POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations

Stefano Lorenzi[a], Antonio Cammi[a], Lelio Luzzi[a,*], Gianluigi Rozza[b]

[a] *Politecnico di Milano – Department of Energy, via la Masa 34 – 20156, Milano, Italy*
[b] *SISSA, International School for Advanced Studies, Mathematics Area, mathLab, via Bonomea 265 – 34136, Trieste, Italy*

Numerical simulation of fluid flows requires important computational efforts but it is essential in engineering applications. Reduced Order Model (ROM) can be employed whenever fast simulations are required, or in general, whenever a trade-off between computational cost and solution accuracy is a preeminent issue as in process optimization and control. In this work, the efforts have been put to develop a ROM for Computational Fluid Dynamics (CFD) application based on Finite Volume approximation, starting from the results available in turbulent Reynold-Averaged Navier–Stokes simulations in order to enlarge the application field of Proper Orthogonal Decomposition-Reduced Order Model (POD-ROM) technique to more industrial fields. The approach is tested in the classic benchmark of the numerical simulation of the 2D lid-driven cavity. In particular, two simulations at Re = 10^3 and Re = 10^5 have been considered in order to assess both a laminar and a turbulent case. Some quantities have been compared with the Full Order Model in order to assess the performance of the proposed ROM procedure i.e., the kinetic energy of the system and the reconstructed quantities of interest (velocity, pressure and turbulent viscosity). In addition, for the laminar case, the comparison between the ROM steady-state solution and the data available in literature has been presented. The results have turned out to be very satisfactory both for the accuracy and the computational times. As a major outcome, the approach turns out not to be affected by the energy blow up issue characterizing the results obtained by classic turbulent POD-Galerkin methods.

Keywords: Proper orthogonal decomposition; Parametrized Navier–Stokes Equation; Reduced order modelling; RANS; Galerkin projection

Highlights

- Reduced order modelling (ROM) for numerical simulation of fluid flows.
- Proper Orthogonal Decomposition (POD) with snapshots technique and Galerkin projection.
- Development of a ROM for CFD applications based on Finite Volume approximation.
- Employment of turbulent RANS simulations to extend the method to more industrial fields.
- Benchmark of the 2D lid-driven cavity both in laminar and turbulent conditions to assess the method.

Received 15 February 2016;
received in revised form 2 August 2016;
accepted 5 August 2016
Available online 11 August 2016

[*] Corresponding author.
E-mail addresses: stefano.lorenzi@polimi.it (S. Lorenzi), antonio.cammi@polimi.it (A. Cammi), lelio.luzzi@polimi.it (L. Luzzi), gianluigi.rozza@sissa.it (G. Rozza).

Published Journal Article available at: http://dx.doi.org/10.1016/j.cma.2016.08.006

Nomenclature

a_i	time coefficient,
$\boldsymbol{a}$	ROM time coefficient vector,
$\boldsymbol{B}$	ROM matrix,
$\boldsymbol{BT}$	ROM matrix,
$\boldsymbol{C}$	ROM matrix,
$\boldsymbol{CT1}$	ROM matrix,
$\boldsymbol{CT2}$	ROM matrix,
$\boldsymbol{D}$	ROM matrix,
e	L^2 error of velocity,
$\boldsymbol{E}$	ROM matrix,
F	face flux, $\mathrm{m}^3\ \mathrm{s}^{-1}$
F_r	ROM face flux, $\mathrm{m}^3\ \mathrm{s}^{-1}$
k	turbulent kinetic energy, $\mathrm{m}^2\ \mathrm{s}^{-2}$
$\boldsymbol{n}$	normal vector,
N_r	number of ROM functions,
N_s	number of snapshots,
p	normalized pressure, $\mathrm{m}^2\ \mathrm{s}^{-2}$
p_r	ROM normalized pressure, $\mathrm{m}^2\ \mathrm{s}^{-2}$
t	time, s
$\boldsymbol{u}$	velocity, $\mathrm{m}\ \mathrm{s}^{-1}$
$\boldsymbol{u}_n$	snapshots velocity, $\mathrm{m}\ \mathrm{s}^{-1}$
$\boldsymbol{u}_r$	ROM velocity, $\mathrm{m}\ \mathrm{s}^{-1}$
$\boldsymbol{u}_t$	derivative of the velocity, $\mathrm{m}\ \mathrm{s}^{-1}$
$\boldsymbol{u}_{BC}$	Dirichlet boundary condition of velocity, $\mathrm{m}\ \mathrm{s}^{-1}$
$\boldsymbol{u}_{FOM}$	FOM velocity, $\mathrm{m}\ \mathrm{s}^{-1}$
$\boldsymbol{x}$	vector of spatial coordinate, m
V	volume, m^3

Greek symbols

$\varGamma$	boundary function,
υ	kinematic viscosity, $\mathrm{m}^2\ \mathrm{s}^{-1}$
υ_{EV}	eddy viscosity in POD-G-ROM, $\mathrm{m}^2\ \mathrm{s}^{-1}$
υ_t	turbulent viscosity, $\mathrm{m}^2\ \mathrm{s}^{-1}$
$\upsilon_{t,r}$	ROM turbulent viscosity, $\mathrm{m}^2\ \mathrm{s}^{-1}$
τ	penalty factor,
$\boldsymbol{\varphi}_i$	velocity spatial modes, $\mathrm{m}\ \mathrm{s}^{-1}$
ϕ_i	turbulent viscosity spatial modes, $\mathrm{m}^2\ \mathrm{s}^{-2}$
χ_i	pressure spatial modes, $\mathrm{m}^2\ \mathrm{s}^{-2}$
ψ_i	face flux spatial modes, $\mathrm{m}^3\ \mathrm{s}^{-1}$
ω	specific dissipation, s^{-1}
$\varOmega$	spatial domain, m^3

1. Introduction

Numerical simulation of fluid flows requires a strong computational effort but it is essential in engineering applications. Even if the computational power is becoming more and more available, the need of finding a trade-off between computational cost and solution accuracy is a preeminent issue especially in process optimization, control or in general in any real time or many query contexts [1–5]. A viable solution is to employ Reduce Order Modelling (ROM) techniques [6]. The aim of a computational reduction technique is to retain the governing dynamics of a system in rapid and reliable way [7]. The main assumption of ROM is that the behaviour of the system with respect to a parameter (physical, geometrical) or the time can be represented by a small number of dominant modes. The reduced order modelling is aimed at approximating a Partial Differential Equation (PDE) solution (or a set of Ordinary Differential Equations (ODEs)) with a reduced number of degrees of freedom [7]. ROM can be employed for instance as the basis for the synthesis and the verification of controllers [8–10] or in some optimization algorithm [11–13]. A suitable offline/online procedure is usually employed in order to efficiently decouple the generation of the basis (made only once — offline phase), which involves the resolution of a Full Order Model (FOM), and the simulation of the ROM that can be run many times as required (online phase). Among the several reduced order techniques, the Proper Orthogonal Decomposition (POD) with the snapshot technique [14,15] is probably the most widespread in the complex fluid flow computation. POD technique was first introduced to study the coherent structures in experimental turbulent flows [16–18] but it has recently become a valuable option in the ROM framework [19–22] due to the capability to select the most energetic modes representing the most significant features of the system. Starting from spatio-temporal solutions of the system obtained from numerical simulations or experimental measurements (i.e., the so called "snapshots"), a POD basis maximizing the energy content in the starting ensemble data is created. By applying the Galerkin projection of the original system into the space spanned by the POD basis, a low-order model can be obtained (POD-G-ROM) [22,23].

