion on page 161 in [25]). The same approach was later used to develop a new type of AD-LES models in [46,54,55]. We emphasize, however, that there is a fundamental difference between the AD-ROM (49) and the AD-LES in [46,54,55]: To develop the alternative AD-ROM formulation (49), we first multiplied by the standard test function and then we used the self-adjointness of the C-DF to move the filtering operator from the terms involving the centering trajectory to the test function. We note that the filtered test functions that appear in (49) do not constitute a challenge in our ROM setting, since they can be precomputed and stored. However, for the discretization of the AD-LES models in [46,54,55] with standard numerical methods (e.g., FE), storing all the filtered basis functions is not possible in realistic settings with a large number of degrees of freedom. Thus, for the AD-LES models in [46,54,55], one has first to use test functions that are multiplied by the inverse of the C-DF and only then use the self-adjointness of the C-DF. The price paid, however, is that fourth-order differential operators need to be discretized in these AD-LES models [46,55].

4.3. Spatial filtering for the AD-ROM

To use the AD-ROM (49) in actual numerical simulations, the continuous C-DF (16)–(17) needs to be replaced by one of its two discrete variants: the FE-DF (22)–(23) or the ROM-DF (25)–(26).

4.3.1. ROM-DF spatial filtering

With this choice, the AD-ROM (49) can be written as the following dynamical system:

$$\dot{\boldsymbol{a}} = \boldsymbol{b} + \boldsymbol{A}\,\boldsymbol{a} + \boldsymbol{A}^{AD}\,\boldsymbol{a}^{AD} + (\boldsymbol{a}^{AD})^{\top}\,\boldsymbol{B}^{AD}\,\boldsymbol{a}^{AD}, \tag{50}$$

which can be written componentwise as follows: For all $i = 1, \dots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} A_{im}\, a_m(t) + \sum_{m=1}^{r} A_{im}^{AD}\, a_m^{AD}(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}^{AD}\, a_m^{AD}(t)\, a_n^{AD}(t), \tag{51}$$

where

$$b_i = -\big(\mathbf{U}\cdot\nabla\mathbf{U}, \overline{\boldsymbol{\varphi}_i}^{r}\big) - \frac{2}{Re}\left(\frac{\nabla\mathbf{U}+\nabla\mathbf{U}^{\top}}{2}, \nabla\overline{\boldsymbol{\varphi}_i}^{r}\right), \tag{52}$$

$$A_{im} = -\frac{2}{Re}\left(\frac{\nabla\boldsymbol{\varphi}_m + \nabla\boldsymbol{\varphi}_m{}^{\top}}{2}, \nabla\boldsymbol{\varphi}_i\right), \tag{53}$$

$$A_{im}^{AD} = -\big(\mathbf{U}\cdot\nabla\boldsymbol{\varphi}_m, \overline{\boldsymbol{\varphi}_i}^{r}\big) - \big(\boldsymbol{\varphi}_m\cdot\nabla\mathbf{U}, \overline{\boldsymbol{\varphi}_i}^{r}\big), \tag{54}$$

$$B_{imn}^{AD} = -\big(\boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{\varphi}_n, \overline{\boldsymbol{\varphi}_i}^{r}\big), \tag{55}$$

and the vector $\boldsymbol{a}^{AD} = \{a_m^{AD}\}_{m=1}^{r}$ in (51) is the solution of the following linear system:

$$\Big((1+\mu)\,\boldsymbol{M}_r + \delta^2\,\boldsymbol{S}_r\Big)\,\boldsymbol{a}^{AD} = \Big(\boldsymbol{M}_r + \delta^2\,\boldsymbol{S}_r\Big)\,\boldsymbol{a}. \tag{56}$$

4.3.2. FE-DF spatial filtering

To our knowledge, the only investigation of the FE-DF in a ROM setting has been carried out in [56]. In this paper, we outline only the FE-DF features that are relevant to the AD-ROM (49). For more details, we refer the reader to [56].

ROM filtering with the FE-DF relies on the fact that the ROM space ($\boldsymbol{X}^r$) is a subspace of the FE space ($\boldsymbol{X}^h$). Indeed, each POD basis function $\boldsymbol{\varphi}_i,\ i = 1, \dots, r$ has a unique FE representation:

$$\boldsymbol{\varphi}_i = \sum_{j=1}^{N} c_{ji}\,\boldsymbol{h}_j, \tag{57}$$

where $\{\boldsymbol{h}_j\}_{j=1}^{N}$ are the FE basis functions. (We note in passing that it is precisely representation (57) that allows the offline stage assembly of the vectors, matrices and tensors in standard ROMs.)

The FE-DF is used in the AD-ROM (49) in two instances: To compute $\overline{\boldsymbol{\varphi}_i}^{h}$ and $\boldsymbol{w}_r^{AD}$. Next, we briefly outline the approach used to compute these terms.

To compute $\overline{\boldsymbol{\varphi}_i}^{h}$, we use the representation (57) for both $\boldsymbol{\varphi}_i$ and $\overline{\boldsymbol{\varphi}_i}^{h}$:

$$\boldsymbol{\varphi}_i = \sum_{j=1}^{N} c_{ji}\,\boldsymbol{h}_j \qquad \text{and} \qquad \overline{\boldsymbol{\varphi}_i}^{h} = \sum_{j=1}^{N} \overline{c}_{ji}\,\boldsymbol{h}_j. \tag{58}$$

Thus, the equation $\overline{\boldsymbol{\varphi}_i}^h \overset{(22)}{=} (I - \delta^2 \Delta)^{-1} \boldsymbol{\varphi}_i$ yields the following linear systems: $\forall\, i = 1, \ldots, r$,

$$\left(\boldsymbol{M}_h + \delta^2 \boldsymbol{S}_h\right) \overline{\boldsymbol{c}}_i = \boldsymbol{M}_h \boldsymbol{c}_i , \tag{59}$$

where $\boldsymbol{M}_h$ and $\boldsymbol{S}_h$ are the FE mass and stiffness matrices defined in Section 3.1.2, and $\overline{\boldsymbol{c}}_i = \{\overline{c}_{ji}\}_{j=1}^N$ and $\boldsymbol{c}_i = \{c_{ji}\}_{j=1}^N$ are the vectors of coefficients in (58), which need to be computed and stored in the offline stage to be used in the AD-ROM in the online stage.

