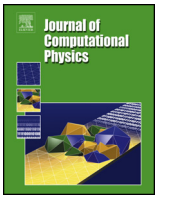
Journal of Computational Physics 486 (2023) 112127

Contents lists available at ScienceDirect

Journal of Computational Physics

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



A hybrid projection/data-driven reduced order model for the Navier-Stokes equations with nonlinear filtering stabilization

Michele Girfoglio [a,*], Annalisa Quaini [b], Gianluigi Rozza [a]

[a] SISSA, International School for Advanced Studies, Mathematics Area, mathLab, via Bonomea, Trieste 265 34136, Italy
[b] Department of Mathematics, University of Houston, Houston TX 77204, USA

ARTICLE INFO

Article history:
Received 11 September 2021
Received in revised form 26 March 2023
Accepted 2 April 2023
Available online 12 April 2023

Keywords:
Proper orthogonal decomposition
Reduced order model
Large Eddy Simulation
Nonlinear filtering stabilization
Projection-based methods
Data-driven strategies

ABSTRACT

We develop a Reduced Order Model (ROM) for the Navier-Stokes equations with nonlinear filtering stabilization. Our approach, that can be interpreted as a Large Eddy Simulation model, combines a three-step algorithm called Evolve-Filter-Relax (EFR) with a computationally efficient finite volume method. The main novelty of our ROM lies in the use within the EFR algorithm of a nonlinear, deconvolution-based indicator function that identifies the regions of the domain where the flow needs regularization. The ROM we propose is a hybrid projection/data-driven strategy: a classical Proper Orthogonal Decomposition Galerkin projection approach for the reconstruction of the velocity and the pressure fields and a data-driven reduction method to approximate the indicator function used by the nonlinear differential filter. This data-driven technique is based on interpolation with Radial Basis Functions. We test the performance of our ROM approach on two benchmark problems: 2D and 3D unsteady flow past a cylinder at Reynolds number $0 \leq Re \leq 100$. The accuracy of the ROM is assessed against results obtained with the full order model for velocity, pressure, indicator function and time evolution of the aerodynamics coefficients.

1. Introduction

Although increasing computational power has become available recently, the resolution of systems of parametric Partial Differential Equations using classical discretization methods (e.g., finite element or finite volume methods, hereinafter called Full Order Models) is still unfeasible in several situations where one needs to evaluate a certain output of interest for a large number of input configurations. This is the case of, e.g., real-time control problems, optimization problems, and uncertainty quantification. In this context, Reduced Order Models (ROMs) have been proposed as an efficient tool to significantly reduce the computational cost required by classical Full Order Models (FOMs).

The basic ROM framework consists of two steps. The first one is the so-called *offline* phase, when a database of several solutions is collected by solving the original FOM for selected parameter values. The second step is the *online* phase, during which the information obtained in the offline phase is used to quickly compute the solution for newly specified values of the parameters. The assumption underlying this framework is that the solution of a parametrized PDE (or a system of such equations) lies on a low-dimensional manifold, which can be approximated by a subspace spanned by the set of basis functions found in the offline phase. For a comprehensive review on ROMs, the reader is referred to, e.g., [1–6].

* Corresponding author.
E-mail addresses: mgirfogl@sissa.it (M. Girfoglio), quaini@math.uh.edu (A. Quaini), grozza@sissa.it (G. Rozza).

https://doi.org/10.1016/j.jcp.2023.112127
0021-9991/

In recent years, there has been a growing interest in constructing stable, accurate, and computationally efficient ROMs for the numerical simulation of higher Reynolds number flows. Traditional ROMs fail to maintain their promise of reduced computational costs when the Kolmogorov n-width of the solution manifold associated with the FOM is large, as is the case in convection-dominated flows. Indeed, if one chooses to retain a large number of modes in order for the ROM to be accurate, then the computational efficiency suffers. If the number of modes is otherwise kept low, a severe loss of information hinders the accurate reconstruction of the solution. In fact, projection based ROMs of turbulent flows are affected by energy stability problems related to the fact that proper orthogonal decomposition (POD) retains the modes biased toward large, high-energy scales, while the turbulent kinetic energy is dissipated at the level of the small turbulent scales. A possible way to tackle this challenging problem is to introduce dissipation via a closure model [7,8]. In [9], it was shown theoretically and numerically that modes have a similar energy transfer mechanism to Fourier modes. Therefore, the use of Large Eddy Simulation (LES) could be beneficial. Following this idea, in this paper we develop a ROM for an LES filtering approach for flows at moderate Reynolds numbers (a few hundreds or a few thousands).

We focus on a nonlinear variant of the Evolve-Filter-Relax (EFR) algorithm [10–14], which describes the effect of the small scales (i.e., the turbulent scales that are not resolved by the discretization mesh) by a set of equations to be added to the discrete Navier-Stokes equations. This extra problem can be interpreted as a differential low-pass filter. Modularity is an appealing feature of this approach since it can be implemented without a major modification of a legacy Navier-Stokes solver. A key role in the EFR algorithm is played by a nonlinear indicator function that identifies the regions of the domain where the flow needs regularization (i.e., where and how much artificial viscosity is needed) [15]. The EFR algorithm has been demonstrated to be an efficient and accurate LES approach for flow at moderate Reynolds numbers (i.e., of the order of a few thousand) [14,16]. At the full order level, the EFR algorithm and its variant without relaxation (called EF) have been extensively studied within a Finite Element framework. Instead, we choose to apply a computationally efficient Finite Volume (FV) method for the space discretization [16,17]. The motivation for this choice is that many commercial codes are based on FV methods. Thus, a FV-based ROM would be appealing for a larger group of CFD practitioners.

The large majority of the regularized ROMs adds the filtering step only at the reduced order level, i.e., the snapshots are obtained by Direct Numerical Simulation. See, e.g., to [18–20]. We proposed a different approach in [21–23] for the EF/EFR algorithm: we apply the filter step at both the full and reduced order level, i.e., we generate the snapshots with under-refined meshes. Such an approach provides a ROM that is fully consistent with the FOM since the same mathematical framework is used during both the *offline* and *online* stage. We use the POD basis related to the evolve velocity to approximate the filtered velocity and we compute the reduced pressure field with a Poisson Pressure Equation (PPE) method [24,25]. The main difference between this work and [21–23] lies in the indicator function. Indeed, the EF or EFR algorithms in [21–23] adopt a linear indicator function. While this was a necessary simplification that allowed us to focus on other challenges posed by a ROM differential filter, it is known that a linear indicator function is insufficiently selective as it applies the same amount of artificial viscosity everywhere in the domain. Thus, in this paper we extend our approach to a nonlinear indicator function.

We propose a hybrid projection/data-driven ROM that draws inspiration from the work in [26], where a ROM framework for the Reynolds-averaged Navier Stokes (RANS) equations is developed. We choose this approach, which exploits a traditional projection method for the computation of the reduced velocity and pressure fields and uses a data-driven technique to compute the reduced coefficients of the indicator function field, because it is accurate and partially non-intrusive. Other ROM techniques available in the literature are known to be either inaccurate or not easy to implement in legacy codes, hence they have not been considered for comparison in this paper. For example, one alternative approach would be to use the same set of reduced coefficients for velocity, pressure, and indicator function. This has been considered for RANS in [27] and it was shown in [26] to provide less accurate results than a hybrid procedure. It is reasonable to assume this lack of accuracy could be observed for LES too. Another option would be to use an EIM/DEIM technique [28,29] for the reconstruction of the indicator function field but its implementation is problem-dependent and intrusive. Another advantage of our hybrid procedure is that it provides a unique computational pipeline for the development of efficient ROMs for flows at higher Reynolds number, no matter if modeled by LES or RANS.

We test our approach on two benchmarks: 2D [30,31] and 3D [30,32,33] flow past a cylinder with time-dependent Reynolds number $0 \leq Re(t) \leq 100$. In order to obtain accurate results, one needs a mesh with about 200k elements for the 2D benchmark and roughly 3 million elements for the 3D benchmark. With the EFR algorithm, one can get accurate results with much coarser meshes [13–16,21–23], which would otherwise lead to inaccurate solutions. We limit our investigation to the ROM reconstruction of the time evolution of the system. Parametric studies, in particular related to key model parameters such as filtering radius and relaxation parameter, will be addressed in a future work. For both tests, we compare the evolution of velocity, pressure, and indicator function with the corresponding FOM quantities. Additionally, we show the accuracy of our approach in the time reconstruction of the lift and drag coefficients.

The work is organized as follows. Sec. 2 describes the full order model and the numerical method we use for it. Sec. 3 presents the ingredients of the reduced order model. The numerical examples are reported in Sec. 4. Sec. 5 provides conclusions and perspectives.

2. The full order model

We consider the motion of an incompressible viscous fluid in a fixed domain $\Omega \subset \mathbb{R}^D$ with $D = 2, 3$ over a time interval of interest $(t_0, T) \subset \mathbb{R}^+$. The flow is described by the incompressible Navier-Stokes equations (NSE):

$$\rho\, \partial_t \boldsymbol{u} + \rho\, \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - 2\mu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega \times (t_0, T), \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega \times (t_0, T), \tag{2}$$

where ρ is the fluid density, μ is the dynamic viscosity, $\boldsymbol{u}$ is velocity, and p is the pressure. Problem (1)-(2) is endowed with suitable boundary conditions

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega_D \times (t_0, T), \tag{3}$$

$$(2\mu \nabla \boldsymbol{u} - p\mathbf{I})\boldsymbol{n} = \mathbf{0} \qquad \text{on } \partial\Omega_N \times (t_0, T), \tag{4}$$

and the initial data $\boldsymbol{u} = \boldsymbol{u}_0$ in $\Omega \times \{t_0\}$. Here $\overline{\partial\Omega_D} \cup \overline{\partial\Omega_N} = \overline{\partial\Omega}$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. In addition, $\boldsymbol{u}_D$ and $\boldsymbol{u}_0$ are given.

In order to characterize the flow regime under consideration, we define the Reynolds number as

$$Re = \frac{UL}{\nu}, \tag{5}$$

where $\nu = \mu/\rho$ is the *kinematic* viscosity of the fluid, and U and L are characteristic macroscopic velocity and length, respectively. We are interested in simulations flows at moderately large Reynolds numbers, for which flow disturbances cannot be neglected and Reynolds-averaged Navier-Stokes (RANS) models [34] are inaccurate. For this reason, we choose to work with a Large Eddy Simulation approach that is described next.

2.1. The evolve-filter-relax algorithm

Our LES approach is a modular algorithm, called Evolve-Filter-Relax [13], that adds a differential filter to the Navier-Stokes equations (1)-(2). This algorithm comes from the decoupling of the time-discrete Leray model [35]. For the implementation of the EFR algorithm, we chose the C++ finite volume library OpenFOAM® [36].

Let $\Delta t \in \mathbb{R}$, $t^n = t_0 + n\Delta t$, with $n = 0, \ldots, N_T$ and $T = t_0 + N_T \Delta t$. We denote by y^n the approximation of a generic quantity y at the time t^n. The EFR algorithm reads as follows: given the velocities $\boldsymbol{u}^{n-1}$ and $\boldsymbol{u}^n$, at t^{n+1}:

i) *Evolve*: find intermediate velocity and pressure $(\boldsymbol{v}^{n+1}, p^{n+1})$ such that

$$\rho\, \frac{3}{2\Delta t} \boldsymbol{v}^{n+1} + \rho\, \nabla \cdot \left(\boldsymbol{v}^n \otimes \boldsymbol{v}^{n+1}\right) - 2\mu \Delta \boldsymbol{v}^{n+1} + \nabla p^{n+1} = \boldsymbol{b}^{n+1}, \tag{6}$$

$$\nabla \cdot \boldsymbol{v}^{n+1} = 0, \tag{7}$$

with boundary conditions

$$\boldsymbol{v}^{n+1} = \boldsymbol{u}_D^{n+1} \quad \text{on } \partial\Omega_D \times (t_0, T), \tag{8}$$

$$(2\mu \nabla \boldsymbol{v}^{n+1} - p^{n+1}\mathbf{I})\boldsymbol{n} = \mathbf{0} \quad \text{on } \partial\Omega_N \times (t_0, T), \tag{9}$$

and initial condition $\boldsymbol{v}^0 = \boldsymbol{u}_0$ in $\Omega \times \{t_0\}$. In eq. (6), we set $\boldsymbol{b}^{n+1} = \rho(4\boldsymbol{u}^n - \boldsymbol{u}^{n-1})/(2\Delta t)$. Notice that this step corresponds to a time discretization with Backward Differentiation Formula of order 2 (BDF2) of problem (1)-(2).