Even if in principle the several ROM techniques can be applied to different approximation schemes (i.e., finite difference, Finite Volume (FV), Finite Elements (FE), spectral methods), the most widespread method is the FE method. This entails that the several reduced-order techniques are usually optimized for this discretization method. The interest to study the FV approximation relies on the fact that from the industrial point of view, the latter is considered (a) robust, (b) computationally inexpensive, and (c) suitable when the conservation of the numerical flux is a relevant issue, as in the fluid-dynamics application [24]. Even if the FE can be more accurate, the FV is usually chosen for industrial applications since it is easily applicable to realistic and physical context, it does not require any particular functional framework as FE and it preserves locally the conservation laws [25,26]. Nowadays, the most widespread computational codes used in the industry, both commercial (CFX, FLUENT, STARCD) [27] and open-source (OpenFOAM) [28], are based on FV discretization. In literature, the general application of ROM technique using FV approximation has been investigated in the work of Haasdonk and Ohlberger [29–33] extending the Reduced Basis (RB) approach to general linear evolution schemes such as FV schemes. Notwithstanding, there is a lack of literature regarding POD-G-ROM with FV approximation of the Navier–Stokes Equations (NSEs). At the best of the authors knowledge, only Östh et al. [34] used a commercial FV code to solve the Large Eddy Simulation (LES) equations in the offline procedure without specifying if a special procedure was used. On the other hand, a flux matching procedure (instead of Galerkin projection) of the Reynold-Averaged Navier–Stokes (RANS) equations was proposed by Rambo et al. [35–37].

Another remarkable issue is that the fluid flow usually considered in the engineering field is turbulent. The eddies created by the turbulence span a large range of length and time scales. Accordingly, they can be modelled with several degree of accuracy according to the resolution needed for the engineering application [25]. Even if LES and Direct Numerical Simulation (DNS) are very accurate (but computationally expensive), for most engineering applications the RANS equations are sufficient to describe the main time-averaged properties of the flow (velocity, pressures, and stresses). To this end, suitable turbulence models are taken into account such as the eddy viscosity models (e.g., Spalart–Allmaras, k-ε, k-ω) [25,38]. A lot of works are present in the literature dealing with POD-ROM models for High Reynolds number starting from snapshots collected by LES and DNS simulations [16,22,23,34,39–45]. As stated in these works, the main problem is the occurrence of unstable time behaviour in the reduced order model. This can be explained, as it was confirmed by Couplet et al. [46], considering the concept of the energy cascade, firstly proposed by Richardson [47] and then confirmed by Kolmogorov's theory [48]. A turbulent flow is composed by different size eddies. The large eddies carry most of the energy, extracting the energy from the mean flow. These eddies are unstable and break up into smaller eddies. In this way, the energy is transferred from the large scale to the smaller one. At a certain point, the eddies reach a sufficiently small length and energy scale that the viscosity can dissipate the kinetic energy into internal energy. Since the POD basis is built considering a maximum energy criterion, the POD truncation error leaves out the higher order modes, which are the less energetic but the most dissipative. Accordingly, the truncated POD-ROM model can blow up. In literature, this issue has been fixed introducing proper closure modelling. Among the possible solutions, the introduction of a fictitious eddy viscosity, modelling a stabilizing dissipative term, was considered in several ways [16,39].

From the standpoint of the aforementioned state-of-the-art and the research issues, in this work, the efforts have been put to develop a ROM for Computational Fluid Dynamics (CFD) application based on FV approximation (POD-FV-ROM), starting from the results available in turbulent RANS simulations in order to enlarge the application field of POD-ROM technique to wider industrial fields. In this light, it is important to adopt the common tools employed for the resolution of NSEs in industrial fields and to modify as less as possible the turbulent models usually employed. To this end, for the offline calculation, i.e., the resolution of the Full Order Model and the basis calculation, the open-source OpenFOAM software is adopted [28,49] as a reliable, tested and flexible tool for CFD application. As a major outcome, the POD-FV-ROM is implemented just introducing some post-processing utilities, avoiding the need to modify the already tested solvers.

The manuscript is devoted to present the new ROM methodology and the several numerical procedures along with a classic numerical test [50–52] aimed at comparing the ROM and the corresponding FOM in the same simulation cases used to create the snapshots. Additional analyses on the proposed POD-FV-ROM methodology is the subject of a parallel work of the authors [53]. In the first part of the present paper, the modifications needed to the classic POD-G-ROM to be adopted in a FV framework are highlighted. In the second one, the approach adopted for the turbulence, starting from RANS simulations, is presented. The adoption of RANS with eddy viscosity models (necessary for the creation of the snapshots) allows avoiding the incorporation of the fictitious eddy viscosity since this quantity is already modelled in the Full Order Model (FOM). The POD-FV-ROM is tested in the classic benchmark of the numerical simulation of the 2D lid-driven cavity [50–52]. In particular, two simulations at $\mathrm{Re} = 10^3$ and $\mathrm{Re} = 10^5$ are considered in order to assess both a laminar and a turbulent case.

The paper is organized as follows. In Section 2, the classic POD-G-ROM and the POD-FV-ROM are illustrated and compared. The RANS turbulence approach is then explained in Section 3. In Section 4, the two benchmark cases are reported. Finally, some conclusions, perspectives and further developments are provided in Section 5.

2. POD-Galerkin projection with Finite Volume

In this section, the attention is paid to define a procedure for obtaining a POD-ROM of the Navier–Stokes equations considering the Finite Volume approximation (POD-FV-ROM). For the moment, we consider the incompressible Navier–Stokes equations without any turbulence treatment (this aspect will be introduced in Section 3). In equations, it reads:

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

where **u** is the velocity, p is a normalized pressure[1] and υ is the kinematic viscosity. The equations are given in a domain Ω with proper boundary and initial conditions.

2.1. The classic POD-Galerkin reduced order modelling

In literature, there are several papers about POD-Galerkin ROM for Navier–Stokes equations. In this subsection, the classic POD-G-ROM methodology is briefly summarized according to [11,22,23,54]. For more details about the POD theory, the reader may refer to [14,15].

The main assumption in the reduced order techniques based on projection method is that the approximated solution of the problem $\boldsymbol{u}_r\,(\boldsymbol{x}, t)$ can be expressed as linear combination of spatial modes $\boldsymbol{\varphi}_i\,(\boldsymbol{x})$ multiplied by temporal coefficients $a_i(t)$. If we consider the velocity, this assumption reads[2]

$$\boldsymbol{u}\,(\boldsymbol{x}, t) \approx \boldsymbol{u}_r\,(\boldsymbol{x}, t) = \sum_{i=1}^{N_r} a_i(t)\boldsymbol{\varphi}_i\,(\boldsymbol{x})\,. \tag{2}$$

[1] p is the thermodynamic pressure divided by the fluid density.

[2] Generally, the velocity field is decomposed in a mean time-independent flow and a linear combination of time-dependent fluctuations. In this paper, we consider a general formulation with no base flow.