To compute $\boldsymbol{w}_r^{AD}$, we need to solve the following equation:

$$\boldsymbol{w}_r^{AD} \overset{(22),\,(40)}{=} \left((I - \delta^2 \Delta)^{-1} + \mu\, I \right)^{-1} \boldsymbol{w}_r , \tag{60}$$

which is equivalent to

$$\begin{aligned} \boldsymbol{w}_r^{AD} + \mu\, (I - \delta^2 \Delta)\, \boldsymbol{w}_r^{AD} \quad &= \quad (I - \delta^2 \Delta)\, \boldsymbol{w}_r \\ &\overset{(45),\,(57)}{=} \sum_{j=1}^{r} a_j \sum_{i=1}^{N} c_{ij}\, (I - \delta^2 \Delta)\, \boldsymbol{h}_i . \end{aligned} \tag{61}$$

We emphasize that (61) needs to be solved in $\boldsymbol{X}^h$, since the FE-DF is defined in $\boldsymbol{X}^h$. There are two natural approaches to solve (61):

(i) In the first approach, we simply let

$$\boldsymbol{w}_r^{AD} = \sum_{i=1}^{N} \alpha_i\, \boldsymbol{h}_i \tag{62}$$

in (61), which yields the following $N \times N$ linear system:

$$\left((1 + \mu)\, \boldsymbol{M}_h + \mu\, \delta^2\, \boldsymbol{S}_h \right) \boldsymbol{\alpha} = \sum_{j=1}^{r} a_j \left((\boldsymbol{M}_h + \delta^2\, \boldsymbol{S}_h)\, \boldsymbol{c}_j \right), \tag{63}$$

where $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^N$ is the vector of coefficients in (62).

(ii) In the second approach for solving (61), we let

$$\boldsymbol{w}_r^{AD} = \sum_{j=1}^{r} a_j \sum_{i=1}^{N} c_{ij}^{AD} \boldsymbol{h}_i = \sum_{j=1}^{r} a_j\, \boldsymbol{\varphi}_j^{AD} , \tag{64}$$

where $\boldsymbol{\varphi}_j^{AD}$ is defined as

$$\boldsymbol{\varphi}_j^{AD} = \sum_{i=1}^{N} c_{ij}^{AD} \boldsymbol{h}_i . \tag{65}$$

We emphasize that $\boldsymbol{\varphi}_j^{AD}$ belongs to the FE space $\mathbf{X}^h$, but does not necessarily belong to the ROM space $\mathbf{X}^r$. To solve (61), we simply solve the following r linear systems of dimension $N \times N$ in the offline stage:

$$\left((1 + \mu)\, \boldsymbol{M}_h + \mu\, \delta^2\, \boldsymbol{S}_h \right) \boldsymbol{c}_j^{AD} = \left(\boldsymbol{M}_h + \delta^2\, \boldsymbol{S}_h \right) \boldsymbol{c}_j , \qquad j = 1, \ldots, r, \tag{66}$$

where $\boldsymbol{c}_j^{AD} = \{c_{ij}^{AD}\}_{i=1}^N$ is the vector of coefficients in (64).

Comparing approaches (i) and (ii) in terms of computational efficiency, we draw the following conclusions: In approach (i), there is no extra storage required, but the $N \times N$ linear system (63) needs to be solved at each time step in the online stage. In approach (ii), the r linear systems (66) are solved only once, in the offline stage, after which the r vectors $\boldsymbol{c}_j^{AD}$, $j = 1, \ldots, r$ (corresponding to $\boldsymbol{\varphi}_j^{AD}$ in (65)) are used to assemble and store the $r \times r$ matrices and $r \times r \times r$ tensor corresponding to the nonlinearity in the AD-ROM (49). Since the extra storage requirements in approach (ii) are relatively low for the small r that we use in this paper, in Section 5 we utilize this approach instead of approach (i), which has a significantly higher CPU cost.

Using approach (ii), the AD-ROM (49) can be written as the following dynamical system:

$$\dot{\boldsymbol{a}} = \boldsymbol{b} + \left(\boldsymbol{A} + \boldsymbol{A}^{AD}\right) \boldsymbol{a} + \boldsymbol{a}^{\top} \boldsymbol{B}^{AD} \boldsymbol{a}, \tag{67}$$

which can be written componentwise as follows: For all $i = 1, \ldots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} \left(A_{im} + A_{im}^{AD}\right) a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{imn}^{AD}\, a_m(t)\, a_n(t), \tag{68}$$

where

$$b_i = -\left(\mathbf{U} \cdot \nabla \mathbf{U}, \overline{\boldsymbol{\varphi}_i}^h\right) - \frac{2}{Re} \left(\frac{\nabla \mathbf{U} + \nabla \mathbf{U}^{\top}}{2}, \nabla \overline{\boldsymbol{\varphi}_i}^h\right), \tag{69}$$

$$A_{im} = -\frac{2}{Re} \left(\frac{\nabla \boldsymbol{\varphi}_m + \nabla {\boldsymbol{\varphi}_m}^{\top}}{2}, \nabla \boldsymbol{\varphi}_i\right), \tag{70}$$

$$A_{im}^{AD} = -\left(\mathbf{U} \cdot \nabla \boldsymbol{\varphi}_m^{AD}, \overline{\boldsymbol{\varphi}_i}^h\right) - \left(\boldsymbol{\varphi}_m^{AD} \cdot \nabla \mathbf{U}, \overline{\boldsymbol{\varphi}_i}^h\right), \tag{71}$$

$$B_{imn}^{AD} = -\left(\boldsymbol{\varphi}_m^{AD} \cdot \nabla \boldsymbol{\varphi}_n^{AD}, \overline{\boldsymbol{\varphi}_i}^h\right). \tag{72}$$

5. Numerical tests

Goals The main goal of this section is to assess the performance of the AD-ROM (49) in the numerical simulation of a 3D flow past a circular cylinder at Reynolds number $Re = 1000$. Since the AD-ROM is an LES-ROM, we follow the standard approach used in LES model assessment, which we briefly describe next (see, e.g., [25,28] for details). To investigate a new LES model, the following models are generally tested on the same coarse mesh: (i) the new LES model; (ii) the under-resolved solver (i.e., the NSE code without an LES model); (iii) the filtered DNS data (which serves as a benchmark); and (iv) the current state-of-the-art LES model. The new LES model is expected to perform significantly better than the under-resolved solver and to show some improvement over the current state-of-the-art LES model.