Remark 2.1. We adopt a first order extrapolation for the convective velocity although a BDF2 scheme is used for the time discretization of problem (1)-(2). This is what the NSE solvers in OpenFOAM do, so discretization (6)-(7) would make it a fair comparison between EFR and NSE algorithms.

ii) *Filter*: find filtered velocity $\overline{\boldsymbol{v}}^{n+1}$ such that

$$-\alpha^2 \nabla \cdot \left(a(\boldsymbol{v}^{n+1}) \nabla \overline{\boldsymbol{v}}^{n+1}\right) + \overline{\boldsymbol{v}}^{n+1} = \boldsymbol{v}^{n+1}, \tag{10}$$

with boundary conditions

$$\overline{\boldsymbol{v}}^{n+1} = \boldsymbol{u}_D^{n+1} \quad \text{on } \partial\Omega_D \times (t_0, T), \tag{11}$$

$$\nabla \overline{\boldsymbol{v}}^{n+1} \boldsymbol{n} = \mathbf{0} \quad \text{on } \partial\Omega_N \times (t_0, T). \tag{12}$$

In eq. (10), $\overline{\boldsymbol{v}}$ is the *filtered velocity* and α can be interpreted as the *filtering radius* (that is, the radius of the neighborhood where the filter extracts information from the unresolved scales). *Indicator function* $a(\cdot)$ is such that:

$a(\boldsymbol{v}) \simeq 0$ where the velocity $\boldsymbol{v}$ does not need regularization;

$a(\boldsymbol{v}) \simeq 1$ where the velocity $\boldsymbol{v}$ does need regularization.

Different choices of $a(\cdot)$ have been proposed and compared in [37,13,38–40]. We choose indicator function:

$$a(\boldsymbol{v}) = |\boldsymbol{v} - F(\boldsymbol{v})|, \tag{13}$$

where we take $F(\boldsymbol{v})$ to be the linear Helmholtz filter operator, i.e., $F(\boldsymbol{v}) = \tilde{\boldsymbol{u}}$ with:

$$\tilde{\boldsymbol{u}}^{n+1} - \alpha^2 \Delta \tilde{\boldsymbol{u}}^{n+1} = \boldsymbol{v}^{n+1}. \tag{14}$$

For more details on this indicator function, we refer to [14].

iii) *Relax*: set

$$\boldsymbol{u}^{n+1} = (1-\chi)\boldsymbol{v}^{n+1} + \chi \overline{\boldsymbol{v}}^{n+1}, \tag{15}$$

where $\chi \in (0, 1]$ is a relaxation parameter.

We consider $\boldsymbol{u}^{n+1}$ the approximation of the velocity t^{n+1}. It is possible to show that the above EFR algorithm is equivalent to a generic viscosity model in LES [41].

Remark 2.2. In this paper, we consider a simplified filter problem with respect to our previous work [14,16,21,17], where we forced the filtered velocity $\overline{\boldsymbol{v}}^{n+1}$ to be solenoidal. We are releasing this constraint as it leads to a substantial simplification and computational time savings since there is one less variable (i.e., the Lagrange multiplier to enforce the incompressibility constraint). As noted in [42], the incompressibility is exactly preserved by the simplified differential filter (10) only for periodic conditions. Thus, in our case the end-of-step velocity $\boldsymbol{u}^{n+1}$ does not strictly satisfy mass conservation. However, we will show in Sec. 4.1 that at discrete level the mass conservation error is very low.

Remark 2.3. The EFR method has an appealing advantage over other LES models: it is modular, i.e., it adds a differential problem to the Navier-Stokes problem instead of extra terms in the Navier-Stokes equations themselves (like, e.g., the popular variational multiscale approach [43]). Thus, thanks to the EF method anybody with a Navier-Stokes solver could simulate higher Reynolds number flows without major modifications to the software core. The reader interested in learning about the accuracy of the EFR algorithm when compared to experimental data from [44] is referred to [14,16]. Comparison between the same data and other LES and RANS approaches can be found in [45].

Finally, we note that while we consider homogeneous Neumann boundary conditions non-homogeneous boundary condition can of course be handled. See [14] to learn more about this.

2.2. Space discrete problem: a finite volume approximation

For the space discretization of problems (6)-(9) and (10)-(14), we adopt a Finite Volume method. We partition the computational domain Ω into cells or control volumes Ω_i, with $i = 1, \ldots, N_c$, where N_c is the total number of cells in the mesh. Let $\mathbf{A}_j$ be the surface vector of each face of the control volume, with $j = 1, \ldots, M$.

The fully discretized form of problem (6)-(7) reads: Find $(\boldsymbol{v}_i^{n+1}, p_i^{n+1})$ such that

$$\rho \frac{3}{2\Delta t} \boldsymbol{v}_i^{n+1} + \rho \sum_j \varphi_j^n \boldsymbol{v}_{i,j}^{n+1} - 2\mu \sum_j (\nabla \boldsymbol{v}_i^{n+1})_j \cdot \mathbf{A}_j + \sum_j p_{i,j}^{n+1} \mathbf{A}_j = \boldsymbol{b}_i^{n+1} \tag{16}$$

$$\sum_j (\nabla p^{n+1})_j \cdot \mathbf{A}_j = \sum_j (\mathbf{H}(\boldsymbol{v}_i^{n+1}))_j \cdot \mathbf{A}_j, \tag{17}$$

where:

$$\mathbf{H}(\boldsymbol{v}_i^{n+1}) = -\rho \sum_j \varphi_j^n \boldsymbol{v}_{i,j}^{n+1} + 2\mu \sum_j (\nabla \boldsymbol{v}_i^{n+1})_j \cdot \mathbf{A}_j + \boldsymbol{b}_i^{n+1} \quad \text{with} \quad \varphi_j^n = \boldsymbol{v}_j^n \cdot \mathbf{A}_j. \tag{18}$$

In (16)-(18), $\boldsymbol{v}_i^{n+1}$ and $\boldsymbol{b}_i^{n+1}$ denote the average velocity and source term in control volume Ω_i, respectively. Moreover, we denote with $\boldsymbol{v}_{i,j}^{n+1}$ and $p_{i,j}^{n+1}$ the velocity and pressure associated with the centroid of face j normalized by the volume of Ω_i. For the solution of the linear system associated with (16)-(17) we used the PISO algorithm [46]. The advantage of this algorithm is the decoupling of the computation of the pressure from the computation of the velocity, which results in low computational costs.

Next, we discretize filter problem (10). We obtain:

$$\overline{\boldsymbol{v}}_i^{n+1} - \alpha^2 \sum_j a_j^{n+1} (\nabla \overline{\boldsymbol{v}}_i^{n+1})_j \cdot \mathbf{A}_j = \boldsymbol{v}_i^{n+1}, \tag{19}$$

where $\overline{\boldsymbol{v}}_i^{n+1}$ is the average value of $\overline{\boldsymbol{v}}^{n+1}$ in control volume Ω_i and $a_j^{n+1} = a(\boldsymbol{v}_j^{n+1})$. To compute a_j^{n+1}, we need to solve the Helmholtz filter problem (14). Once discretized, it reads: Find the average value of $\tilde{\boldsymbol{v}}^{n+1}$ in Ω_i, i.e., $\tilde{\boldsymbol{v}}_i^{n+1}$, such that:

$$\tilde{\boldsymbol{v}}_i^{n+1} - \alpha^2 \sum_j (\nabla \tilde{\boldsymbol{v}}_i^{n+1})_j \cdot \mathbf{A}_j = \boldsymbol{v}_i^{n+1}. \tag{20}$$

Obviously, problems (19) and (20) are easier to solve than problem (16)-(17), i.e., the filter problem is computationally much less demanding than the Navier-Stokes problem.

3. The reduced order model

The Reduced Order Model we propose can be seen as an extension to an LES framework of the model introduced in [26] for RANS. The key idea is the following: we use a Galerkin projection method related to the computation of the reduced velocity and pressure fields, while we use an interpolation procedure based on Radial Basis Functions (RBF) for the computation of the reduced coefficients of the indicator function. We call this hybrid approach *data-driven POD-Galerkin ROM*. In Sec 3.1 we describe the details of our approach and in Sec. 3.2 we present the strategy we choose for pressure stabilization at reduced order level.

The ROM computations have been carried out using ITHACA-FV [47], an in-house open source C++ library.

3.1. Our data-driven POD-Galerkin method

We approximate velocity fields $\boldsymbol{v}$ and $\overline{\boldsymbol{v}}$, pressure field p, and indicator function a as linear combinations of the dominant modes (basis functions), assumed to depend on space variables only, multiplied by scalar coefficients that depend only on time:

$$\boldsymbol{v} \approx \boldsymbol{v}_r = \sum_{i=1}^{N_{v_r}} \beta_i(t) \boldsymbol{\varphi}_i(\boldsymbol{x}), \quad p \approx p_r = \sum_{i=1}^{N_{p_r}} \gamma_i(t) \psi_i(\boldsymbol{x}), \tag{21}$$

$$\overline{\boldsymbol{v}} \approx \overline{\boldsymbol{v}}_r = \sum_{i=1}^{N_{\overline{v}_r}} \overline{\beta_i}(t) \boldsymbol{\varphi}_i(\boldsymbol{x}), \quad a \approx a_r = \sum_{i=1}^{N_{a_r}} \delta_i(t) \eta_i(\boldsymbol{x}). \tag{22}$$

In (21)-(22), N_{Φ_r} denotes the cardinality of a reduced basis for the space Φ belongs to.

Using (21) to approximate $\boldsymbol{v}^{n+1}$ and p^{n+1} in (6)-(7), we obtain

$$\rho \frac{3}{2\Delta t} \boldsymbol{v}_r^{n+1} + \rho \nabla \cdot \left(\boldsymbol{v}_r^n \otimes \boldsymbol{v}_r^{n+1} \right) - 2\mu \Delta \boldsymbol{v}_r^{n+1} + \nabla p_r^{n+1} = \boldsymbol{b}_r^{n+1}, \tag{23}$$

$$\nabla \cdot \boldsymbol{v}_r^{n+1} = 0. \tag{24}$$

Then, using (22) to approximate $\overline{\boldsymbol{v}}^{n+1}$ and $a(\boldsymbol{v}^{n+1})$ in (10) we get:

$$\overline{\boldsymbol{v}}_r^{n+1} - \alpha^2 \nabla \cdot (a_r^{n+1} \nabla \overline{\boldsymbol{v}}_r^{n+1}) = \boldsymbol{v}_r^{n+1}. \tag{25}$$

Remark 3.1. As mentioned above, we use a data-driven interpolation for the approximation of indicator function a defined in (13). For this reason, we do not need to compute a reduced order approximation of $\tilde{\boldsymbol{v}}$.

Remark 3.2. We use the reduced basis $\boldsymbol{\varphi}_j$ associated with $\boldsymbol{v}$ also for the approximation of $\overline{\boldsymbol{v}}$ in (22). Thus, in the ROM velocity $\overline{\boldsymbol{v}}$ is divergence free, although the same is not true in the FOM.