The selection of the spatial modes is one of the crucial point in the reduced order modelling. A correct choice of these functions leads to an efficient ROM, reducing the online simulation time or increasing the accuracy with respect to the Full Order Model. Among the different options (e.g., eigenfunctions, POD modes, Dynamic Mode Decomposition (DMD) modes [55]), the POD is usually selected in fluid flow computation due to the capability to select the most energetic modes. The POD basis

$$X_{N_r}^{POD} := span\left\{\boldsymbol{\varphi}_i\right\} \quad i = 1, \ldots, N_r \tag{3}$$

can be built starting from a set of velocity solutions sampled at different and evenly spaced times (i.e., the snapshots)

$$\boldsymbol{u}_n\left(\boldsymbol{x}\right) := \boldsymbol{u}\left(\boldsymbol{x}, t_n\right) \quad n = 1, \ldots,\ N_s. \tag{4}$$

The snapshot can be numerical solutions of the NSEs (typical from LES and DNS simulations or even by the RANS equations as considered in this work) or they are obtained from experimental results. The POD basis minimizes the difference between the snapshots and the projection of the snapshots on the spatial modes in the X-norm, given the orthonormality of the modes. If the L^2-norm is chosen, the POD basis is optimal considering the energy contained in the snapshots.

$$\begin{gathered} X_{N_r}^{POD} = \arg\min \frac{1}{N_s} \sum_{n=1}^{N_s} \left\| \boldsymbol{u}_n\left(\boldsymbol{x}\right) - \sum_{i=1}^{N_r} \left\langle \boldsymbol{u}_n\left(\boldsymbol{x}\right), \boldsymbol{\varphi}_i\left(\boldsymbol{x}\right)\right\rangle_{L^2} \boldsymbol{\varphi}_i\left(\boldsymbol{x}\right) \right\|_{L^2}^2, \\ \left\langle \boldsymbol{\varphi}_i\left(\boldsymbol{x}\right), \boldsymbol{\varphi}_j\left(\boldsymbol{x}\right)\right\rangle_{L^2} = \delta_{ij}. \end{gathered} \tag{5}$$

In order to solve Eq. (5), the following eigenvalue problem is considered

$$C\xi_i = \lambda_i \xi_i \quad i = 1, \ldots, N_s \tag{6}$$

where $C \in \mathbb{R}^{N_s\, x\, N_s}$ is the correlation matrix whose components are calculated as follows

$$[C]_{kl} = \frac{1}{N_r} \left\langle \boldsymbol{u}_k\left(\boldsymbol{x}\right), \boldsymbol{u}_l\left(\boldsymbol{x}\right)\right\rangle_{L^2}. \tag{7}$$

The (λ_i, ξ_i) eigenvalue–eigenvector pair is used to construct the functions of the POD basis

$$\boldsymbol{\varphi}_i\left(\boldsymbol{x}\right) = \frac{1}{\sqrt{\lambda_i}} \sum_{n=1}^{N_s} \xi_{i,n} \boldsymbol{u}_n\left(\boldsymbol{x}\right) \quad i = 1, \ldots, N_r. \tag{8}$$

It is worthwhile to remind that, since the eigenvalues are sorted in descending order, the first modes have the property to retain most of the energy present in the original solutions [17]. This is an important feature when considering the turbulence effects. In addition, the functions are orthogonal and they can be suitably normalized in order to obtain $\left\langle \boldsymbol{\varphi}_i\left(\boldsymbol{x}\right), \boldsymbol{\varphi}_j\left(\boldsymbol{x}\right)\right\rangle_{L^2} = \delta_{ij}$.

Replacing the velocity $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in Eq. (1) and applying a Galerkin projection of the resulted system on the functions of the POD basis, the following POD-Galerkin ROM (POD-G-ROM) is obtained

$$\frac{da_j(t)}{dt} = \upsilon \sum_{i=1}^{N_r} B_{ji} a_i\left(t\right) - \sum_{k=1}^{N_r} \sum_{i=1}^{N_r} C_{jki} a_k\left(t\right) a_i\left(t\right) \quad j = 1, \ldots, N_r \tag{9}$$

where

$$B_{ji} = \left\langle \nabla \boldsymbol{\varphi}_j, \nabla \boldsymbol{\varphi}_i \right\rangle_{L^2} \tag{10}$$

$$C_{jki} = \left\langle \boldsymbol{\varphi}_j, \left(\boldsymbol{\varphi}_k \cdot \nabla\right) \boldsymbol{\varphi}_i \right\rangle_{L^2} \tag{11}$$

$$a_j(0) = \left\langle \boldsymbol{\varphi}_j, \boldsymbol{u}_1\left(\boldsymbol{x}\right)\right\rangle_{L^2}. \tag{12}$$

Eq. (9) can be expressed as the following autonomous dynamical system in which the unknowns are the time-dependent coefficients $a_i(t)$:

$$\dot{\boldsymbol{a}} = \upsilon \boldsymbol{B}\boldsymbol{a} - \boldsymbol{a}^{\boldsymbol{T}} \boldsymbol{C}\boldsymbol{a}. \tag{13}$$

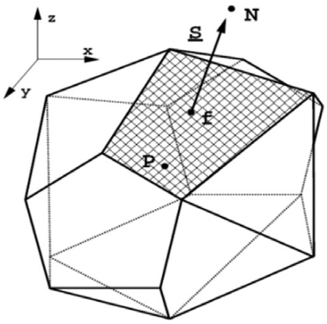


Fig. 1. Example of a control volume [57].

In deriving the POD-G-ROM, two relevant assumptions are made:

(a) The first one is the lack of a pressure term in Eq. (9). This can be explained considering that the POD modes are linear combination of the snapshots, which are divergence-free since they satisfy the continuity equation. In this way, the POD modes preserve the divergence-free property (in discrete sense). The Galerkin projection of the pressure term reads:

$$\left\langle \boldsymbol{\varphi}_i, \nabla p \right\rangle_{L^2} = \int_{\Omega} \boldsymbol{\varphi}_i \cdot \nabla p \, d\boldsymbol{x} = -\int_{\Omega} p \cdot \left(\nabla \cdot \boldsymbol{\varphi}_i\right) \, d\boldsymbol{x} + \int_{\partial\Omega} p \cdot \left(\boldsymbol{\varphi}_i \cdot \boldsymbol{n}\right) \, d\boldsymbol{x} \tag{14}$$

where the first term is null and the second one is also zero in case of enclosed flows [11]. In general, the pressure term can be neglected if the computational domain is large enough [22], or proper boundary conditions are satisfied [23]. On the other hand, in some cases it is not possible to eliminate this term, hence some additional terms [56] or the construction of a basis for the pressure are needed [44].

(b) The second point is that the term B_{ji} (Eq. (9)) representing the diffusive term is derived keeping into account that $\Delta u = \text{div}(\nabla u)$ and applying the Green formula for the divergence operator. This procedure is typical of the weak formulation of differential problems in the FE approach and it is extended also to POD-G-ROM [22].

These two observations are relevant in the following application of POD-G-ROM to the FV discretization.

2.2. The Finite Volume method: issues for the reduced order modelling

The Finite Volume method is a discretization method based on a "balance" approach, well suited for the solution of equations based on conservation laws. A local balance, obtained from the discretization of the integral form of the governing equations, is written on each discretization cell (i.e., control volume — see Fig. 1).