To assess the performance of the new AD-ROM, we follow the standard LES approach described above. Thus, we also use the spatially filtered DNS results as a benchmark. Furthermore, we also include G-ROM (9) results as a ROM counterpart of the under-resolved solver used in the LES assessment. Finally, we also make a brief qualitative comparison of the new AD-ROM with the eddy viscosity ROMs (EV-ROMs) [8,14] (which, to our knowledge, are the current state-of-the-art LES-ROMs) and the regularized ROMs (Reg-ROMs) [38]. Just as in the standard LES model assessment, we expect the new AD-ROM to perform significantly better than the G-ROM and show some improvement over the EV-ROMs and Reg-ROMs.

Computational Domain In this section, the x-axis represents the streamwise direction, the y-axis represents the normal direction, and the z-axis represents the spanwise direction. Furthermore, 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).

The computational domain is cylindrical and is centered around the z-axis, i.e., along the spanwise direction. The cylinder of diameter D is also centered around the z-axis. The diameter of the cylindrical computational domain is $15D$ and its spanwise length is $2D$. The free-stream flow is in the positive x-direction. The same computational domain was used in [14,35] (see Fig. 9 and Appendix A in [35]).

Criteria The qualitative behavior of the ROMs is judged according to the following six criteria [14]: (i) the kinetic energy spectrum; (ii) the mean velocity; (iii) the Reynolds stresses; (iv) the root mean square (rms) values of the velocity fluctuations; (v) the time evolution of the POD coefficients; and (vi) the Strouhal number. The first four criteria are statistics that measure the temporal and spatial average behavior of the ROMs, whereas the fifth criterion measures the instantaneous behavior of the ROMs. We also include a computational efficiency assessment for all the ROMs. Next, we briefly present these criteria.

For all statistics, we use the averaging operator $\langle\cdot\rangle = \langle\cdot\rangle_{tyz}$, which consists of averaging in time (over the time interval [0, 300]) and in the y- and z-directions. Specifically, to compute $\langle q\rangle_{tyz}$ for a given quantity q, for each fixed point x, we have

$$\langle q\rangle_{tyz}(x) = \frac{1}{T\, L_y\, L_z} \sum_{t,y,z} q(x, y, z, t), \tag{73}$$

where T is the total time length (i.e., $T = 300$), L_y is the dimension of the computational domain in the y-direction, and L_z is the dimension of the computational domain in the z-direction. Since the numerical results with spatial averaging in the xz-direction are qualitatively similar to those with spatial averaging in the yz-direction, we do not include them in the paper.

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).

Mean Velocity Components: The following mean velocity components are plotted: $\langle u\rangle$ (the mean streamwise velocity), $\langle v\rangle$ (the mean normal velocity), and $\langle w\rangle$ (the mean spanwise velocity).

Reynolds Stresses: The following Reynolds stresses are plotted: $\langle u-\langle u\rangle, v-\langle v\rangle\rangle$ (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).

RMS Values of Velocity Fluctuations: The following rms values of velocity fluctuations are plotted: $\langle u\rangle_{rms} = \langle u-\langle u\rangle, u-\langle u\rangle\rangle$ (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).

Strouhal Number: The Strouhal number (St) is computed as [57,58]

$$St = \frac{f\, D}{U_\infty}, \tag{74}$$

where f is the shedding frequency, U_∞ is the freestream velocity (i.e., the velocity at the inlet), and D is the diameter of the cylinder. Since we have limited the scope of this paper to just velocity ROMs, we do not consider a ROM representation of the pressure field. Hence we do not calculate the lift and drag coefficient spectra. Thus, we compute the shedding frequency f as the frequency corresponding to the first spike in the energy spectrum (see discussion on page 410 in [59]).

Numerical Methods We investigate all 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 length scale and the freestream velocity (U_∞) as the velocity scale. A parallel CFD solver is employed on the time interval [0, 300] to generate the DNS data. Details on the numerical discretization are presented in [57,58,60].

We obtain the POD basis by collecting 1000 snapshots of the DNS velocity field (u, v, w) over the time interval [0, 75] and applying the method of snapshots developed in [7]. These POD modes are then interpolated onto a structured quadratic FE triangulation with nodes coinciding with the nodes used in the original DNS finite difference discretization. All the ROMs use the first $r = 6$ POD modes, which capture 84% of the energy of the velocity fluctuations. As temporal discretization, all the ROMs employ the explicit Euler method with a time step $\Delta t = 7.5 \times 10^{-4}$. (We note that decreasing the time step by a factor of 10 does not significantly change the numerical results.) As spatial discretization, all the ROMs use piecewise quadratic Lagrange FEs. Finally, all the ROMs are investigated on the time interval [0, 300].

Model Parameters In Section 4.3, we discussed two possible discretizations of the C-DF: the ROM-DF and the FE-DF. Since in our preliminary numerical tests the FE-DF yielded more accurate results than the ROM-DF, in this section we use the FE-DF and the resulting AD-ROM formulation in Eqs. (67)–(72).

One of the most important parameters of the AD-ROM is the radius of the ROM spatial filter, δ. In our numerical investigation, we tried only three δ values: $\delta = 0.03$, $\delta = 0.3$ and $\delta = 0.4$. The low value (i.e., $\delta = 0.03$) yielded inaccurate results. The high values (i.e., $\delta = 0.3$ and $\delta = 0.4$) yielded similar results, significantly more accurate than the results for $\delta = 0.03$. Thus, in what follows, we present results for $\delta = 0.3$. We emphasize, however, that the value

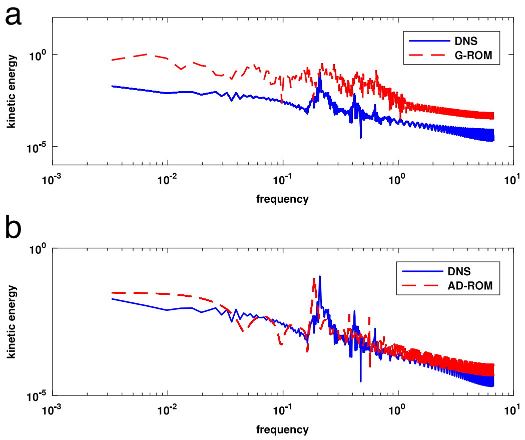

Fig. 1. Kinetic energy spectrum of the spatially filtered DNS (blue) and ROMs (red): (a) the G-ROM, and (b) the AD-ROM. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\delta = 0.3$ does not represent an optimal value; it is just the best value of the very few we tried. To find the optimal value, numerous δ values (and different test problems) should be thoroughly investigated. We plan to carry out this investigation in a future study (see the discussion in Section 6).