In the literature, one can find several techniques to generate the reduced basis spaces, e.g., Proper Orthogonal Decomposition, the Proper Generalized Decomposition and the Reduced Basis with a greedy sampling strategy. See, e.g., [48–50,2,51–53,6]. We choose to find the reduced basis by using the method of snapshots. For this purpose, we solve the FOM described in Sec. 2 for each time $t^k \in \{t^1, \ldots, t^{N_s}\} \subset (t_0, T]$. The snapshots matrices are obtained from the full-order snapshots:

$$\boldsymbol{\mathcal{S}}_\Phi = [\Phi(t^1), \ldots, \Phi(t^{N_s})] \in \mathbb{R}^{N_{\Phi_h} \times N_s} \quad \text{for} \quad \Phi = \{\boldsymbol{v}, p, a\}, \tag{26}$$

where the subscript h denotes a solution computed with the FOM and N_{Φ_h} is the dimension of the space Φ belong to for the FOM. Note that Φ could be either a scalar or a vector field. The POD problem consists in finding, for each value of the dimension of the POD space $N_{POD} = 1, \dots, N_s$, the scalar coefficients $c_1^1, \dots, c_1^{N_s}, \dots, c_{N_s}^1, \dots, c_{N_s}^{N_s}$ and functions $\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_{N_s}$, that minimize the error between the snapshots and their projection onto the POD basis. In the L^2-norm, we have

$$E_{N_{POD}} = \arg\min \sum_{i=1}^{N_s} ||\Phi_i - \sum_{k=1}^{N_{POD}} c_i^k \boldsymbol{\zeta}_k|| \quad \forall N_{POD} = 1, \dots, N_s$$
$$\text{with} \quad (\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j)_{L_2(\Omega)} = \delta_{i,j} \quad \forall i, j = 1, \dots, N_s. \tag{27}$$

It can be shown [54] that eq. (27) is equivalent to the following eigenvalue problem

$$\boldsymbol{\mathcal{C}}^{\Phi} \boldsymbol{Q}^{\Phi} = \boldsymbol{Q}^{\Phi} \boldsymbol{\Lambda}^{\Phi}, \tag{28}$$
$$\mathcal{C}_{ij}^{\Phi} = (\Phi(t^i), \Phi(t^j))_{L_2(\Omega)} \quad \text{for} \quad i, j = 1, \dots, N_s, \tag{29}$$

where $\boldsymbol{\mathcal{C}}^{\Phi}$ is the correlation matrix computed from the snapshot matrix $\boldsymbol{\mathcal{S}}_{\Phi}$, $\boldsymbol{Q}^{\Phi}$ is the matrix of eigenvectors and $\boldsymbol{\Lambda}^{\Phi}$ is a diagonal matrix whose diagonal entries are the eigenvalues of $\boldsymbol{\mathcal{C}}^{\Phi}$. Then, the basis functions are obtained as follows:

$$\boldsymbol{\zeta}_i = \frac{1}{N_s \Lambda_i^{\Phi}} \sum_{j=1}^{N_s} \Phi_j Q_{ij}^{\Phi}. \tag{30}$$

The POD modes resulting from the aforementioned methodology are:

$$L_{\Phi} = [\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_{N_{\Phi_r}}] \in \mathbb{R}^{N_{\Phi_h} \times N_{\Phi_r}}, \tag{31}$$

where $N_{\Phi_r} < N_s$ are chosen according to the eigenvalue decay. The reduced order model can be obtained through a Galerkin projection of the governing equations onto the POD spaces.

Let

$$M_{r_{ij}} = (\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j)_{L_2(\Omega)}, \quad A_{r_{ij}} = (\boldsymbol{\varphi}_i, \Delta \boldsymbol{\varphi}_j)_{L_2(\Omega)}, \quad B_{r_{ij}} = (\boldsymbol{\varphi}_i, \nabla \psi_j)_{L_2(\Omega)}, \tag{32}$$
$$P_{r_{ij}} = (\psi_i, \nabla \cdot \boldsymbol{\varphi}_j)_{L_2(\Omega)}, \quad G_{r_{ijk}} = (\boldsymbol{\varphi_i}, \nabla \cdot (\boldsymbol{\varphi_j} \otimes \boldsymbol{\varphi}_k))_{L_2(\Omega)}, \tag{33}$$

where $\boldsymbol{\varphi}_i$ and ψ_i are the basis functions in (21). The reduced algebraic system at time t^{n+1} for problem (23)-(24) is:

$$\rho \frac{3}{2\Delta t} \boldsymbol{M}_r \boldsymbol{\beta}^{n+1} + \rho (\boldsymbol{\beta}^n)^T \boldsymbol{G}_r \boldsymbol{\beta}^{n+1} - 2\mu \boldsymbol{A}_r \boldsymbol{\beta}^{n+1} + \boldsymbol{B}_r \boldsymbol{\gamma}^{n+1} =$$
$$\frac{\rho}{2\Delta t} \boldsymbol{M}_r \left((1-\chi)\left(4\boldsymbol{\beta}^n - \boldsymbol{\beta}^{n-1}\right) + \chi \left(4\overline{\boldsymbol{\beta}}^n - \overline{\boldsymbol{\beta}}^{n-1}\right)\right), \tag{34}$$
$$\boldsymbol{P}_r \boldsymbol{\beta}^{n+1} = 0, \tag{35}$$

where vectors $\boldsymbol{\beta}^{n+1}$ and $\boldsymbol{\gamma}^{n+1}$ contain the values of coefficients β_i and γ_i in (21) at time t^{n+1}.

Next, let

$$A_{r_{ijk}} = (\boldsymbol{\varphi}_i, \nabla \cdot \eta_j \nabla \boldsymbol{\varphi}_k)_{L_2(\Omega)}, \tag{36}$$

where η_i are the basis functions in (22). The reduced algebraic system at time t^{n+1} for problem (25) is

$$\boldsymbol{M}_r \overline{\boldsymbol{\beta}}^{n+1} - \alpha^2 (\boldsymbol{\delta}^{n+1})^T \boldsymbol{A}_r \overline{\boldsymbol{\beta}}^{n+1} = \boldsymbol{M}_r \boldsymbol{\beta}^{n+1}, \tag{37}$$

where vectors $\overline{\boldsymbol{\beta}}^{n+1}$ and $\boldsymbol{\delta}^{n+1}$ contain the values of coefficients $\overline{\beta}_i$ and δ_i in (22) at time t^{n+1}.

The coefficients $\delta_i(t)$ in (22) are computed with a data-driven approach that uses interpolation with Radial Basis Functions [55]. The interpolation procedure is carried out for each mode separately. Let η_i be the indicator function mode under consideration. Function $G_i(t)$ that interpolates $\delta_i(t)$ using RBF functions can be written as:

$$G_i(t) = \sum_{j=1}^{N_s} w_{i,j} \zeta_{i,j}(|t - t^j|), \quad \text{for} \quad i = 1, 2, \dots, N_{a_r}, \ t \in (t_0, T] \tag{38}$$

where $w_{i,j}$ are suitable weights and $\zeta_{i,j}$ are the radial basis functions, which are chosen to be Gaussian functions. We observe that $\zeta_{i,j}$ is centered at time t^j. In order to compute the weights $w_{i,j}$, we use the fact that G_i has to interpolate δ_i at time nodes t^k, i.e.:

$$G_i(t^k) = \delta_i(t^k) \quad \text{for} \quad k = 1, 2, \ldots, N_s.$$

The coefficients $\delta_i(t^k)$ are obtained from projecting the k-th snapshot onto the i-th mode:

$$\delta_i(t^k) = (a(t^k), \eta_i)_{L^2(\Omega)},$$

$a(t^k)$ being the k-th column of the snapshot matrix $\boldsymbol{\mathcal{S}}_a$ (26). Then, we have:

$$G_i(t^k) = \sum_{j=1}^{N_s} w_{i,j}\zeta_{i,j}\left(|t^k - t^j|\right) = \delta_i(t^k),$$

which can be written as a linear system:

$$\boldsymbol{A}_i^{\zeta}\boldsymbol{w}_i = \boldsymbol{Y}_i, \quad \text{with } (\boldsymbol{A}_i^{\zeta})_{kj} = \zeta_{i,j}(|t^k - t^j|). \tag{39}$$

System (39) is solved *offline* to get the weights $\boldsymbol{w}_i$. During the *online* phase, for every new time instant t^* we compute $\delta_i(t^*)$ given by:

$$\delta_i(t^*) \approx G_i(t^*) = \sum_{j=1}^{N_s} w_{i,j}\zeta_{i,j}\left(|t^* - t^j|\right).$$

The initial conditions for the ROM algebraic system (34)-(35), (37) are obtained with a Galerkin projection of the initial full order conditions onto the POD basis spaces:

$$\beta^0{}_i = (\boldsymbol{v}(\boldsymbol{x}, t_0), \boldsymbol{\varphi}_i)_{L_2(\Omega)},$$
$$\overline{\beta}^0{}_i = (\overline{\boldsymbol{v}}(\boldsymbol{x}, t_0), \boldsymbol{\varphi}_i)_{L_2(\Omega)}.$$

Finally, we use the lifting function method [21] to account for non-homogeneous Dirichlet boundary conditions. The velocity snapshots are modified according to:

$$\boldsymbol{v}_h' = \boldsymbol{v}_h - \sum_{j=1}^{N_{BC}} u_{BC_j}(t)\boldsymbol{\chi}_j(\boldsymbol{x}),$$
$$\overline{\boldsymbol{v}}_h' = \overline{\boldsymbol{v}}_h - \sum_{j=1}^{N_{BC}} u_{BC_j}(t)\boldsymbol{\chi}_j(\boldsymbol{x}),$$

where N_{BC} is the number of non-homogeneous Dirichlet boundary conditions, $\boldsymbol{\chi}(\boldsymbol{x})$ are the divergence free control functions that satisfy the boundary conditions, and u_{BC_j} are suitable temporal coefficients. The POD is applied to the snapshots satisfying the homogeneous boundary conditions and then the boundary value is added back:

$$\boldsymbol{v}_r = \sum_{j=1}^{N_{BC}} u_{BC_j}(t)\boldsymbol{\chi}_j(\boldsymbol{x}) + \sum_{i=1}^{N_{v_r}} \beta_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x}),$$
$$\overline{\boldsymbol{v}}_r = \sum_{j=1}^{N_{BC}} u_{BC_j}(t)\boldsymbol{\chi}_j(\boldsymbol{x}) + \sum_{i=1}^{N_{\overline{v}_r}} \overline{\beta_i}(t)\boldsymbol{\varphi}_i(\boldsymbol{x}).$$

3.2. Pressure field reconstruction and stability

In order to obtain a stable and accurate reconstruction of the pressure field at the reduced level, we choose to adopt the Poisson pressure equation method used, e.g., in [24,21,22]. To obtain the Poisson pressure equation, we take the divergence of eq. (6) and account for divergence free condition (7):

$$\Delta p^{n+1} = -\rho\,\nabla\cdot\left(\nabla\cdot\left(\boldsymbol{v}^n \otimes \boldsymbol{v}^{n+1}\right)\right) + \nabla\cdot\boldsymbol{b}^{n+1}, \tag{40}$$

with boundary conditions (8) and:

$$\partial_n p^{n+1} = -2\mu\boldsymbol{n}\cdot\left(\nabla\times\nabla\times\boldsymbol{v}^{n+1}\right) - \boldsymbol{n}\cdot\left(\rho\frac{3}{2\Delta t}\boldsymbol{v}^{n+1} - \boldsymbol{b}^{n+1}\right) \tag{41}$$

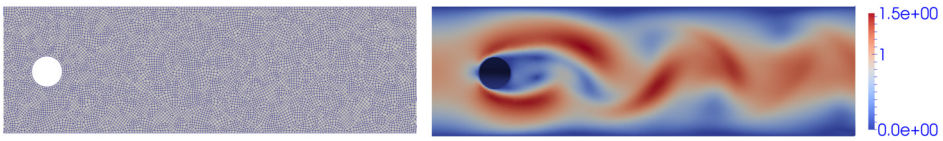


Fig. 1. 2D flow past a cylinder: (left) part of the mesh under consideration and (right) illustrative representation of the velocity field in part of the domain at $t = 6$.

on $\partial\Omega_N \times (t_0, T)$, where ∂_n denotes the derivative with respect to the normal vector $\boldsymbol{n}$. In eq. (40), we retain the term $\nabla \cdot \boldsymbol{b}^{n+1}$ because at full order level the filtered velocity is not divergence free. For further details about the derivation of non-homogeneous Neumann conditions for the pressure field, we refer the reader to [56,57].