This method is quite attractive for CFD since the quantities of interest (mass flow, momentum) are conserved at the discrete level [57]. Considering the momentum balance of Eq. (1), the integral form on a generic control volume V_P reads

$$\int_{t}^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_{V_P} \boldsymbol{u}_t dV + \int_{V_P} (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} dV - \upsilon \int_{V_P} \Delta \boldsymbol{u} dV + \int_{V_P} \nabla p dV \right] dt = 0. \tag{15}$$

In general, the variation of the velocity $\mathbf{u}$ around the centroid of the control volume (denoted as P in Fig. 1) is taken as linear in order to have a second-order accurate method. In the following, this assumption is adopted.

In order to obtain a POD-FV-ROM, it is important to remind three relevant numeric issues characterizing the FV discretization of the Navier–Stokes equations:

(a) the convective (non-linear) term, given the linear variation of $\mathbf{u}$ and applying the generalized form of the Gauss' theorem, is discretized as [57]

$$\int_{V_P} (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} = \int_{V_P} \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) \approx \sum_f \boldsymbol{S}_f \cdot \boldsymbol{u}_f \boldsymbol{u}_f = \sum_f F \boldsymbol{u}_f \tag{16}$$

where S_f is the face area vector of the face (see Fig. 1) and F is the face flux. This entails that the face flux field should be considered in the POD-FV-ROM procedure in order to be consistent with the full order modelling.

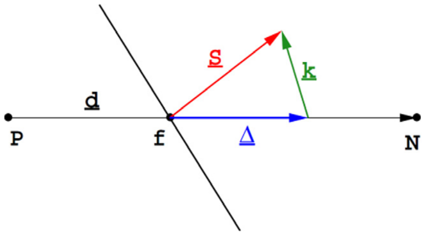


Fig. 2. Non-orthogonality treatment of the diffusive term [57].

(b) The continuity equation is discretized as follows

$$\int_{V_P} \nabla \cdot \boldsymbol{u} = \sum_f \boldsymbol{S}_f \cdot \boldsymbol{u}_f = \sum_f F = 0 \tag{17}$$

i.e., the divergence-free constraint is applied not to the cell centre value, but to the face flux. In this way, it is not possible anymore to neglect the pressure term, as seen for the POD-G-ROM, since the snapshots, calculated in the centre cell value, are not divergence-free.[3]

(c) The diffusive term is discretized as

$$\upsilon \int_{V_P} \Delta \boldsymbol{u} = \upsilon \sum_f \boldsymbol{S}_f \cdot \nabla \boldsymbol{u}_f = \upsilon \sum_f |\Delta| \frac{\boldsymbol{u}_N - \boldsymbol{u}_P}{|\boldsymbol{d}|} + \boldsymbol{k} \cdot (\nabla \boldsymbol{u})_f \tag{18}$$

where the first term is the orthogonal contribution and the second one the non-orthogonal correction (Fig. 2). A second possible option for the discretization of the diffusive term can be undertaken. If we considered $\Delta u = \mathrm{div}(\nabla u)$ and if we applied the Green formula for the divergence operator, the following discretization would be obtained

$$\begin{aligned} \upsilon \int_{V_P} \nabla \cdot \nabla \boldsymbol{u} &= \upsilon \sum_f \boldsymbol{S}_f \cdot \nabla \boldsymbol{u}_f \\ &= \upsilon \sum_f \boldsymbol{S}_f \cdot \left[f_x \left(\frac{1}{V} \sum_f \boldsymbol{S}_f \boldsymbol{u}_f \right)_P + (1 - f_x) \left(\frac{1}{V} \sum_f \boldsymbol{S}_f \boldsymbol{u}_f \right)_N \right]. \end{aligned} \tag{19}$$

It is clear that the Green formula leads to a different discretization of the diffusive term. Eq. (18) is usually preferred to Eq. (19) since, although both are second-order accurate, the second one involves a larger computational molecule and the first term of the truncation error is four times larger than the first one [57]. Accordingly, the Green formula cannot be exploited in the POD-FV-ROM procedure since it would introduce discretization discrepancies between the Full Order Model and the Reduced Order one. As additional consequence, the boundary conditions cannot be explicitly incorporated in the ROM as in the POD-G-ROM case.

Finally, in the application of POD-G-ROM to the FV discretization, the three aforementioned issues typical of FV discretization should be carefully handled, considering the assumptions of the POD-G-ROM pointed out in Section 2.1. In the next section, the modifications needed in the track of the previous considerations are pointed out.

2.3. The POD-Galerkin reduced order modelling for Finite Volume discretization

A POD-Galerkin Reduced Order Modelling for Finite Volume discretization should take into account the remarks outlined in Section 2.2, i.e., the need of calculating the face flux, the incorporation of the pressure term and the problem of considering the boundary conditions since the Green formula cannot be used for the discretization of the

[3] Actually, the snapshots are "almost" divergence-free due to the relation between the centre and the flux value. Notwithstanding, it is better not to consider the snapshot fields as solenoidal to avoid introducing error in the ROM due to the discretization process.

diffusive term. The procedure is intended to be the less "invasive" as possible to avoid modifying the full order model, i.e., the code and solver usually adopted in industrial problems.

The first two issues can be solved expanding the face flux and the pressure as linear combination of the first N_r spatial modes:

$$F\left(\boldsymbol{x},t\right) \approx F_r\left(\boldsymbol{x},t\right) = \sum_{i=1}^{N_r} a_i(t)\psi_i\left(\boldsymbol{x}\right) \tag{20}$$

$$p\left(\boldsymbol{x},t\right) \approx p_r\left(\boldsymbol{x},t\right) = \sum_{i=1}^{N_r} a_i(t)\chi_i\left(\boldsymbol{x}\right) \tag{21}$$

where $\psi_i\left(\boldsymbol{x}\right)$ and $\chi_i\left(\boldsymbol{x}\right)$ are the functions of the spatial basis for the face flux and the pressure, respectively. These spatial bases are constructed considering the eigenvectors of the correlation matrix of the velocity (Eq. (7)) and the snapshots of the face flux/pressure ($F_n\left(\boldsymbol{x}\right)$ and $p_n\left(\boldsymbol{x}\right)$, obtained from the full order model) as follows:

$$\psi_i\left(\boldsymbol{x}\right) = \frac{1}{\sqrt{\lambda_i}}\sum_{n=1}^{N_s}\xi_{i,n}F_n\left(\boldsymbol{x}\right) \quad i = 1, \dots, N_r, \tag{22}$$

$$F_n\left(\boldsymbol{x}\right) := F\left(\boldsymbol{x},t_n\right) \quad n = 1, \dots,\ N_s, \tag{23}$$

$$\chi_i\left(\boldsymbol{x}\right) = \frac{1}{\sqrt{\lambda_i}}\sum_{n=1}^{N_s}\xi_{i,n}p_n\left(\boldsymbol{x}\right) \quad i = 1, \dots, N_r, \tag{24}$$

$$p_n\left(\boldsymbol{x}\right) := p\left(\boldsymbol{x},t_n\right) \quad n = 1, \dots,\ N_s. \tag{25}$$

Please note that the time coefficients for the approximated velocity, face flux and pressure are the same therefore only the momentum equation is needed to solve them. Limited to the pressure, this approach was used also in [44]. On the other hand, in this work the weights employed to build the spatial basis are calculated from the eigenvectors of the correlation matrix of the velocity, and not a combination of velocity and pressure, as in the previous mentioned reference. In this way, the physical meaning of retaining the most energetic modes in the basis is conserved.