Another important parameter of the AD-ROM is the regularization parameter μ used in the Lavrentiev regularization of the AD procedure. In the image processing and inverse problems communities, numerous approaches are proposed to determine the regularization parameters [50–52]. In our numerical investigation, for the fixed $\delta = 0.3$ value, we choose the μ value that ensures that the time average of the L^2-norm of the AD-ROM solution is as close as possible to the time average of the L^2-norm of the filtered DNS data. This approach yields the following optimal value: $\mu = 0.0285$.

5.1. Numerical results

In Fig. 1, we plot the energy spectra of the AD-ROM and, for comparison purposes, of the G-ROM. The two energy spectra are compared with the spatially filtered DNS energy spectrum. The energy spectrum of the AD-ROM is significantly more accurate (i.e., closer to the filtered DNS energy spectrum) than the energy spectrum of the G-ROM. The latter overestimates the energy spectrum of the filtered DNS.

In Fig. 2, we plot the mean velocity components for different x values. Fig. 2 yields the following conclusions: Both the AD-ROM and the G-ROM yield an accurate approximation of the streamwise velocity. The AD-ROM yields an accurate approximation of the mean normal velocity, but the G-ROM does not. Finally, both the AD-ROM and the G-ROM yield inaccurate approximations of the mean spanwise velocity.

In Fig. 3, we plot the Reynolds stresses for different x values. For all the Reynolds stresses, the AD-ROM yields accurate approximations, whereas the G-ROM yields inaccurate approximations.

In Fig. 4, we plot the rms values for different x values. For all the rms values, the AD-ROM yields accurate approximations, whereas the G-ROM yields inaccurate approximations.

The time evolutions of the ROMs' POD basis coefficients a_1, a_2 and a_6 on the entire time interval $[0, 300]$ are shown in Fig. 5. We note that the other POD coefficients have a similar behavior. Thus, for clarity of exposition, we include only a_1, a_2 and a_6. The G-ROM results are clearly inaccurate, since the coefficients' magnitudes are several times larger than those of the filtered DNS data. The AD-ROM yields significantly more accurate results for all the coefficients.

The Strouhal numbers predicted by the AD-ROM, G-ROM and filtered DNS are listed in Table 1. The Strouhal number predicted by the AD-ROM is slightly lower than the Strouhal number computed from the filtered DNS data because the AD-ROM's shedding frequency is lower than the filtered DNS' shedding frequency (see Fig. 1). We note that the G-ROM fails to predict a clear shedding frequency and, thus, cannot yield a Strouhal number.

To measure the computational efficiency of the AD-ROM (which is one of the main requirements for any successful ROM), we list in Table 2 its offline and online CPU times together with the G-ROM offline and online CPU times and the DNS CPU time. We draw the following conclusions from Table 2:

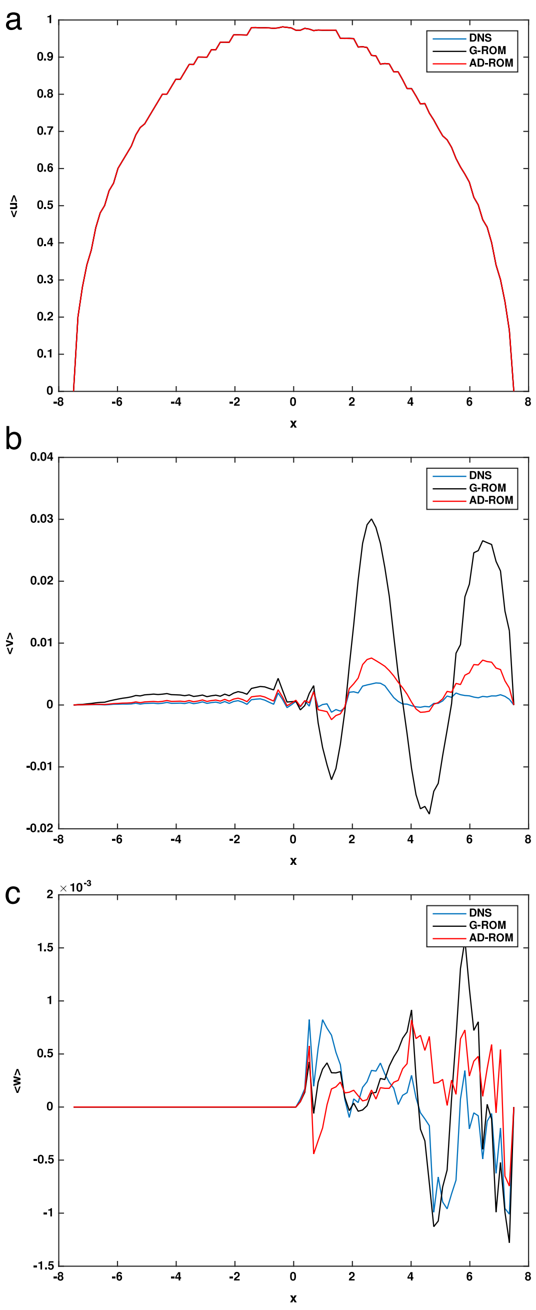

Fig. 2. Mean velocity components of the spatially filtered DNS and ROMs: (a) $\langle u \rangle$ (the mean streamwise velocity), (b) $\langle v \rangle$ (the mean normal velocity), and (c) $\langle w \rangle$ (the mean spanwise velocity). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Strouhal numbers predicted by ROMs.