By using (21) for the approximation of $\boldsymbol{v}^{n+1}$ and p^{n+1} in (40), we obtain

$$\Delta p_r^{n+1} = -\rho \nabla \cdot \left(\nabla \cdot \left(\boldsymbol{v}_r^n \otimes \boldsymbol{v}_r^{n+1} \right) \right). \tag{42}$$

The term $\nabla \cdot \boldsymbol{b}_r^{n+1}$ vanishes in eq. (42) because at reduced order level the filtered velocity is divergence free as explained in Remark 3.2.

The matrix form of eq. (42) reads:

$$\begin{aligned} &\boldsymbol{D}_r \boldsymbol{\gamma}^{n+1} + \rho \left(\boldsymbol{\beta}^n\right)^T \boldsymbol{J}_r \boldsymbol{\beta}^{n+1} - 2\mu \boldsymbol{N}_r \boldsymbol{\beta}^{n+1} - \frac{3\rho}{2\Delta t} \boldsymbol{F}_r \boldsymbol{\beta}^{n+1} \\ &\quad = \frac{\rho}{2\Delta t} \boldsymbol{F}_r \left((1-\chi) \left(4\boldsymbol{\beta}^n - \boldsymbol{\beta}^{n-1} \right) + \chi \left(4\overline{\boldsymbol{\beta}}^n - \overline{\boldsymbol{\beta}}^{n-1} \right) \right), \end{aligned} \tag{43}$$

where

$$D_{r_{ij}} = (\nabla \psi_i, \nabla \psi_j)_{L_2(\Omega)}, \quad N_{r_{ij}} = (\boldsymbol{n} \times \nabla \psi_i, \nabla \times \boldsymbol{\varphi}_j)_{L_2(\partial\Omega)}, \tag{44}$$

$$F_{r_{ij}} = (\psi_i, \boldsymbol{n} \cdot \boldsymbol{\varphi}_j)_{L_2(\partial\Omega)}, \quad J_{r_{ijk}} = (\nabla \psi_i, \nabla \cdot (\boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_k))_{L_2(\Omega)}. \tag{45}$$

To conclude, the ROM algebraic system that has to be solved at every time step is (34), (37) and (43).

4. Numerical results

We test our approach on two well-known test cases [31,30]: 2D and 3D flow past a cylinder at $0 \leq Re(t) \leq 100$. The Reynolds number depends on time since the maximum inflow velocity starts from zero at the beginning of the time interval and reaches its peak in the middle of such interval before returning to zero at the end of the simulation. Details are reported in Sec. 4.1 and 4.2. Our goal is a thorough assessment of our ROM model in the reconstruction of the time evolution of the flow field.

4.1. 2D flow past a cylinder

The computational domain is a 2.2×0.41 rectangular channel with a cylinder of radius 0.05 centered at (0.2, 0.2), when taking the bottom left corner of the channel as the origin of the axes. Fig. 1 (left) shows part of the computational domain. The channel is filled with fluid with density $\rho = 1$ and viscosity $\mu = 10^{-3}$. We impose a no-slip boundary condition on the upper and lower wall and on the cylinder. At the inflow, we prescribe the following velocity profile:

$$\boldsymbol{v}(0, y, t) = \left(\frac{6}{0.41^2} \sin(\pi t/8)\, y\, (0.41 - y), 0 \right), \quad y \in [0, 0.41], \quad t \in (0, 8], \tag{46}$$

and $\partial p / \partial \boldsymbol{n} = 0$. At the outflow we prescribe $\nabla \boldsymbol{v} \cdot \boldsymbol{n} = 0$ and $p = 0$. We start the simulations from fluid at rest.

The quantities of interest for this benchmark are the drag and lift coefficients [31,30]:

$$C_D(t) = \frac{2}{\rho L_r U_r^2} \int_S \left((2\mu \nabla \boldsymbol{u} - q\boldsymbol{I}) \cdot \boldsymbol{n} \right) \cdot \boldsymbol{t}\, dS, \tag{47}$$

$$C_L(t) = \frac{2}{\rho L_r U_r^2} \int_S \left((2\mu \nabla \boldsymbol{u} - q\boldsymbol{I}) \cdot \boldsymbol{n} \right) \cdot \boldsymbol{n}\, dS, \tag{48}$$

where $U_r = 1$ is the maximum velocity at the inlet/outlet, $L_r = 0.1$ is the cylinder diameter, S is the cylinder surface, and $\boldsymbol{t}$ and $\boldsymbol{n}$ are the tangential and normal unit vectors to the cylinder, respectively.

We have previously investigated this benchmark at the FOM level in [16] and at the ROM level in [21,23]. However, while at the FOM level we considered the EFR algorithm with a linear indicator function and with indicator function (13) [16],

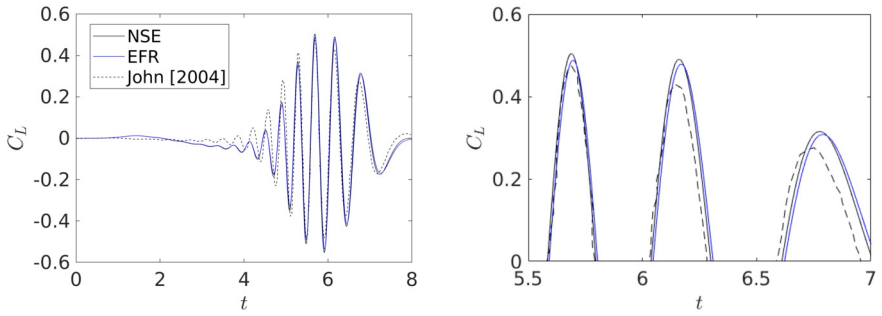


Fig. 2. 2D flow past a cylinder: (left) time evolution of the lift coefficient given by the NSE and EFR for $\alpha = h_{avg}$ and $\chi = \Delta t$ compared against the results in [31] and (right) close-up view.

Table 1
2D flow past a cylinder: maximum C_L and time instant at which it occurs given by NSE and EFR algorithms. Reference values from [31] are also reported.

	$t_{C_{L,max}}$	$C_{L,max}$
NSE	5.688	0.504
EFR	5.699	0.488
[31]	5.694	0.478

the ROM study is limited to the EFR algorithm with a linear indicator function [21,23]. In [16], we showed that at the FOM level the oscillatory pattern of the flow field (see Fig. 1 (right)) can be recovered only with a nonlinear indicator function. Thus, we expect the results obtained with the ROM presented in this paper to be far more accurate than the results in [21].

We consider a hexaedral computational grid with $h_{min} = 4.2e-3$, $h_{avg} = 7.5e-3$ and $h_{max} = 1.1e-2$ for a total of $1.59e4$ cells. The quality of the mesh is high: it features very low values of maximum non-orthogonality (36°), average non-orthogonality (4°), skewness (0.7), and maximum aspect ratio (2). Fig. 1 (left) shows a part of the mesh. We chose this mesh because it is the coarsest among all the meshes considered in [16], featuring only 8% of the number of elements needed to obtain an accurate solution [30,32]. Thus, it is the most challenging for our filtering approach.

4.1.1. Validation of the FOM

Before applying the ROM, we test the EFR algorithm at FOM level and compare its results with the ones produced by a NSE solver in OpenFOAM. For the convective term, we use a second-order accurate central difference scheme that features low dissipation [58]. This is a difference with respect to [16] where we used a second-order accurate upwind scheme. We set $\Delta t = 4e-4$ which allows to obtain $CFL_{max} \approx 0.25$ at the time when the velocity reaches its maximum value. We set $\chi = \Delta t$ since this is a reasonable choice for academic problems such as the one we are considering [13]. More realistic applications require a suitable formula to set χ [14,16]. We set $\alpha = h_{avg}$.

Fig. 2 (left) shows the evolution of C_L over time computed by EFR and NSE and a comparison with the results from [31]. Fig. 2 (right) shows a close-up of the time window next to the time of maximum C_L. We observe that the lift coefficient computed with EFR is slightly closer the reference results from [31]. To quantify this better agreement, we report in Table 1 the computed values of the maximum lift coefficient and the corresponding time instant, together with the values from [31]. Then, we can conclude that EFR is a little more accurate than NSE model when using a coarse mesh even at the low Reynolds numbers we are considering. For a more evident improvement introduced by EFR over NSE, see the 3D case in Sec. 4.2.

Finally, we show in Fig. 3 (left) the mass conservation error over time defined as follows:

$$\varepsilon_{\Phi} = \frac{1}{\Omega}\int_{\Omega} \nabla \cdot \Phi d\Omega \quad \text{for} \quad \Phi = \{\boldsymbol{v}, \boldsymbol{u}\}. \tag{49}$$

We see that although the incompressibility constraint for the filtered velocity $\overline{\boldsymbol{v}}$ is not enforced in the model, the mass conservation error for the end-of-step velocity $\boldsymbol{u}$ is comparable to the mass conservation error for the intermediate velocity $\boldsymbol{v}$ (which, instead, is divergence free). We also evaluated the conservation of mass at the selected axial locations x for $t = 6$ (i.e., when $\varepsilon_{\boldsymbol{u}}$ reaches its maximum value) using the error metric proposed in [59]:

$$E_{Q_{\Phi}} = \frac{Q_{\Phi} - \widetilde{Q}}{\widetilde{Q}} \quad \text{for} \quad \Phi = \{\boldsymbol{v}, \boldsymbol{u}\}, \tag{50}$$

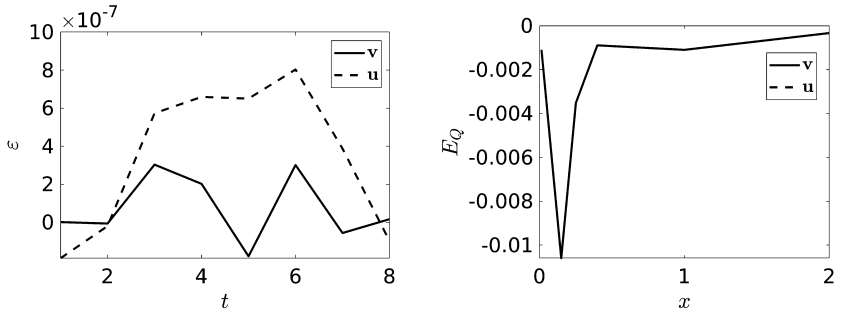


Fig. 3. 2D flow past a cylinder: (left) mass conservation error (49) for the intermediate velocity $\boldsymbol{v}$ and the end-of-step velocity $\boldsymbol{u}$ at selected times in the time interval of interest and (right) mass conservation error (50) for the intermediate velocity $\boldsymbol{v}$ and the end-of-step velocity $\boldsymbol{u}$ for $t = 6$ at selected axial locations.

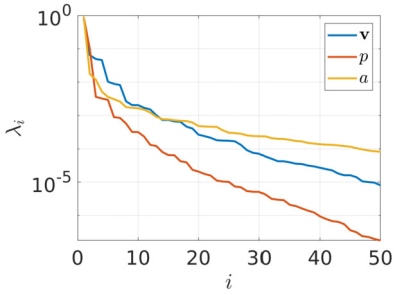


Fig. 4. 2D flow past a cylinder - experiment (i): eigenvalue decay for the intermediate velocity, pressure, and indicator function.

where $Q_{\boldsymbol{\Phi}}$ is the volumetric flow rate computed from the numerical axial velocity profiles and $\widetilde{Q}$ is the exact volumetric flow rate. Fig. 3 (right) shows $E_{Q_{\boldsymbol{v}}}$ and $E_{Q_{\boldsymbol{u}}}$, which are overlapped over the whole axis.

Next, we are going to validate our ROM approach. Since this benchmark is characterized by a flow field exhibiting a wide spectrum of frequencies, we adopt an idea proposed in [23]: we test the performances of our ROM approach (i) over the entire time window of interest $[0, 8]$ and (ii) over the second half of the time interval $[4, 8]$ where the high-frequency modes are dominant.