This approach can be interpreted as if the state vector of the variables of interest was expanded as linear combination of state vector spatial modes:

$$\begin{pmatrix}\boldsymbol{u}\left(\boldsymbol{x},t\right)\\ F\left(\boldsymbol{x},t\right)\\ p\left(\boldsymbol{x},t\right)\end{pmatrix} \approx \begin{pmatrix}\boldsymbol{u_r}\left(\boldsymbol{x},t\right)\\ F_r\left(\boldsymbol{x},t\right)\\ p_r\left(\boldsymbol{x},t\right)\end{pmatrix} = \sum_{i=1}^{N_r} a_i(t)\begin{pmatrix}\boldsymbol{\varphi}_i\left(\boldsymbol{x}\right)\\ \psi_i\left(\boldsymbol{x}\right)\\ \chi_i\left(\boldsymbol{x}\right)\end{pmatrix}. \tag{26}$$

Replacing the velocity $\boldsymbol{u}$ with $\boldsymbol{u}_r$ and p with p_r in Eq. (1), employing the approximated face flux F_r in the convective term, and applying the Galerkin projection, the following POD-Galerkin ROM for Finite Volume discretization (POD-FV-ROM) is obtained

$$\frac{da_j(t)}{dt} = \upsilon\sum_{i=1}^{N_r} B_{ji}a_i\left(t\right) - \sum_{k=1}^{N_r}\sum_{i=1}^{N_r} C_{jki}a_k\left(t\right)a_i\left(t\right) - \sum_{i=1}^{N_r} A_{ji}a_i\left(t\right) \tag{27}$$

where

$$B_{ji} = \left\langle \boldsymbol{\varphi}_j, \Delta\boldsymbol{\varphi}_i\right\rangle_{L^2}, \tag{28}$$

$$C_{jki} = \left\langle \boldsymbol{\varphi}_j, \nabla\cdot\left(\psi_k, \boldsymbol{\varphi}_i\right)\right\rangle_{L^2}, \tag{29}$$

$$A_{ji} = \left\langle \boldsymbol{\varphi}_j, \nabla\chi_i\right\rangle_{L^2}. \tag{30}$$

The dynamical system of the time-dependent coefficient can be expressed as

$$\dot{\boldsymbol{a}} = \upsilon\boldsymbol{B}\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} - \boldsymbol{A}\boldsymbol{a}. \tag{31}$$

Please note that for the term B_{ji} the Green's formula has not been applied due to the inferences mentioned in Section 2.2. Therefore, the Boundary Conditions (BCs) are "embedded" in the B_{ji} term and not explicit present in the ROM formulation. This may be of concern when dealing with parametrized ROM, e.g., in control-oriented applications. In particular, in the fluid dynamics field, the classic control variable is the velocity at the boundary [58] since it could be used to control the velocity field in the domain or an output variable of interest. In this view, if the reduced order model is directed to the synthesis of controllers, it should have the possibility to vary the velocity in order to test the several control action. Even if in this work the variation of the BCs is not considered in the numerical simulations, the possibility to parametrize the velocity at the BC in the reduced order model is taken into account. To this aim, a POD penalty method enforcing the BCs is considered [59]. As in the spectral methods [60], the Dirichlet BCs are directly incorporated in the Galerkin projection of the NSE as constraints

$$\left\langle \boldsymbol{\varphi}_j, \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} - \upsilon\Delta\boldsymbol{u} + \nabla p + \tau\Gamma\,(\boldsymbol{u} - \boldsymbol{u_{BC}})\right\rangle_{L^2} = 0 \tag{32}$$

where $\mathbf{u}_{BC}$ is the Dirichlet boundary condition, τ the penalty factor and Γ is a null function except on the boundary where the condition is imposed [59]. The POD-penalty method allows incorporating and handling Dirichlet boundary conditions and it has two additional significant advantages. The first one is that this procedure enforces the approximated velocity $\mathbf{u}_r$ to respect the BC of the problem. This should not be taken for granted since the approximated velocity is a linear combination of spatial functions, which in general do not respect the Dirichlet BC[4], except in the case of homogeneous one. The second advantage in using the POD penalty method lays in the fact that now the model is not autonomous anymore. In this way, some wrong long-time integration behaviour and the initial condition issue are less troublesome [59]. As for the penalty factor, this number is usually tuned with a sensitivity analysis [57,61]. In general if τ tends to zero, the BCs are not enforced. On the other hand, if τ tends to infinity the ROM becomes ill-conditioned [59].

The POD-FV-ROM system is then modified accordingly:

$$\frac{da_i(t)}{dt} = \upsilon\sum_{i=1}^{N_r} B_{ji}a_i\,(t) - \sum_{k=1}^{N_r}\sum_{i=1}^{N_r} C_{jki}a_k\,(t)\,a_i\,(t) - \sum_{i=1}^{N_r} A_{ji}a_i\,(t) + \tau\left(\boldsymbol{u_{BC}}\cdot\boldsymbol{D}_j - \sum_{i=1}^{N_r} E_{ji}a_i\,(t)\right) \tag{33}$$

where the additional terms with respect to Eq. (27) are projected on the boundary considered as follows;

$$\boldsymbol{D}_j = \left\langle \boldsymbol{\varphi}_j \right\rangle_{L^2,\,\partial\Omega}, \tag{34}$$

$$E_{ji} = \left\langle \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i \right\rangle_{L^2,\,\partial\Omega}. \tag{35}$$

The POD-FV-ROM dynamical system for incompressible laminar NSE reads as

$$\dot{\boldsymbol{a}} = \upsilon\boldsymbol{Ba} - \boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{Ca} - \boldsymbol{Aa} + \tau\left(\boldsymbol{u_{BC}D} - \boldsymbol{Ea}\right). \tag{36}$$

3. RANS turbulent POD-FV-ROM

The POD-FV-ROM presented in Section 2 is suitable for laminar flows since no turbulence treatment is considered in the Full Order Model (Eq. (1)). Notwithstanding, the majority of the industrial applications deals with turbulent flows and the ROM should handle this aspect in order to become a powerful tool. To this end, the first choice to be made concerns the degree of detail in the modelling of turbulent flow. Even if in literature LES or DNS simulations are usually adopted for the FOM (i.e., for the snapshots as in [16,22,23,34,39–45]), in this work the RANS approach is considered as in [35–37]. This choice is due to the fact that in the modelling of industrial turbulent flows, this approach is preferred to the DNS or LES approaches since the latter are computationally expensive for complex geometry [25]. Employing the RANS in the FOM and for the snapshot generation allows preserving the typical industrial modelling approach in the application of the reduced order modelling, as we will see in the following. Contrary to the work of [35–37], where a flux matching procedure is proposed, in this work the Galerkin projection is used. In this sense, this work constitutes the first attempt in literature to use the POD-Galerkin approach to handle turbulence issues starting from RANS simulations.

[4] The functions of the basis do not respect the Dirichlet BC since they are linear combination of snapshots (Eq. (8)) which respect the BC in turn.