	DNS	G-ROM	AD-ROM
St	0.2083	–	0.1888

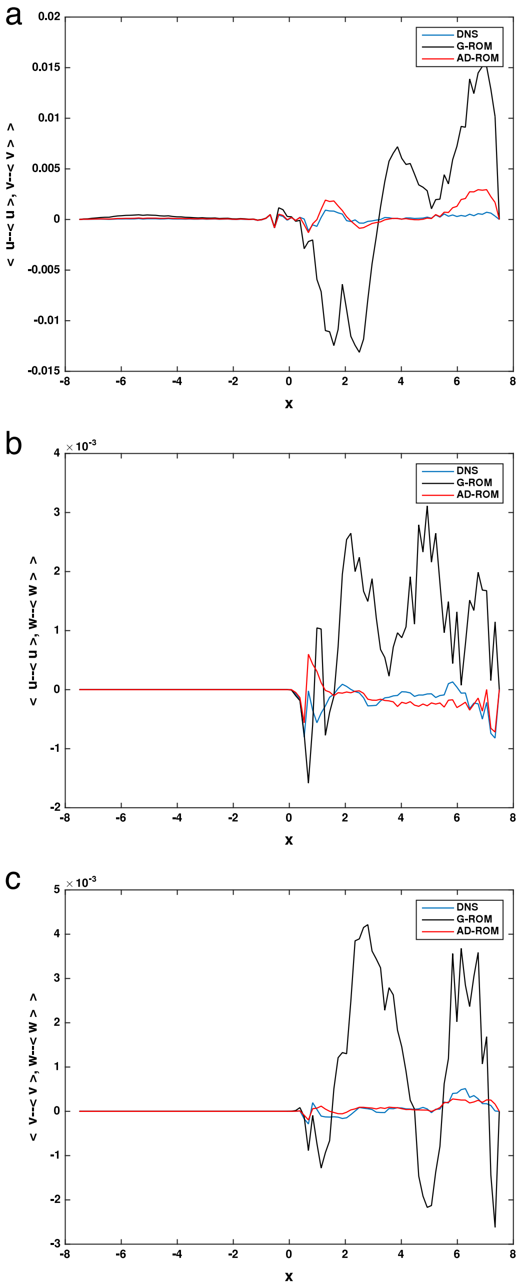

Fig. 3. Reynolds stresses of the spatially filtered DNS and ROMs: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$ (the xy-component of the Reynolds stress), (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$ (the xz-component of the Reynolds stress), and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$ (the yz-component of the Reynolds stress). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

First, the AD-ROM online CPU time is on the same order as the G-ROM online CPU time. Both online CPU times are about four orders of magnitude lower than the DNS CPU time.

Second, the offline CPU times of both the AD-ROM and the G-ROM are about two orders of magnitude lower than the DNS CPU time. We note, however, that the AD-ROM and G-ROM offline CPU times are significantly higher than their corresponding online CPU times. Indeed, it is well known in the ROM community that in the offline stage, in order to assemble the vectors, matrices and tensors used in the dynamical systems formulation of the G-ROM, the underlying numerical discretization (e.g., FE discretization) needs to be used. Thus, the G-ROM offline CPU

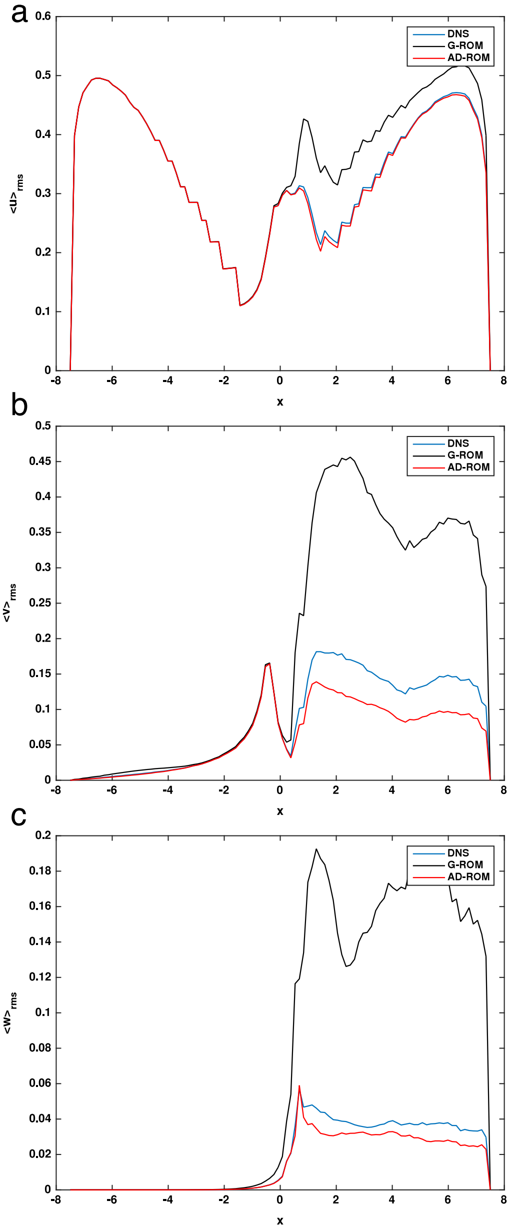

Fig. 4. Rms values of the velocity fluctuations of the spatially filtered DNS and ROMs: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$ (the rms value of the streamwise velocity fluctuations), (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$ (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). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

time is significant. We also note that, since we use the FE-DF in the numerical discretization of the AD-ROM (see Section 4.3), additional vectors and matrices need to be assembled. This makes the AD-ROM offline CPU time higher than the G-ROM offline CPU time.

Finally, we note that in the applications generally targeted by ROMs, the models are used numerous times and for long time intervals [2,4,5]. Thus, in this type of applications, the bulk of the computational cost of the ROMs is paid in the online stage. Since, as mentioned above, the AD-ROM (and G-ROM) online CPU time is significantly lower than the DNS CPU time, we deem the AD-ROM as computationally efficient.