4.1.2. Validation of the ROM (i)

We collect 400 FOM snapshots, one every 0.02 s, i.e., we use an equispaced grid in time. Fig. 4 shows the eigenvalue decay for the intermediate velocity, pressure, and indicator function. For the ROM simulations, we collected the solutions every 0.01 s. This means that the set of time samples includes the samples used in the offline stage and samples in between two consecutive offline samples. The reason for this choice is that we want to assess how accurate the reduced order approximation is for time instants that were not in the training set.

We performed a convergence test as the number of modes increases. We considered three different energy thresholds using the first 50 most energetic POD modes: 99% (11 modes for $\boldsymbol{v}$, 4 modes for p and 15 modes for a), 99.9% (26 modes for $\boldsymbol{v}$, 10 modes for p and 41 modes for a) and 99.99% (43 modes for $\boldsymbol{v}$, 22 modes for p and 50 modes for a). We calculated the L^2 relative error:

$$E_{\Phi}(t) = \frac{||\Phi_h(t) - \Phi_r(t)||_{L^2(\Omega)}}{||\Phi_h(t)||_{L^2(\Omega)}}, \tag{51}$$

where Φ_h is a field computed with the FOM ($\boldsymbol{u}_h$, p_h or a_h) and Φ_r is the corresponding field computed with the ROM ($\boldsymbol{u}_r$, p_r or a_r). Fig. 5 shows errors (51) and Table 2 reports minimum, average, and maximum relative errors for 99.99% of the cumulative energy. From Fig. 5, we observe that the majority of the relative errors is less than 1 at all the time instants. The only exception is the velocity: we see that the error increases steeply past $t \approx 4$ and its value reaches 1 towards the end of the time interval. The relative error for the velocity and pressure is significantly lower for $t < 4$. Both errors increase when the vortex shedding starts at around $t = 4$. The relative error associated with the indicator function seems to be less critical since its value remains below 10^{-1} for most of the time interval of interest for 99.9% and 99.99% of the cumulative energy. Larger errors for the indicator function at the beginning of simulation might be due to the transient nature of the flow. This different behavior for the velocity and pressure errors on one side and the indicator function on the other side could be explained by the fact that different strategies are used for the ROM reconstruction (a projection method for $\boldsymbol{u}$ and p and an interpolation procedure for a). Indeed, for what concerns the velocity, by moving from 99%, to 99.9% and 99.99%, it becomes lower. We observe that going from 99% to 99.9% of the cumulative energy there is a general improvement of the errors,

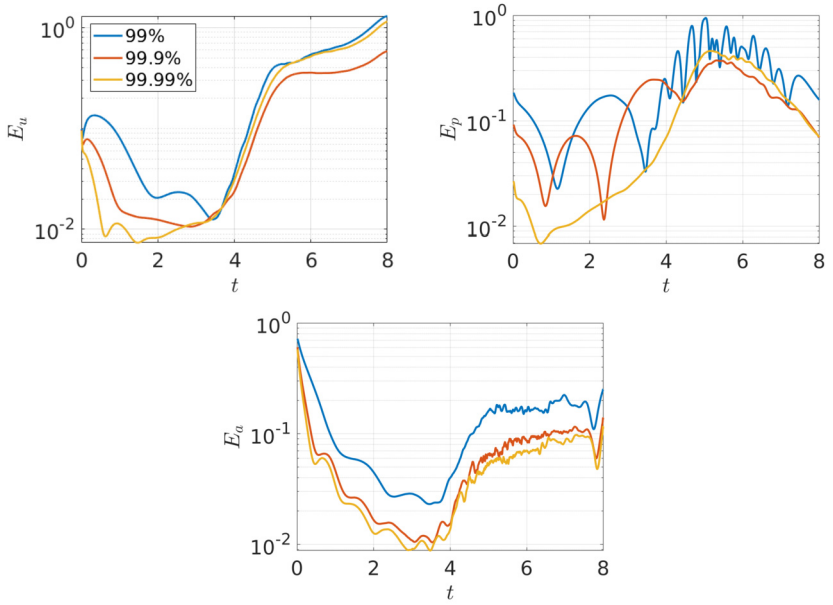


Fig. 5. 2D flow past a cylinder - experiment (i): time history of the relative L^2 error (51) for velocity $\boldsymbol{u}$ (top left), pressure p (top right), and for the indicator function a (bottom) for different thresholds of cumulative energy.

Table 2
2D flow past a cylinder - experiment (i): maximum, average, and minimum relative L^2 errors for the end-of-step velocity, pressure, and indicator function for 99.99% of the cumulative energy. The first three columns refer to the first half of the time interval (i.e., [0, 4]), while the last three refer to the second half of the time interval (i.e., [4, 8]).

	$\boldsymbol{u}$	p	a	$\boldsymbol{u}$	p	a
Maximum E_Φ	0.09	0.07	0.58	1.15	0.46	0.12
Average E_Φ	0.01	0.02	0.04	0.5	0.26	0.06
Miminum E_Φ	0.007	0.007	0.01	0.03	0.07	0.01

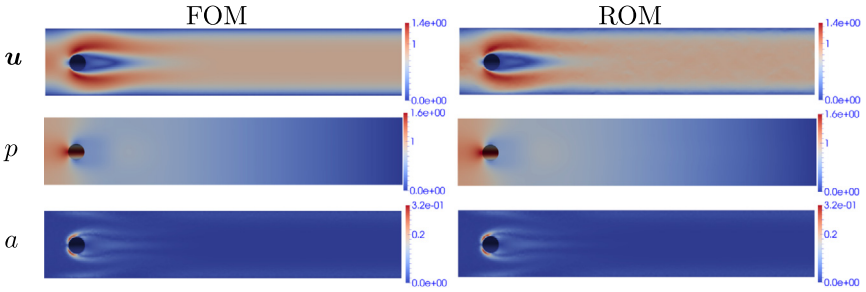


Fig. 6. 2D cylinder problem - experiment (i): comparison between velocity $\boldsymbol{u}$ (first row), pressure (second row), and indicator function (3rd row) computed by the FOM (left) and the ROM (right) at $t = 1.9$ for 99.99% of the cumulative energy.

while such improvement lessens when going from 99.9% to 99.99%. From Fig. 5 (top right), we see that the oscillations in the pressure error for $t > 4$ are damped when a larger amount of energy snapshots is retained.

Figs. 6 and 7 display a qualitative comparison between the computed FOM and ROM fields at two different times: $t = 1.9$ (first half of the time interval) and $t = 5.5$ (second half). As we can see from Fig. 6, our ROM provides a good reconstruction of all the variables at $t = 1.9$. On the other hand, the ROM fails to provide an accurate approximation of velocity and pressure at $t = 5.5$, as shown in Fig. 7.

Fig. 8 reports a more quantitative comparison: the aerodynamics coefficients (47)-(48) computed by FOM and ROM with different thresholds of cumulative energy. We observe that the time evolution of the drag coefficient is correctly reconstructed by ROM, while the ROM reconstruction of the lift coefficient is accurate till about $t = 4$. For $t > 4$, the lift coefficient computed by the ROM is off in terms of both phase and amplitude regardless of the percentage of retained energy.

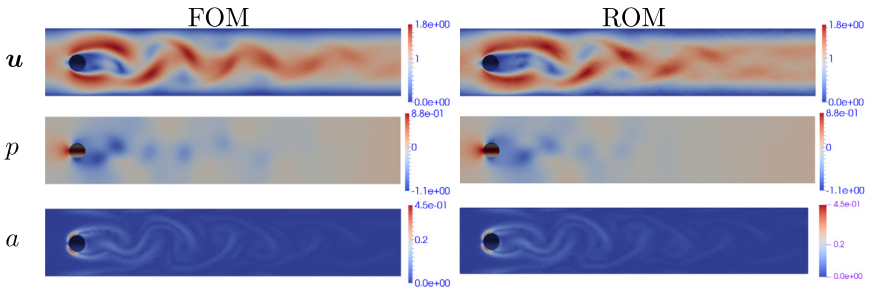


Fig. 7. 2D cylinder problem - experiment (i): comparison between velocity $\boldsymbol{u}$ (first row), pressure (second row), and indicator function (3rd row) computed by the FOM (left) and the ROM (right) at $t = 5.5$ for 99.99% of the cumulative energy.

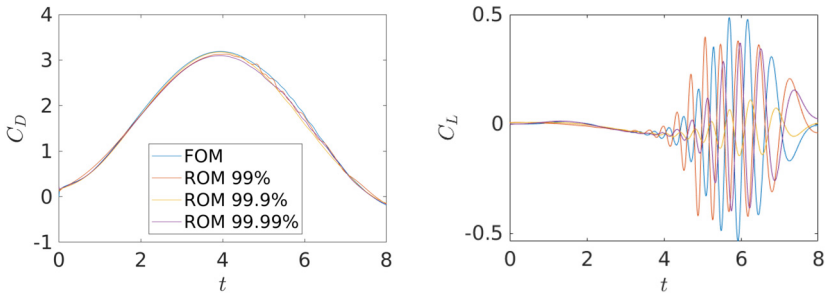


Fig. 8. 2D flow past a cylinder - experiment (i): aerodynamic coefficients C_D (left) and C_L (right) computed by FOM and ROM for different thresholds of cumulative energy.

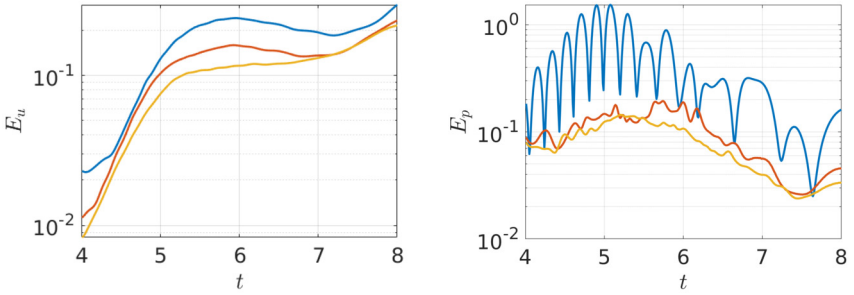


Fig. 9. 2D flow past a cylinder - experiment (ii): time history of the relative L^2 error (51) for velocity $\boldsymbol{u}$ (left) and pressure p (right) for different thresholds of cumulative energy.

4.1.3. Validation of the ROM (ii)

In this second experiment, we focus on the second half of the time interval of interest, $[4, 8]$ with the goal of improving the ROM reconstruction of velocity and pressure in that time window. We use the same database of FOM snapshots used in experiment (i), but we apply the POD only to the last 200 high-fidelity snapshots, i.e., those related to $[4, 8]$. Just like in experiment (i), for the ROM simulations we included both samples used in the training phase (i.e., the offline samples) and samples in between the offline samples.

Also for this experiment, we considered the first 50 most energetic POD modes and performed a convergence test based on three different energy thresholds: 99% (11 modes for $\boldsymbol{v}$, 5 modes for p and 20 modes for a), 99.9% (26 modes for $\boldsymbol{v}$, 12 modes for p and 44 modes for a), and 99.99% (42 modes for $\boldsymbol{v}$, 24 modes for p and 50 modes for a). Fig. 9 shows errors (51) and Table 3 reports minimum, average, and maximum relative errors for 99.99% of the cumulative energy. We observe that the relative error for the velocity reaches much lower values than in experiment (i): it drops below 0.22 during the entire time interval when the 99.99% of the snapshots energy is retained. Moreover, there is a monotonic convergence as the number of the modes is increased. The improvement of the pressure reconstruction is also rather significative: compare Fig. 5 (top right) with Fig. 9 (left). Once again, we note that there is not much difference in the relative errors for velocity and pressure when going from 99.9% to 99.99% of the cumulative energy.

Fig. 10 shows the comparison between the computed FOM and ROM fields at $t = 5.5$. The ROM reconstructions of velocity and pressure are much more accurate than in experiment (i). Fig. 11 reports the quantitative FOM/ROM comparison for the lift coefficient C_L. The improvement with respect to experiment (i) is evident: the time evolution of C_L computed

Table 3
2D flow past a cylinder - experiment (ii): maximum, average, and minimum relative L^2 errors for the end-of-step velocity and pressure for 99.99% of the cumulative energy.