The proposed procedure for incorporating the turbulence treatment in the POD-FV-ROM is different from the standard literature approach [16,39]. In the latter case, the POD-G-ROM is usually modified introducing a fictitious eddy viscosity, in order to prevent the occurrence of the blow-up of the system. This undesired behaviour can be ascribed to the discard of the high order modes which contribute to the energy dissipation. The POD-G-ROM of Eq. (9) is usually modified as follows:

$$\frac{da_j(t)}{dt} = \upsilon \sum_{i=1}^{N_r} B_{ji} a_i(t) - \sum_{i=1}^{N_r} H_{ji} a_i(t) - \sum_{k=1}^{N_r}\sum_{i=1}^{N_r} C_{jki} a_k(t) a_i(t) - \sum_{i=1}^{N_r} A_{ji} a_i(t) \tag{37}$$

where

$$H_{ji} = \left\langle \nabla \boldsymbol{\varphi}_j, \upsilon_{EV} \nabla \boldsymbol{\varphi}_i \right\rangle_{L^2}. \tag{38}$$

There are several closure models for eddy viscosity υ_{EV} available in literature, the reader may refer to [39] for a general review. It should be noted that, differently from the literature cases that employ LES and DNS simulations for the snapshots creation, a term representing an eddy (turbulent) viscosity already exists if the RANS is adopted as FOM [25]. In particular, if we consider a general RANS eddy viscosity model, the equations of the FOM read:

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla \cdot \left[-p\boldsymbol{I} + (\upsilon + \upsilon_t)\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\boldsymbol{T}}\right) - 2/3k\boldsymbol{I}\right] \\ \nabla \cdot \boldsymbol{u} = 0. \end{cases} \tag{39}$$

The turbulent viscosity υ_t is usually function of one ($\widetilde{\upsilon}$ in Spalart–Allmaras) or two variables (k and ε/ω in the respective models) [25]. The equations of these quantities are usually complicated, depending on several parameters and functions. In order to avoid the implementation of these complex relationships in the ROM, only the eddy viscosity is taken into account. The idea is once again to expand the eddy viscosity of Eq. (39) as linear combination of spatial modes

$$\upsilon_t(\boldsymbol{x}, t) \approx \upsilon_{t,r}(\boldsymbol{x}, t) = \sum_{i=1}^{N_r} a_i(t)\phi_i(\boldsymbol{x}) \tag{40}$$

where $\phi_i(\boldsymbol{x})$ are the functions of the spatial basis for the eddy viscosity. As for the face flux and pressure, the spatial basis can be built starting from the snapshots of the viscosity ($\upsilon_{t,n}(\boldsymbol{x})$) and the eigenvectors of the correlation matrix of the velocity (Eq. (7)).

$$\phi_i(\boldsymbol{x}) = \frac{1}{\sqrt{\lambda_i}} \sum_{n=1}^{N_s} \xi_{i,n} \upsilon_{t,n}(\boldsymbol{x}) \quad i = 1, \ldots, N_r \tag{41}$$

$$\upsilon_{t,n}(\boldsymbol{x}) := \upsilon_t(\boldsymbol{x}, t_n) \quad n = 1, \ldots, N_s. \tag{42}$$

Also in this case, the time-coefficients are the same for velocity, face flux, pressure and turbulent viscosity, and just the momentum equation is needed to solve them. By adding the viscosity, the state vector of Eq. (26) can be updated as:

$$\begin{pmatrix} \boldsymbol{u}(\boldsymbol{x}, t) \\ F(\boldsymbol{x}, t) \\ p(\boldsymbol{x}, t) \\ \upsilon_t(\boldsymbol{x}, t) \end{pmatrix} \approx \begin{pmatrix} \boldsymbol{u_r}(\boldsymbol{x}, t) \\ F_r(\boldsymbol{x}, t) \\ p_r(\boldsymbol{x}, t) \\ \upsilon_{t,r}(\boldsymbol{x}, t) \end{pmatrix} = \sum_{i=1}^{N_r} a_i(t) \begin{pmatrix} \boldsymbol{\varphi}_i(\boldsymbol{x}) \\ \psi_i(\boldsymbol{x}) \\ \chi_i(\boldsymbol{x}) \\ \phi_i(\boldsymbol{x}) \end{pmatrix}. \tag{43}$$

The POD-FV-ROM for the RANS eddy viscosity model reads:

$$\begin{aligned} \frac{da_i(t)}{dt} = {} & \upsilon \sum_{i=1}^{N_r} B_{ji} a_i(t) + \upsilon \sum_{i=1}^{N_r} BT_{ji} a_i(t) - \sum_{k=1}^{N_r}\sum_{i=1}^{N_r} C_{jki} a_k(t) a_i(t) \\ & + \sum_{k=1}^{N_r}\sum_{i=1}^{N_r} CT1_{jki} a_k(t) a_i(t) + \sum_{k=1}^{N_r}\sum_{i=1}^{N_r} CT2_{jki} a_k(t) a_i(t) \end{aligned}$$

$$+\sum_{i=1}^{N_r} A_{ji} a_i(t) - \tau \left(\boldsymbol{u_{BC}} \cdot \boldsymbol{D}_j - \sum_{i=1}^{N_r} E_{ji} a_i(t) \right) \tag{44}$$

where the additional terms with respect to Eq. (33) are:

$$BT_{ji} = \left\langle \boldsymbol{\varphi}_j, \nabla \cdot \left(\nabla \boldsymbol{\varphi}_i{}^{\boldsymbol{T}} \right) \right\rangle_{L^2} \tag{45}$$

$$CT1_{jki} = \left\langle \boldsymbol{\varphi}_j, \phi_k \Delta \boldsymbol{\varphi}_i \right\rangle_{L^2} \tag{46}$$

$$CT2_{jki} = \left\langle \boldsymbol{\varphi}_j, \nabla \cdot \phi_k \left(\nabla \boldsymbol{\varphi}_i{}^{\boldsymbol{T}} \right) \right\rangle_{L^2}. \tag{47}$$

Please note that the $2/3k\boldsymbol{I}$ term is neglected in the ROM since it can be incorporated in the pressure term [38]. The dynamical system of the time-dependent coefficients for turbulence can be expressed as

$$\dot{\boldsymbol{a}} = \upsilon \left(\boldsymbol{B} + \boldsymbol{BT} \right) \boldsymbol{a} - \boldsymbol{a}^{\boldsymbol{T}} \left(\boldsymbol{C} - \boldsymbol{CT}1 - \boldsymbol{CT}2 \right) \boldsymbol{a} - \boldsymbol{A}\boldsymbol{a} + \tau \left(\boldsymbol{u_{BC}} \boldsymbol{D} - \boldsymbol{E}\boldsymbol{a} \right). \tag{48}$$

The procedure proposed in this work has the advantage to be more flexible possible and less dependent to the turbulent modelling. In particular, the approach can be applied to any model that expresses the momentum equation as Eq. (39), disregarding the specific modelling of the turbulent viscosity.[5] In addition, since in the POD-FV-ROM the diffusive term is not modified with the Green formula, the approach can be used also whether wall functions are applied. Indeed, problems may arise in the treatment of the wall functions if the BCs are directly incorporated in the ROM, since the wall functions of the turbulent quantities may be neither constant nor time-independent.

Even if the procedure proposed for the turbulent treatment is different from the classic one available in literature [16,39], Eq. (48) is similar to the system obtained if we apply a modal eddy viscosity [34,56,62]. In these works, the eddy viscosity υ_{EV} is mode dependent and it is obtained by solution matching [62] or power balance [56]. In this work, υ_t is expanded in spatial modes calculated according to RANS simulations.