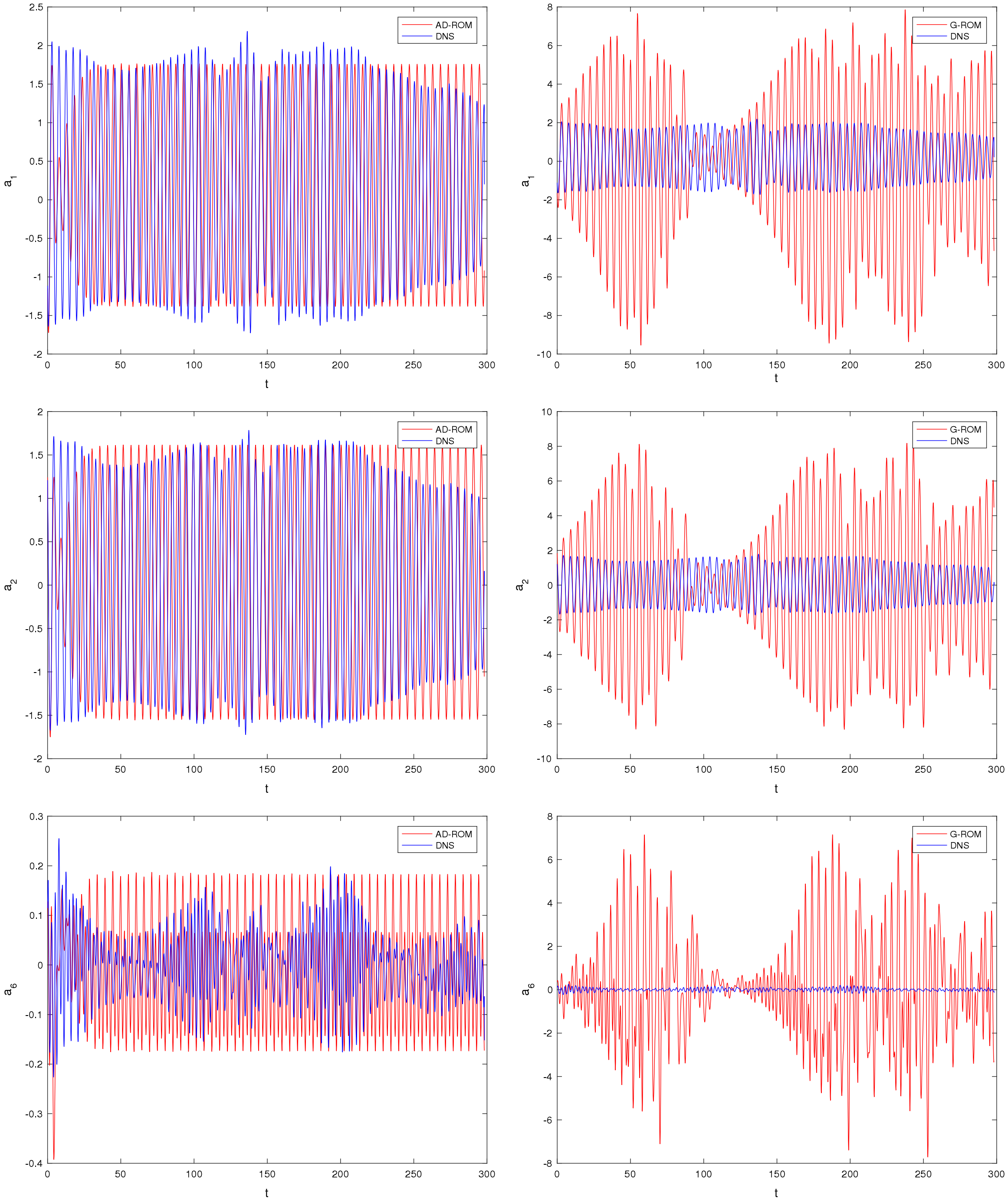

Fig. 5. Time evolution of the POD basis coefficients a_1 (first row), a_2 (second row) and a_6 (third row) of the spatially filtered DNS (blue) and ROMs (red): the AD-ROM (left column) and G-ROM (right column). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. *Comparison with other ROMs*

In Section 5.1, to assess the performance of the new AD-ROM, we compared it with the G-ROM and the filtered DNS data. Overall, the AD-ROM results were significantly more accurate than the G-ROM results. Thus, the AD-ROM satisfied the basic accuracy criterion of any successful ROM. Furthermore, the AD-ROM results were

Table 2
Offline and online CPU times (in seconds) of DNS and ROMs.

DNS	G-ROM offline	G-ROM online	AD-ROM offline	AD-ROM online
2.3e+06	5.9e+04	9.2e+01	6.1e+04	1.4e+02

relatively close to the filtered DNS results, which prompted us to conclude that the AD-ROM was relatively accurate. A natural question, however, is how the AD-ROM's accuracy compares with the accuracies of other ROMs for convection-dominated flows.

We emphasize that a thorough comparison of the AD-ROM with the plethora of ROMs for convection-dominated flows would require the numerical investigation of different types of ROM strategies for various computational settings and numerous parameter choices. Thus, while needed, this comparison is beyond the scope of this paper. We will, however, present next a brief qualitative comparison of the AD-ROM with two other representative classes of ROMs: the EV-ROMs [3,8,14] and the Reg-ROMs [37,38]. Since both the test problem and the statistics used in [14,38] are identical to those used in this paper, our qualitative comparison is meaningful.

Overall, the AD-ROM results are similar to those for the EV-ROMs in [14] and Reg-ROMs in [37,38]: For example, the accuracy of the AD-ROM energy spectrum (see Fig. 1 in this paper) is similar to the accuracy of the energy spectra of the *best* EV-ROMs (see Fig. 3 in [14]) and *best* Reg-ROMs (see Fig. 2 in [38]). The AD-ROM's accuracy is also similar to that of the best EV-ROMs and Reg-ROMs with respect to other statistics and time evolution of the ROM coefficients (compare, e.g., Fig. 5 in this paper, Fig. 7 and Fig. 8 in [14], and Fig. 6 in [38]).

We believe that the AD-ROM's performance is impressive given that we did not use any explicit numerical dissipation mechanisms, which are generally expected for ROMs of convection-dominated flows. Indeed, we emphasize that the EV-ROMs include terms that allow explicit dissipation [3,8,14] and the Reg-ROMs increase the numerical stability by using explicit spatial filtering of various terms in the models [37,38]. The AD-ROM, however, does not use any explicit dissipation mechanism. We followed this approach in order to decouple the AD and dissipation effects, which, we believe, is necessary in the first step of a careful assessment the AD procedure in a ROM setting.

6. Conclusions and future work

We used explicit ROM spatial filtering to develop a large eddy simulation ROM (LES-ROM) framework. Within this LES-ROM framework, we proposed an approximate deconvolution ROM (AD-ROM). We assessed the AD-ROM in the numerical simulation of the 3D flow past a circular cylinder at $Re = 1000$. The numerical investigation showed that the accuracy of the AD-ROM was significantly higher than the accuracy of the standard G-ROM and on the same order as the accuracy of the eddy viscosity ROMs (EV-ROMs) [14] and Reg-ROMs [38]. Finally, we emphasize that the CPU time of the AD-ROM was orders of magnitude lower than the CPU time of the DNS.

These first steps in the investigation of the LES-ROM framework and AD-ROM yielded encouraging results. We emphasize, however, that there are numerous open questions, both at a fundamental (theoretical and modeling) and practical (discretization) level. This is, of course, natural, since in the LES field the same issues have been tackled over decades. We believe that the following are some of the most stringent issues that need to be addressed in the LES-ROM framework (in general) and AD-ROM (in particular).

LES-ROM Framework Open Questions. We believe that the choice of the *explicit ROM spatial filter* used to develop the LES-ROM framework should be thoroughly investigated. Indeed, in the LES field, several spatial filters are commonly used, e.g., the Gaussian, box, sharp cutoff and differential filters (see [25,28] for more details). The effect of spatial filtering on the resulting LES model is relatively well understood. In the ROM field, however, to our knowledge, only two ROM spatial filters have been used so far: the ROM projection and the ROM differential filter. Their effect on the resulting ROMs has been investigated in [14,37,38,53,56]. We believe, however, that finding the best explicit ROM spatial filter for the LES-ROM framework is an important open question.