	$\boldsymbol{u}$	p
Maximum E_Φ	0.22	0.14
Average E_Φ	0.11	0.08
Miminum E_Φ	0.008	0.02

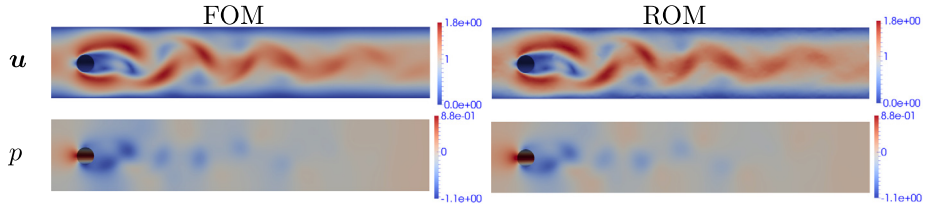


Fig. 10. 2D cylinder problem - experiment (ii): comparison between velocity $\boldsymbol{u}$ (first row) and pressure (second row) computed by the FOM (left) and the ROM (right) at $t = 5.5$ for 99.99% of the cumulative energy.

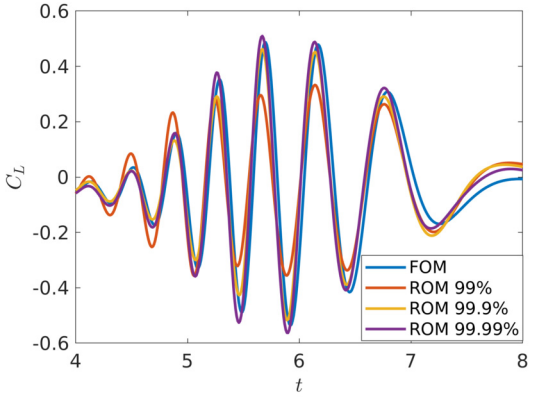


Fig. 11. 2D flow past a cylinder - experiment (ii): lift coefficient C_L computed by FOM and ROM for different thresholds of cumulative energy.

by the ROM is very accurate when the 99.9% or 99.99% of the energy is retained. For a further quantitative assessment, we computed the following error

$$E_{C_L} = \frac{||C_L(t)^{FOM} - C_L(t)^{ROM}||_{L^2(4,8)}}{||C_L(t)^{FOM}||_{L^2(4,8)}}. \tag{52}$$

We obtain $E_{C_L} = 0.5, 0.39$, and 0.38, for 99%, 99.9% and 99.99% of the cumulative energy, respectively.

We conclude by proving some information about the efficiency of our ROM approach. In Fig. 12, we show the averaged relative error for velocity and pressure fields in time versus the relative wall time. The relative wall time is given by the ratio between the CPU time needed by the ROM and the CPU time required by a FOM simulation. The FOM simulation takes 398 s. In the ROM CPU time, we include the time required to solve the reduced algebraic systems (34), (37), (43) and to reconstruct and save fields (21)-(22). We fix $N_{p_r} = 24$ and $N_{a_r} = 50$ and let N_{v_r} vary. As expected, we see that in general the relative errors decrease and the relative wall time increases as N_{v_r} increases. The relative errors reach their minimum value for $N_{v_r} = 46$. For N_{v_r} = 50, we observe that the relative wall time is larger than 1, which means that the ROM is more costly than the FOM and thus it becomes pointless. In addition, we see that for N_{v_r} = 50 the relative errors are larger than for $N_{v_r} = 46$. Recall that we retain 99.99% of the total energy with 42 modes. So, the increased errors for N_{v_r} = 50 might be due to the fact that the highest modes could be noisy and therefore spoil the solution.

4.2. 3D flow past a cylinder

The 3D benchmark we consider has been studied for the first time in [30] and further investigated in [33,32]. The computational domain is a 2.5 × 0.41 × 0.41 parallelepiped with a cylinder whose axis is parallel to the z-axis and center is located at (0.5, 0.2) when taking the bottom left corner of the channel as the origin of the axes. Fig. 13 (left) shows

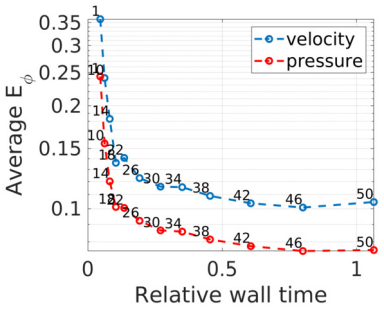


Fig. 12. 2D flow past a cylinder: Pareto plots for the velocity (blue) and pressure (red) fields - averaged relative error in time versus relative wall time for varying N_{v_r}. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

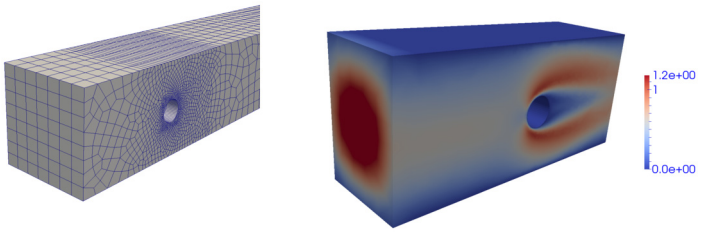


Fig. 13. 3D flow past a cylinder: (left) part of the mesh under consideration and (right) illustrative representation of the velocity field for $t = 6$.

Table 4
3D flow past a cylinder: maximum lift coefficient given by NSE and EFR with $\alpha = h_{avg}$ and $\chi = \Delta t$, together with reference values from [30,33,32].

	max C_L
NSE	0.0011
EFR	0.0018
Ref. values [30,33,32]	[0.002, 0.004]

part of the computational domain. The channel is filled with fluid with density $\rho = 1$ and viscosity $\mu = 0.001$. We impose a no-slip boundary condition on the channel walls and on the cylinder. At the inflow, we prescribe the following velocity profile:

$$\boldsymbol{u}(0, y, z, t) = \left(\frac{36}{0.41^4} \sin(\pi t/8)\, yz\,(0.41 - y)\,(0.41 - z), 0, 0 \right), \tag{53}$$

for $y, z \in [0, 0.41]$ and $t \in (0, 8]$. In addition, on the channel walls, cylinder, and at the inlet we impose $\partial p/\partial \boldsymbol{n} = 0$ where $\boldsymbol{n}$ is the outward normal. At the outflow, we prescribe $\nabla \boldsymbol{u} \cdot \boldsymbol{n} = 0$ and $p = 0$. Note that the Reynolds number is time-dependent, with $0 \leq Re \leq 100$ [30,33,32]. Like for the 2D benchmark, we start the simulations from fluid at rest.

We consider a hexahedral grid with $h_{min} = 9e-3$, $h_{avg} = 3.75e-2$, and $h_{max} = 6.6e-2$ and a total of 1.07e4 cells. The mesh features very low values of maximum non-orthogonality (34°), average non-orthogonality (7°), skewness (0.6), and maximum aspect ratio (25). In addition, the mesh is refined next to the cylinder, like the meshes used in [33,32]. However, notice that this level of refinement is very far from the one required by a DNS [33,32], i.e., roughly 3 million elements. To underline the need for EFR with the given severely under-refined mesh, Fig. 14 shows the evolution of the lift coefficient computed by the NSE solver and EFR with $\alpha = h_{avg}$ and $\chi = \Delta t = 5e-3$ [13,21]. Like in the 2D case, we use a second-order accurate Central Differencing scheme [58] for the discretization of the convective term. From Fig. 14, we see that the EFR algorithm dampens the unphysical oscillations in the NSE solution and increases the maximum lift coefficient. Table 4 reports the computed values of the maximum lift coefficient and compares them with the reference values from [30,33,32]. Fig. 14 and Table 4 show that the EFR method outperforms the NSE solver even at the relatively small values of Re under consideration, which nonetheless require a very fine mesh. The EFR method allows for the use of much coarser meshes without sacrificing accuracy, thereby reducing the computational burden at full order level. The results of the NSE solver in Fig. 14 could be improved by, e.g., changing from a Central Differencing scheme to an upwind scheme such as WENO. However, that is beyond the scope of this article.

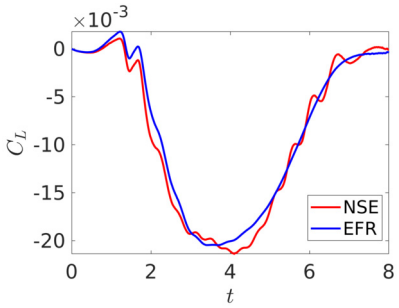


Fig. 14. 3D flow past a cylinder: time evolution of the lift coefficient given by the NSE with EFR for $\alpha = h_{avg}$ and $\chi = \Delta t$.

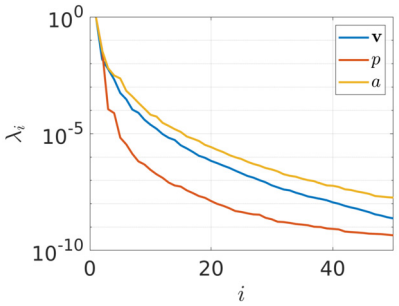


Fig. 15. 3D flow past a cylinder: eigenvalue decay for the intermediate velocity, pressure, and indicator function.

Table 5
3D flow past a cylinder: maximum, average, and minimum relative L^2 errors (51) for the end-of-step velocity, pressure, and indicator function.

	$\boldsymbol{u}$	p	a
Maximum E_Φ	0.12	0.18	0.24
Average E_Φ	0.02	0.06	0.01
Miminum E_Φ	0.01	0.005	0.004

As well documented in the literature [30,33,32], the main difference with respect to the 2D test case is that the flow field does not exhibit an oscillatory pattern, as shown in Fig. 13 (right). Thus, we perform only one numerical experiment and test the performances of our ROM approach over the entire time window of interest [0 8] only.

We collect 400 FOM snapshots for the training in the offline phase, one every 0.02 s (equispaced grid in time). Fig. 15 shows the eigenvalues decay for velocity $\boldsymbol{v}$, pressure and indicator function. By a comparison with Fig. 4, we see that the decay is faster for all the variables in the 3D case. Therefore, a smaller number of basis functions needs to be considered. This is due to the fact that the 2D flow is more complex.

For the ROM simulations, we collected a set of 800 equispaced temporal instants (i.e., one every 0.01 s), which includes both samples used in the offline stage and samples in between. In order to retain 99.99% of the snapshots energy, we needs 9 modes for $\boldsymbol{v}$, 4 modes for p and 12 modes for a. Fig. 16 shows errors (51) and Table 5 reports minimum, average, and maximum relative errors. The relative errors for the velocity $\boldsymbol{u}$ and the indicator functions stay below 10^{-1} for most of the time interval, except at the beginning and the end of the simulation, i.e., when the flow pattern is affected by transient effects. As for the pressure, we see a relative error larger than 10^{-1} at $t \approx 6$. These results are qualitatively similar to the ones showed by a Leray model in [21]. This is expected, since the EFR algorithm can be seen as a splitting scheme for the Leray model [14].

Figs. 17 and 18 display a qualitative comparison for velocity $\boldsymbol{u}$, pressure, and indicator function computed by FOM and ROM on the midsection ($z = 0.205$) at times $t = 1.9$ and $t = 5.5$, respectively. Our ROM provides a good reconstruction of all the variables at both times. For a further comparison, Figs. 19 and 20 show the profiles of all the variables at $t = 4$ along a line upstream of the cylinder ($x = 0.25$, $y \in [0\ 0.41]$, $z = 0.205$) and a line downstream of the cylinder ($x = 0.55$, $y \in [0\ 0.41]$, $z = 0.205$), respectively. We see great agreement between FOM and ROM profiles for all the variables with the exception of the pressure in Fig. 19 (top right), for which we observe a small difference.