4. Numerical simulation tests

In this section, the POD-FV-ROM is tested in the classic benchmark of the numerical simulation of the 2D lid-driven cavity [50–52]. It is worthwhile to remind that this is not a reduction from a 3D model to a 2D one and the 2D geometry is adopted for the sake of simplicity even if the approach can be easily extended to the 3D case. In addition, we focus on the numerical aspects and aims of the benchmark, leaving out all the physical implications [63]. Two simulations at Re $= 10^3$ and Re $= 10^5$ are considered in order to assess both the laminar and turbulent flows. The lid-driven cavity is used as classical test problem for the evaluation of numerical techniques and validation of incompressible Navier–Stokes codes due to the simplicity of both the geometry and boundary conditions, despite the presence of unphysical singularities at its corners [51]. The geometry of the benchmark is depicted in Fig. 3, i.e., a square cavity of unit length. The boundary conditions are $\mathbf{u} = (-1, 0)$ on the side $y = 1$ and $\mathbf{u} = (0, 0)$ on the other three sides.

As figures of merit to evaluate the behaviour of the POD-FV-ROM, the kinetic energy, the velocity, the pressure and the respective L^2 error norm are considered. The results of the FOM are taken as reference to assess the ROM. In addition, for the laminar case, the comparison between the ROM steady-state solution and the data available in literature is presented.

4.1. Implementation aspects

The offline procedure, i.e., the calculation of the FOM solutions, the POD bases creation and the construction of the matrices of Eq. (48), is performed in the OpenFOAM environment. OpenFOAM is an open source library for numerical simulation in continuum mechanics. The toolkit is very flexible thanks to the object-oriented programming, allowing users to customize, extend and implement complex physical model. Even if the POD creation is already

[5] This is due to the choice to not project the equations that govern the eddy viscosity behaviour as the turbulent kinetic energy (k) and the specific rate dissipation (ω) in the k-ω modelling.

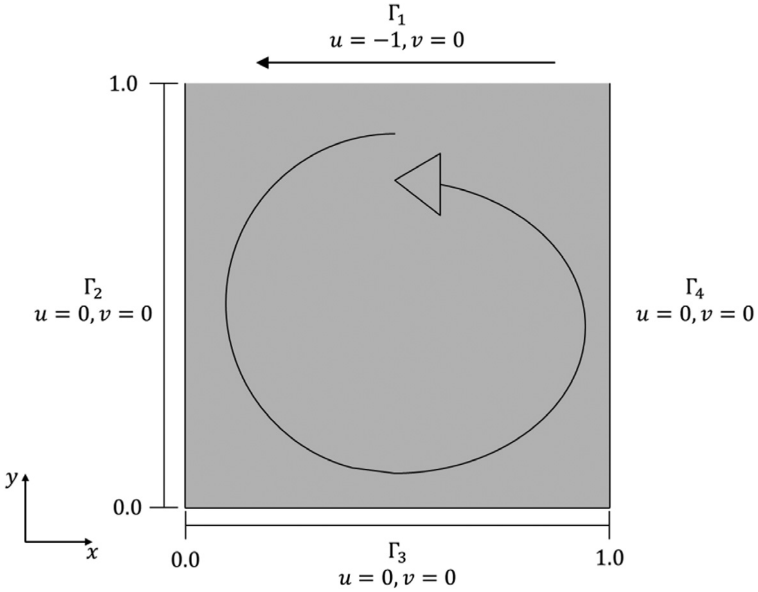
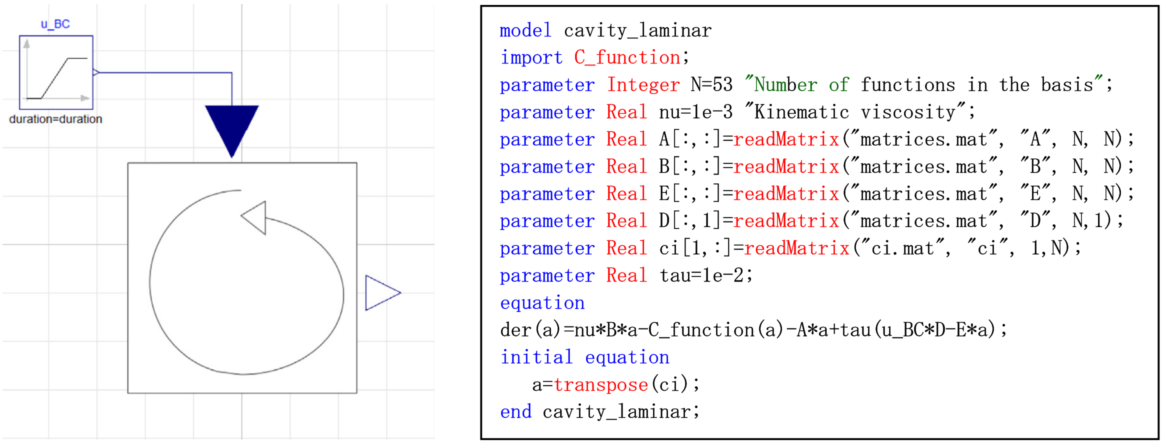


Fig. 3. Geometry and BCs of the 2D lid-driven cavity test case.



```
model cavity_laminar
import C_function;
parameter Integer N=53 "Number of functions in the basis";
parameter Real nu=1e-3 "Kinematic viscosity";
parameter Real A[:,:]=readMatrix("matrices.mat", "A", N, N);
parameter Real B[:,:]=readMatrix("matrices.mat", "B", N, N);
parameter Real E[:,:]=readMatrix("matrices.mat", "E", N, N);
parameter Real D[:,1]=readMatrix("matrices.mat", "D", N,1);
parameter Real ci[1,:]=readMatrix("ci.mat", "ci", 1,N);
parameter Real tau=1e-2;
equation
der(a)=nu*B*a-C_function(a)-A*a+tau(u_BC*D-E*a);
initial equation
   a=transpose(ci);
end cavity_laminar;
```

Fig. 4. Cavity component and Modelica code for the laminar case.

present in the extended version of OpenFOAM [64], some utilities have been updated or created in order to implement the offline phase of POD-FV-ROM in the library.

The online phase, i.e., the calculation of the ROM solutions and the implementation of the equation set of Eq. (48), is performed with the object-oriented Modelica language [65,66] in Dymola simulation environment [67,68]. The modelling approach consists in connecting different components (i.e., objects described by equations), through rigorously defined interfaces (connectors) corresponding to the physical interactions occurring with the external environment or other objects. In the ROM framework, Modelica with its component approach represents a powerful tool since it is possible to update or substitute a component with the respective ROM without compromising the rest of the model. The component "cavity_laminar" with the Modelica code used to implement the cavity ROM for the laminar case is shown in Fig. 4.

For the sake of clarity, some numerical clarifications should be pointed out. As for the Full Order Model, a merged PISO-SIMPLE (PIMPLE) algorithm [25,49] available in OpenFOAM is employed to solve the NSE and the RANS in order to obtain for each time step at least a convergence of 10^{-5} for the main variables (i.e., velocity, pressure, turbulent quantities, etc.). The fourth-order numerical schemes for spatial discretization available in OpenFOAM

Table 1
Laminar case ($\mathrm{Re} = 10^3$). Number of basis functions versus truncation error.