Once the ROM spatial filter is chosen, another open question for the LES-ROM framework is finding the *optimal filter radius,* δ. We emphasize that, although heuristic scalings (e.g., $\delta \sim h$, where h is the mesh size) are often used in LES, the optimal scaling is, to our knowledge, still an open question in the LES field [25,28]. To answer this question in our LES-ROM framework, we will use the LES-ROM discretization error (which consists of the error of the standard numerical discretization, the filtering error [25,45,47] and the ROM truncation [61]) to develop δ scalings, which we will then numerically investigate for different test problems.

In the derivation of the LES-ROM framework in Section 3, we assumed that the differentiation and spatial filtering operators commute. We note that in the LES field, this assumption fails in certain situations (for example, when the radius of the spatial filter, δ, is variable) [25,28]. In this cases, the *LES-ROM commutation error*, i.e., the error made by using the above assumption, must be addressed. Using a variable δ seems appropriate for our LES-ROM framework as well, especially when the POD modes display highly localized spatial features. Thus, we plan to investigate the magnitude of the LES-ROM commutation error when a variable δ is used. If the commutation error is relatively large compared with the discretization and modeling errors, then we will use various strategies to model it, just as in LES [25].

LES: Before or After? A natural question is whether spatial filtering and LES should be used to generate the snapshot data (i.e., before the ROM is used) or only to develop the ROM. We believe that in realistic, turbulent flow applications, spatial filtering and LES should be used both in the data generation and the ROM development (i.e., *both "before" and "after"*). The same conclusion seems to be yielded by the numerical investigation of a flow past the Ahmed body at $Re = 300{,}000$ in [18], in which a standard LES model (the Smagorinsky model) was used to generate the snapshots and a functional ROM closure model was utilized to stabilize the ROM. Since this paper represents the first step in the investigation of the new AD-ROM, we decided to separate the spatial filtering effects in the data generation from those in the ROM development. Thus, in this paper we only considered the "after" case. However, for the numerical simulation of realistic, high Reynolds number flows, we plan to use spatial filtering and LES both in the data generation and ROM development (i.e., both "before" and "after").

AD-ROM Open Questions. The choice of other *regularization methods* in the AD-ROM should be carefully investigated. Indeed, as a first step, in this paper we considered only one of the simplest regularization methods, the Lavrentiev regularization. We plan, however, to investigate other regularization strategies, both variational and iterative [50–52].

Probably the most important research direction that we plan to pursue is to investigate the AD-ROM in the numerical simulation of *realistic turbulent flows that display recurrent structures* (see, e.g., [18,24]). We emphasize, however, that in these challenging settings, the current AD-ROM should probably be supplemented with a *numerical dissipation mechanism* (e.g., time relaxation [45]), just as in LES [30–32,49]. In this paper we investigated the AD-ROM without any additional numerical dissipation mechanism. We believe that decoupling the AD and dissipation mechanisms was necessary in order to understand the effect of the former, which, to our knowledge, is new in a ROM setting. We note, however, that for higher Reynolds number flows, adding numerical dissipation would probably be needed in order to account for the subgrid-scale effects due to the inherently coarse meshes used in the data generation.

In our numerical investigation, the accuracy of the new AD-ROM was similar to that of the EV-ROMs [14] and Reg-ROMs [38]. We believe that this is impressive, given that the new AD-ROM did not use any explicit numerical dissipation mechanisms, whereas the other ROMs did. For high Reynolds number flows, we plan to add numerical dissipation (e.g., time relaxation) to the AD-ROM and perform a *thorough comparison with other types of ROMs* (e.g., EV-ROMs [3,8,14], Reg-ROMs [37,38] and calibrated ROMs [62–65]).

Acknowledgments

We greatly appreciate the financial support of the National Science Foundation through grants DMS-1016450, DMS-1522672 and DMS-1344962. The third author was partially supported by the office of the Vice President for Research at University of South Carolina through the ASPIRE-I program (21400-E102). We also thank the two anonymous referees for their comments and suggestions, which significantly improved this paper.

References

[1] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, R. Scrofani, Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization, J. Comput. Phys. 315 (2016) 609–628.
[2] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer, 2015.
[3] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge, 1996.
[4] B.R. Noack, M. Morzynski, G. Tadmor, Reduced-Order Modelling for Flow Control, vol. 528, Springer Verlag, 2011.

[5] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, vol. 92, Springer, 2015.

[6] R. Ştefănescu, A. Sandu, I.M. Navon, POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation, J. Comput. Phys. 295 (2015) 569–595.

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

[8] 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.

[9] M.J. Balajewicz, E.H. Dowell, B.R. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation, J. Fluid Mech. 729 (2013) 285–308.

[10] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, Internat. J. Numer. Methods Engrg. 102 (2015) 1136–1161.

[11] S. Giere, T. Iliescu, V. John, D. Wells, SUPG reduced order models for convection-dominated convection–diffusion-reaction equations, Comput. Methods Appl. Mech. Engrg. 289 (2015) 454–474.

[12] P. Pacciarini, G. Rozza, Stabilized reduced basis method for parametrized advection–diffusion PDEs, Comput. Methods Appl. Mech. Engrg. 274 (2014) 1–18.

[13] A. Quarteroni, G. Rozza, A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, J. Math. Ind. 1 (2011) 1–49.

[14] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison, Comput. Methods Appl. Mech. Engrg. 237–240 (2012) 10–26.

[15] M.F. Barone, I. Kalashnikova, D.J. Segalman, H.K. Thornquist, Stable Galerkin reduced order models for linearized compressible flow, J. Comput. Phys. 228 (2009) 1932–1946.

[16] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives, in: Reduced Order Methods for Modeling and Computational Reduction, Springer, 2014, pp. 235–273.

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

[18] J. Östh, B.R. Noack, S. Krajnović, D. Barros, J. Borée, On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body, J. Fluid Mech. 747 (2014) 518–544.

[19] B. Protas, B.R. Noack, J. Östh, Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows, J. Fluid Mech. 766 (2015) 337–367.

[20] 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.

[21] D. Amsallem, C. Farhat, Stabilization of projection-based reduced-order models, Internat. J. Numer. Methods. Engrg. 91 (2012) 358–377.

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

[23] 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.