Fig. 21 reports the drag and lift coefficients computed by FOM and ROM. As for the 2D case, we see that C_D is well reconstructed by the ROM while the reconstruction of the time evolution of C_L is not as accurate. Error (52) is $E_{C_L} = 0.25$.

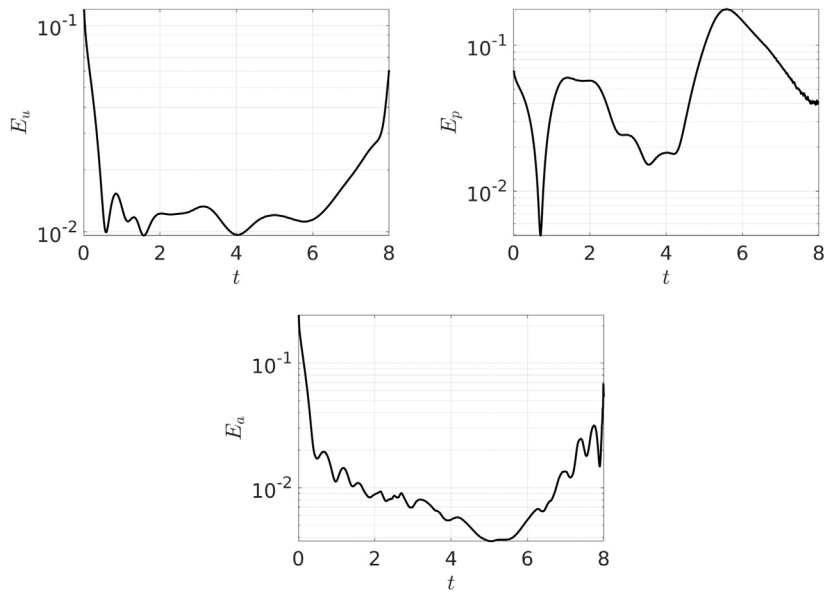


Fig. 16. 3D flow past a cylinder: time history of L^2 norm of the relative error (51) for velocity $\boldsymbol{u}$ (top left), pressure field (top right), and indicator function field (bottom).

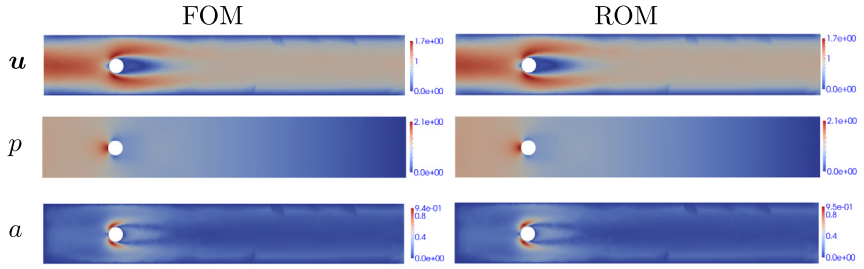


Fig. 17. 3D flow past a cylinder: comparison between velocity $\boldsymbol{u}$ (first row), pressure (second row), and indicator function (third row) computed by the FOM (left) and the ROM (right) on the midsection ($z = 0.205$) at $t = 1.9$.

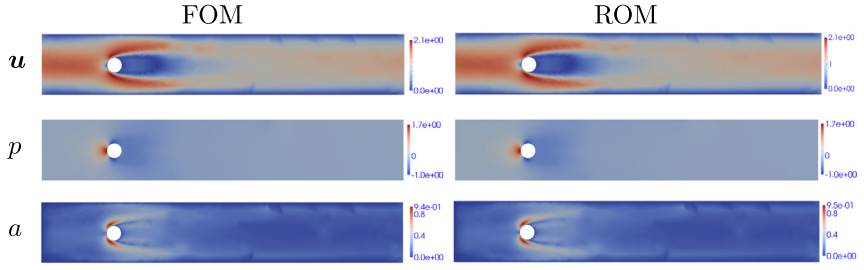


Fig. 18. 3D flow past a cylinder: comparison between velocity $\boldsymbol{u}$ (first row), pressure (second row), and indicator function (third row) computed by the FOM (left) and the ROM (right) on the midsection ($z = 0.205$) at $t = 5.5$.

A better reconstruction of C_L can be obtained by increasing the number of modes for the pressure, as shown in Fig. 21 (right). When switching from $N_{p_r} = 4$ to $N_{p_r} = 9$, the error E_{C_L} decreases from 0.25 to 0.13.

To stress the importance of the differential filter in the online phase, Fig. 22 shows a comparison of C_L computed by the ROM proposed in this paper, denoted with EFR-EFR since the EFR algorithm is used in both the online and the offline phase, and a ROM that does not use the differential filter in the online phase, denoted with EFR-NSE. The latter method corresponds to having no ROM closure model. Note that for both ROMs the snapshots are computed with EFR since a NSE solver is not accurate with the coarse mesh under consideration (see Fig. 14). From Fig. 22, we see that the ROM proposed in this paper provides a more accurate reconstruction of the C_L than the EFR-NSE ROM. Hence, although the Reynolds number for this test case is low, the use of the differential filter in the offline *and* online phase leads to a significant improvement of the ROM.

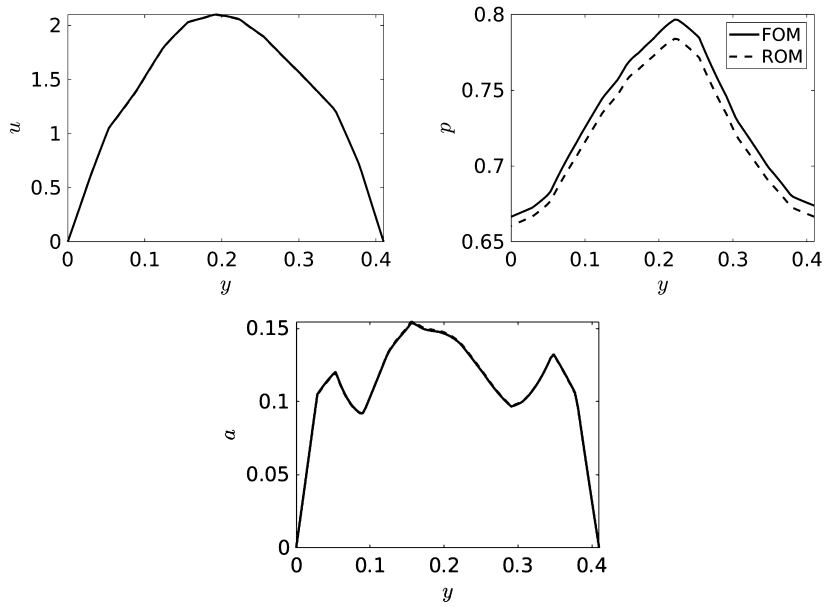


Fig. 19. 3D flow past a cylinder: velocity (top left), pressure (top right), and indicator function (bottom) profiles computed by FOM and ROM along line $x = 0.25$, $y \in [0, 0.41]$, $z = 0.205$ at $t = 4$.

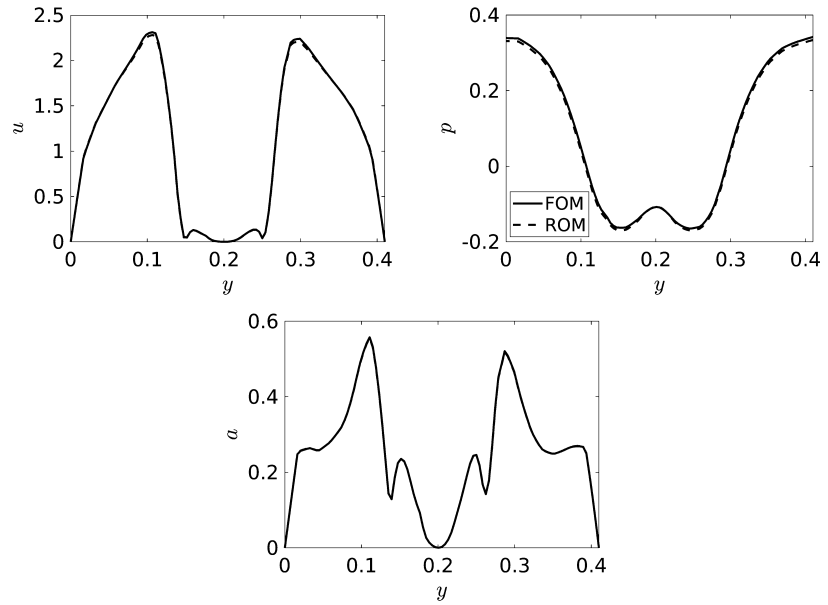


Fig. 20. 3D flow past a cylinder: velocity (top left), pressure (top right), and indicator function (bottom) profiles computed by FOM and ROM along line $x = 0.55$, $y \in [0, 0.41]$, $z = 0.205$ at $t = 4$.

Finally, we comment on the computational costs. The FOM simulation takes 190 s. In Fig. 23, we show the averaged relative error for velocity and pressure fields in time versus the relative wall time. We fix $N_{p_r} = 4$ and $N_{a_r} = 12$ and vary N_{v_r}. We observe that the relative errors reach their minimum value or $N_{v_r} = 30$. The increased errors for larger N_{v_r} could be explained as in the 2D case, i.e., the highest modes could contain noise and spoil the ROM solution.

5. Conclusions and perspectives

This work presents a hybrid Reduced Order Method or the Evolve-Filter-Relax algorithm with a nonlinear indicator function: we exploit a data-driven reduction method based on interpolation with Radial Basis Functions to approximate the indicator function and a classical POD-Galerkin projection approach for the reconstruction of the velocity and the pressure fields. This mixed strategy has been recently used for the RANS equations and here we have extended it to an LES

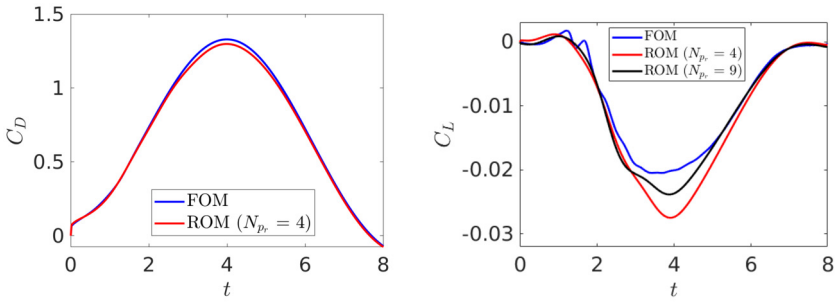


Fig. 21. 3D flow past a cylinder: aerodynamic coefficients C_D (left) and C_L (right) computed by FOM and ROM. The ROM approximation of C_L is improved as N_{p_r} is increased from 4 to 9. The ROM approximation of C_D for $N_{p_r} = 9$ is not reported because it marginally improves the approximation obtained with $N_{p_r} = 4$.

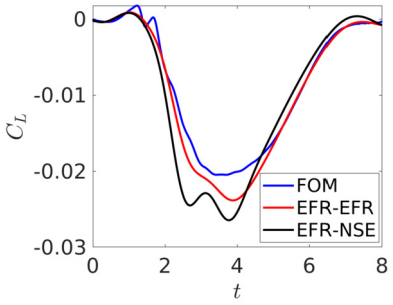


Fig. 22. 3D flow past a cylinder: lift coefficient C_L computed by FOM and two ROMs, the ROM proposed in this paper (EFR-EFR) and a ROM with no differential filter in the online phase (EFR-NSE). For the ROMs, we set $N_{p_r} = 9$.

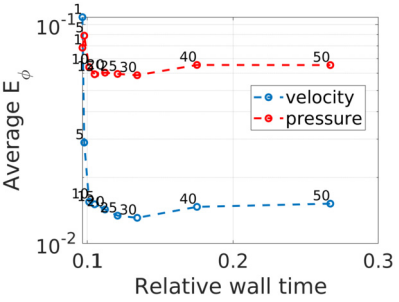


Fig. 23. 3D flow past a cylinder: Pareto plots for velocity (blue) and pressure (red) fields - averaged relative error in time versus relative wall time for varying N_{v_r}.

framework. To the best of our knowledge, it is the first time that a ROM nonlinear differential filter (i.e., a ROM spatial filter that uses an explicit lengthscale) is proposed. We assessed our ROM approach through two classical benchmarks: 2D and 3D flow past a cylinder. We found that our ROM can capture the flow features and the evolution of the aerodynamics coefficients with good accuracy when compared to the full order model. In addition, we showed the our approach is more computationally efficient in the 3D test.

Moving forward, we will test our ROM approach with more realistic applications involving flow at larger Reynolds numbers (a few thousands) like, e.g., the FDA benchmark (see [16]). Moreover, we plan to run a parametric study for the viscosity and for key model parameters, such as δ and χ. This would help us understand how to set the model parameter to obtain the most accurate results when compared to DNS.

CRediT authorship contribution statement

G. Rozza: proofreading, funding, review, literature. **A. Quaini:** drafting, revising, literature. **M. Girfoglio:** implementation and results, draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We acknowledge the support provided by the European Research Council Executive Agency by the Consolidator Grant project AROMA-CFD "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" - GA 681447, H2020-ERC CoG 2015 AROMA-CFD, PI G. Rozza, and INdAM-GNCS 2019-2020 projects. This work was also partially supported by US National Science Foundation through grant DMS-1620384 and DMS-195353.

References

[1] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer International Publishing, 2016.
[2] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, Springer International Publishing, 2016.
[3] P. Benner, M. Ohlberger, A. Patera, K. Rozza, G. Urban, Model Reduction of Parametrized Systems, 1st ed. 2017, MS&A Series, Springer, 2017.
[4] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (4) (2015) 483–531, https://doi.org/10.1137/130932715.
[5] E. Bader, M. Kärcher, M.A. Grepl, K. Veroy, Certified reduced basis methods for parametrized elliptic optimal control problems with distributed controls, SIAM J. Sci. Comput. 75 (2018) 276–307, https://doi.org/10.1137/16m1059898.
[6] P. Benner, W. Schilders, S. Grivet-Talocia, A. Quarteroni, G. Rozza, L.M. Silveira, Model Order Reduction, De Gruyter, Berlin, Boston, 2020, https://www.degruyter.com/view/title/523453.
[7] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, Comput. Methods Appl. Mech. Eng. 237–240 (2012) 10–26, https://doi.org/10.1016/j.cma.2012.04.015.
[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, https://doi.org/10.1017/s0022112088001818.
[9] 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, https://doi.org/10.1017/S0022112003005615.
[10] J.P. Boyd, Two comments on filtering (artificial viscosity) for Chebyshev and Legendre spectral and spectral element methods: preserving boundary conditions and interpretation of the filter as a diffusion, J. Comput. Phys. 143 (1) (1998) 283–288, https://doi.org/10.1006/jcph.1998.5961, http://www.sciencedirect.com/science/article/pii/S0021999198959617.
[11] P. Fischer, J. Mullen, Filter-based stabilization of spectral element methods, C. R. Acad. Sci., Ser. 1 Math. 332 (3) (2001) 265–270, https://doi.org/10.1016/S0764-4442(00)01763-8, http://www.sciencedirect.com/science/article/pii/S0764444200017638.
[12] A. Dunca, Y. Epshteyn, On the Stolz-Adams deconvolution model for the large-eddy simulation of turbulent flows, SIAM J. Math. Anal. 37 (6) (2005) 1890–1902.
[13] W. Layton, L. Rebholz, C. Trenchea, Modular nonlinear filter stabilization of methods for higher Reynolds numbers flow, J. Math. Fluid Mech. 14 (2012) 325–354.
[14] L. Bertagna, A. Quaini, A. Veneziani, Deconvolution-based nonlinear filtering for incompressible flows at moderately large Reynolds numbers, Int. J. Numer. Methods Fluids 81 (8) (2016) 463–488.
[15] A. Bowers, L. Rebholz, Numerical study of a regularization model for incompressible flow with deconvolution-based adaptive nonlinear filtering, Comput. Methods Appl. Mech. Eng. 258 (2013) 1–12.
[16] M. Girfoglio, A. Quaini, G. Rozza, A finite volume approximation of the Navier-Stokes equations with nonlinear filtering stabilization, Comput. Fluids 187 (2019) 27–45, https://doi.org/10.1016/j.compfluid.2019.05.001.
[17] M. Girfoglio, A. Quaini, G. Rozza, Fluid–structure interaction simulations with a LES filtering approach in solids4Foam, Commun. Appl. Ind. Math. 12 (2021) 13–28.
[18] X. Xie, F. Bao, C. Webster, Evolve filter stabilization reduced-order model for stochastic Burgers equation, Fluids 3 (2018) 84, https://doi.org/10.3390/fluids3040084.
[19] D. Wells, Z. Wang, X. Xie, T. Iliescu, An evolve-then-filter regularized reduced order model for convection-dominated flows, Int. J. Numer. Methods Fluids 84 (2017) 598–615.
[20] M. Gunzburger, T. Iliescu, M. Mohebujjaman, M. Schneier, An evolve-filter-relax stabilized reduced order stochastic collocation method for the time-dependent Navier–Stokes equations, SIAM/ASA J. Uncertain. Quantificat. 7 (2019) 1162–1184, https://doi.org/10.1137/18M1221618.
[21] M. Girfoglio, A. Quaini, G. Rozza, A POD-Galerkin reduced order model for a LES filtering approach, J. Comput. Phys. 436 (2021) 110260.
[22] M. Girfoglio, A. Quaini, G. Rozza, Pressure stabilization strategies for a LES filtering reduced order model, Fluids 6 (2021) 302.
[23] M. Strazzullo, F. Ballarin, M. Girfoglio, T. Iliescu, G. Rozza, Consistency of the full and reduced order models for evolve-filter-relax regularization of convection-dominated, marginally-resolved flows, Int. J. Numer. Methods Eng. 32 (2021) 3148–3178.
[24] G. Stabile, G. Rozza, Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations, Comput. Fluids 173 (2018) 273–284.
[25] 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 (3) (2009) 213–237, https://doi.org/10.1007/s00162-009-0112-y.
[26] S. Hijazi, G. Stabile, A. Mola, G. Rozza, Data-driven POD-Galerkin reduced order model for turbulent flows, J. Comput. Phys. 416 (2020) 109513, https://doi.org/10.1016/j.jcp.2020.109513.
[27] 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. Eng. 311 (2016) 151–179, https://doi.org/10.1016/j.cma.2016.08.006.
[28] M. Barrault, N.C. Nguyen, Y. Maday, A.T. Patera, An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. 339 (2004) 667–672.

[29] S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.
[30] S. Turek, M. Schäfer, Benchmark computations of laminar flow around cylinder, in: E. Hirschel (Ed.), Flow Simulation with High-Performance Computers II, in: Notes on Numerical Fluid Mechanics, vol. 52, Vieweg, 1996.
[31] V. John, Reference values for drag and lift of a two dimensional time-dependent flow around a cylinder, Int. J. Numer. Methods Fluids 44 (2004) 777–788.
[32] V. John, On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3D flow around a cylinder, Int. J. Numer. Methods Fluids 50 (2006) 845–862, https://doi.org/10.1002/fld.1080.
[33] E. Bayraktar, O. Mierka, S. Turek, Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow, Int. J. Comput. Sci. Eng. 7 (2012) 253–266, https://doi.org/10.1504/IJCSE.2012.048245.
[34] S. Pope, Turbulent Flows, Cambridge University Press, Cambridge, 2000.
[35] J. Leray, Essai sur le mouvement d'un fluide visqueux emplissant l'espace, Acta Math. 63 (1934) 193–248.
[36] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (6) (1998) 620–631.
[37] J. Borggaard, T. Iliescu, J. Roop, A bounded artificial viscosity large eddy simulation model, SIAM J. Numer. Anal. 47 (2009) 622–645.
[38] J. Hunt, A. Wray, P. Moin, Eddies stream and convergence zones in turbulent flows, Tech. Rep. CTR-S88, CTR report, 1988.
[39] A. Vreman, An eddy-viscosity subgrid-scale model for turbulent shear flow: algebraic theory and applications, Phys. Fluids 16 (10) (2004) 3670–3681.
[40] A.L. Bowers, L.G. Rebholz, A. Takhirov, C. Trenchea, Improved accuracy in regularization models of incompressible flow via adaptive nonlinear filtering, Int. J. Numer. Methods Fluids 70 (7) (2012) 805–828.
[41] M. Olshanskii, X. Xiong, A connection between filter stabilization and eddy viscosity models, Numer. Methods Partial Differ. Equ. 29 (6) (2013) 2061–2080.
[42] V. Ervin, W. Layton, M. Neda, Numerical analysis of filter based stabilization for evolution equations, SIAM J. Numer. Anal. 50 (2010) 2307–2335, https://doi.org/10.2307/41713724.
[43] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Eng. 197 (1) (2007) 173–201.
[44] P. Hariharan, M. Giarra, V. Reddy, S. Day, K. Manning, S. Deutsch, S. Stewart, M. Myers, M. Berman, G. Burgreen, E. Paterson, R. Malinauskas, Multilaboratory particle image velocimetry analysis of the FDA benchmark nozzle model to support validation of computational fluid dynamics simulations, J. Biomech. Eng. 133 (2011) 041002.
[45] S. Stewart, E. Paterson, G. Burgreen, P. Hariharan, M. Giarra, V. Reddy, S. Day, K. Manning, S. Deutsch, M. Berman, M. Myers, R. Malinauskas, Assessment of CFD performance in simulations of an idealized medical device: results of FDA's first computational inter laboratory study, Cardiovascular Engineering and Technology 3 (2) (2012) 139–160.
[46] R.I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1) (1986) 40–65.
[47] G. Stabile, G. Rozza, ITHACA-FV - in real time highly advanced computational applications for finite volumes, http://www.mathlab.sissa.it/ithaca-fv. (Accessed 30 January 2018).
[48] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (3) (2008) 229, https://doi.org/10.1007/s11831-008-9019-9.
[49] F. Chinesta, A. Huerta, G. Rozza, K. Willcox, Model order reduction, Encyclopedia of Computational Mechanics (2016).
[50] I. Kalashnikova, M.F. Barone, On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment, Int. J. Numer. Methods Eng. 83 (10) (2010) 1345–1375, https://doi.org/10.1002/nme.2867.
[51] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (4) (2011) 395, https://doi.org/10.1007/s11831-011-9064-7.
[52] A. Dumon, C. Allery, A. Ammar, Proper general decomposition (PGD) for the resolution of Navier-Stokes equations, J. Comput. Phys. 230 (4) (2011) 1387–1407, https://doi.org/10.1016/j.jcp.2010.11.010.
[53] V. Tsiolakis, M. Giacomini, R. Sevilla, C. Othmer, A. Huerta, Parametric solutions of turbulent incompressible flows in OpenFOAM via the proper generalised decomposition, J. Comput. Phys. 449 (2022) 110802.
[54] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2) (2002) 492–515, https://doi.org/10.1137/S0036142900382612.
[55] D. Lazzaro, L. Montefusco, Radial basis functions for the multivariate interpolation of large scattered data sets, J. Comput. Appl. Math. 140 (2002) 521–536, https://doi.org/10.1016/S0377-0427(01)00485-X.
[56] S.A. Orszag, M. Israeli, M. Deville, Boundary conditions for incompressible flows, J. Sci. Comput. 1 (1) (1986) 75–111, https://doi.org/10.1007/BF01061454.
[57] H. Johnston, J.-G. Liu, Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term, J. Comput. Phys. 199 (1) (2004) 221–259, https://doi.org/10.1016/j.jcp.2004.02.009.
[58] P. Lax, B. Wendroff, System of conservation laws, Commun. Pure Appl. Math. 13 (1960) 217–237.
[59] T. Passerini, A. Quaini, U. Villa, A. Veneziani, S. Canic, Validation of an open source framework for the simulation of blood flow in rigid and deformable vessels, Int. J. Numer. Methods Biomed. Eng. 29 (11) (2013) 1192–1213.