[24] J. Östh, Unsteady numerical simulations and reduced-order modelling of flows around vehicles (Ph.D. thesis), Chalmers University of Technology, 2014.

[25] L.C. Berselli, T. Iliescu, W.J. Layton, Mathematics of Large Eddy Simulation of Turbulent Flows, in: Scientific Computation, Springer-Verlag, Berlin, 2006.

[26] B. Geurts, Elements of Direct and Large-Edddy Simulation, R.T. Edwards, Inc., 2003.

[27] T.C. Rebollo, R. Lewandowski, Mathematical and Numerical Foundations of Turbulence Models and Applications, Springer, 2014.

[28] P. Sagaut, Large Eddy Simulation for Incompressible Flows, third ed., in: Scientific Computation, Springer-Verlag, Berlin, 2006.

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

[30] S. Stolz, N. Adams, An approximate deconvolution procedure for large-eddy simulation, Phys. Fluids 11 (1999) 1699–1701.

[31] S. Stolz, N. Adams, L. Kleiser, An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows, Phys. Fluids 13 (2001) 997–1015.

[32] S. Stolz, N. Adams, L. Kleiser, The approximate deconvolution model for large-eddy simulations of compressible flows and its application to shock-turbulent-boundary-layer interaction, Phys. Fluids 13 (2001) 2985–3001.

[33] 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.

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

[35] 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.

[36] S. Aradag, S. Siegel, J. Seidel, K. Cohen, T. McLaughlin, Filtered POD-based low-dimensional modeling of the 3D turbulent flow behind a circular cylinder, Internat. J. Numer. Methods Fluids 66 (2011) 1–16.

[37] F. Sabetghadam, A. Jafarpour, α regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation, Appl. Math. Comput. 218 (2012) 6012–6026.

[38] D. Wells, Z. Wang, X. Xie, T. Iliescu, An evolve-then-filter regularized reduced order model for convection-dominated flows, Internat. J. Numer. Methods Fluids (2016) in revision. Available as arXiv preprint at http://arxiv.org/abs/1506.07555.

[39] P. Gresho, R. Sani, Incompressible Flow and the Finite Element Method, Isothermal Laminar Flow, vol. 2, John Wiley & Sons, 2000, Paperback.

[40] A. Ern, J.-L. Guermond, Theory and Practice of Finite Elements, in: Applied Mathematical Sciences, vol. 159, Springer-Verlag, New York, 2004.

[41] A. Caiazzo, T. Iliescu, V. John, S. Schyschlowa, A numerical investigation of velocity-pressure reduced order models for incompressible flows, J. Comput. Phys. 259 (2014) 598–616.

[42] W.R. Graham, J. Peraire, K.Y. Tang, Optimal control of vortex shedding using low-order models. Part I—open-loop model development, Internat. J. Numer. Methods Engrg. 44 (1999) 945–972.

[43] M. Germano, Differential filters for the large eddy numerical simulation of turbulent flows, Phys. Fluids 29 (1986) 1755–1757.

[44] M. Germano, Differential filters of elliptic type, Phys. Fluids 29 (1986) 1757–1758.

[45] W.J. Layton, L.G. Rebholz, Approximate Deconvolution Models of Turbulence: Analysis, Phenomenology and Numerical Analysis, vol. 2042, Springer, 2012.

[46] S. Kaya, C.C. Manica, Convergence analysis of the finite element method for a fundamental model in turbulence, Math. Models Methods Appl. Sci. 22 (2012) 1250033.

[47] V. John, Large Eddy Simulation of Turbulent Incompressible Flows, in: Lecture Notes in Computational Science and Engineering, vol. 34, Springer-Verlag, Berlin, 2004, Analytical and Numerical Results for a Class of LES Models.

[48] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations, Comput. Methods Appl. Mech. Engrg. (2016).

[49] S. Stolz, N. Adams, Large-eddy simulation of high-Reynolds-number supersonic boundary layers using the approximate deconvolution model and a rescaling and recycling technique, Phys. Fluids 15 (2003) 2398–2412.

[50] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, Bristol, 1998.

[51] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, vol. 7, Society for Industrial and Applied Mathematics, 2010.

[52] C.R. Vogel, Computational Methods for Inverse Problems, vol. 23, Society for Industrial and Applied Mathematics (SIAM), 2002.

[53] X. Xie, Approximate deconvolution reduced order modeling (Master's thesis), Virginia Tech, 2015, http://theses.lib.vt.edu/theses/available/etd-11232015-161835/unrestricted/Xie_X_T_2015.pdf.

[54] C.C. Manica, S.K. Merdan, Finite element error analysis of a zeroth order approximate deconvolution model based on a mixed formulation, J. Math. Anal. Appl. 331 (2007) 669–685.

[55] K.J. Galvin, L.G. Rebholz, C. Trenchea, Efficient, unconditionally stable, and optimally accurate FE algorithms for approximate deconvolution models, SIAM J. Numer. Anal. 52 (2014) 678–707.

[56] D. Wells, Stabilization of POD-ROMs (Ph.D. thesis), Virginia Tech, 2015, http://vtechworks.lib.vt.edu/bitstream/handle/10919/52960/Wells_DR_D_2015.pdf?sequence=1&isAllowed=y.

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

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

[59] A.G. Kravchenko, P. Moin, Numerical studies of flow over a circular cylinder at $Re_D = 3900$, Phys. Fluids 12 (2000) 403–417.

[60] I. Akhtar, J. Borggaard, J.A. Burns, H. Imtiaz, L. Zietsman, Using functional gains for effective sensor location in flow control: a reduced-order modelling approach, J. Fluid Mech. 781 (2015) 622–656.

[61] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, Numer. Math. 90 (2001) 117–148.

[62] A.K. Alekseev, I.M. Navon, The analysis of an ill-posed problem using multi-scale resolution and second-order adjoint techniques, Comput. Methods Appl. Mech. Engrg. 190 (2001) 1937–1953.

[63] L. Cordier, B. Abou El Majd, J. Favier, Calibration of POD reduced-order models using Tikhonov regularization, Internat. J. Numer. Methods Fluids 63 (2010) 269–296.

[64] Y. Wang, I.M. Navon, X. Wang, Y. Cheng, 2D Burgers equations with large Reynolds number using POD/DEIM and calibration, 2015. Preprint, http://people.sc.fsu.edu/˜inavon/pubs/poddeim_burgers.pdf.

[65] J. Weller, E. Lombardi, A. Iollo, Robust model identification of actuated vortex wakes, Physica D 238 (2009) 416–427.