N	e_{POD}
1	$2.1 \cdot 10^{-2}$
5	$5.2 \cdot 10^{-4}$
10	$3.0 \cdot 10^{-5}$
15	$2.5 \cdot 10^{-6}$
20	$2.5 \cdot 10^{-7}$
30	$2.3 \cdot 10^{-9}$
40	$2.0 \cdot 10^{-11}$
50	$1.2 \cdot 10^{-12}$
53	$1.0 \cdot 10^{-12}$

are used, whereas a second order Backward Differentiation Formula (BDF) implicit scheme is adopted for the time discretization. A (1024 × 1024) structured equispaced mesh is employed to obtain an accurate solution without compromising the ROM performance since the ROM computational time does not depend on the degree of freedom of the FOM. The offline procedure was performed on the CINECA GALILEO cluster with 64 processors. As for the online phase, the integration algorithm DASSL [69] is used to solve the ODE system with a 10^{-5} relative tolerance. The ROM simulation is carried out on a personal computer (single processor).

4.2. Laminar case

In the first simulation, an unsteady incompressible laminar flow at $\mathrm{Re} = 10^3$ (i.e., $\upsilon = 10^{-3}$) in the 2D lid-driven cavity is considered. Let $\Omega = (0, 1) \times (0, 1)$ and $T > 0$ the simulation time, the governing equations read

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} - \upsilon \Delta \boldsymbol{u} + \nabla p = 0 & \text{on } [0, T] \ x \ \Omega \\ \nabla \cdot \boldsymbol{u} = 0 & \text{on } [0, T] \ x \ \Omega \\ \boldsymbol{u} = (-1, 0) & \text{on } [0, T] \ x \ \Gamma_1 \\ \boldsymbol{u} = (0, 0) & \text{on } [0, T] \ x \ \Gamma_{2,3,4} \\ \boldsymbol{u} = (0, 0) & \text{on } [0] \ x \ \Omega. \end{cases} \tag{49}$$

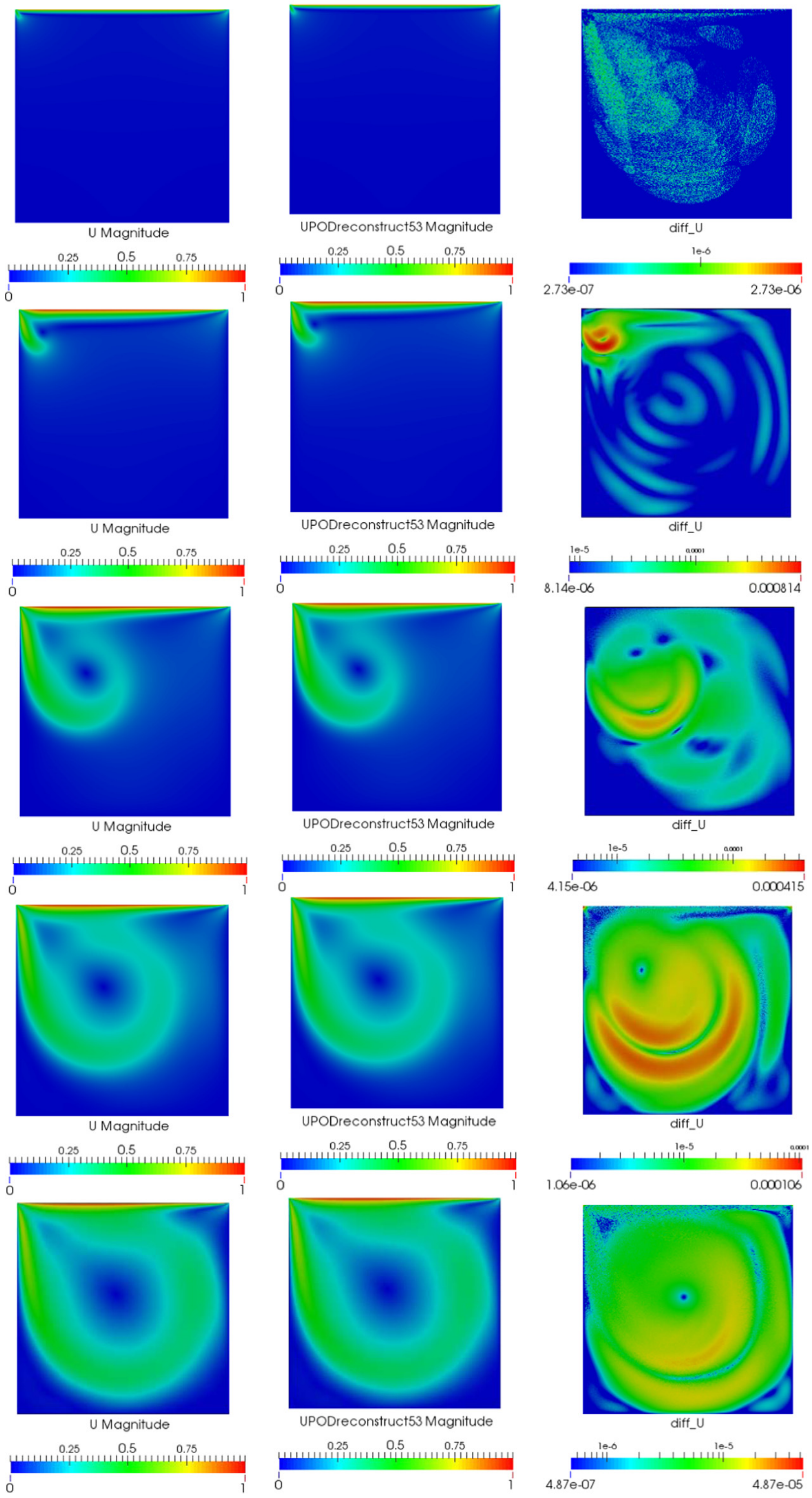
The full order simulation was performed until $T = 100$ s with a constant time step of $5 \cdot 10^{-4}$ s. Figs. 5(a) and 6(a) present the velocity and the pressure of the full order simulation at different times. A total number of 1000 snapshots were considered, sampling the velocity and the pressure each 0.1 s. During the offline phase, the basis for the velocity, the face flux and the pressure was calculated following the procedure described in Section 2.2.

As for the POD bases creation, the first three functions of the velocity and pressure bases are shown in Fig. 7 and the decay of the normalized POD eigenvalues is given in Fig. 8. From the latter figure, it is possible to establish the number of the functions in the basis given a "tolerance" on the normalized eigenvalues.[6] For the laminar case, 50–60 basis functions are sufficient to keep the normalized eigenvalue below 10^{-12}–10^{-14}. In particular, $N_r = 53$ basis functions are used for the laminar case hereinafter in order to obtain a truncation error of 10^{-12} (Table 1). The truncation error of the POD procedure is defined as

$$e_{POD}\,(N) = 1 - \sum_{i=1}^{N} \lambda_i. \tag{50}$$

Once having calculated the matrices of Eq. (33), the set of the ODEs is simulated in the Dymola environment. A sensitivity analysis was performed to establish a proper value of the penalizing factor τ between 10^{-5} and 100. A value of $\tau = 10^{-2}$ is found to be sufficient to enforce the BCs without afflicting the ODE system with ill-conditioning

[6] It is important to remind that, even if the sum of the eigenvalues can be usually referred to the energy content in the snapshots, this number is not indicative of the relative error of the reduced order model.

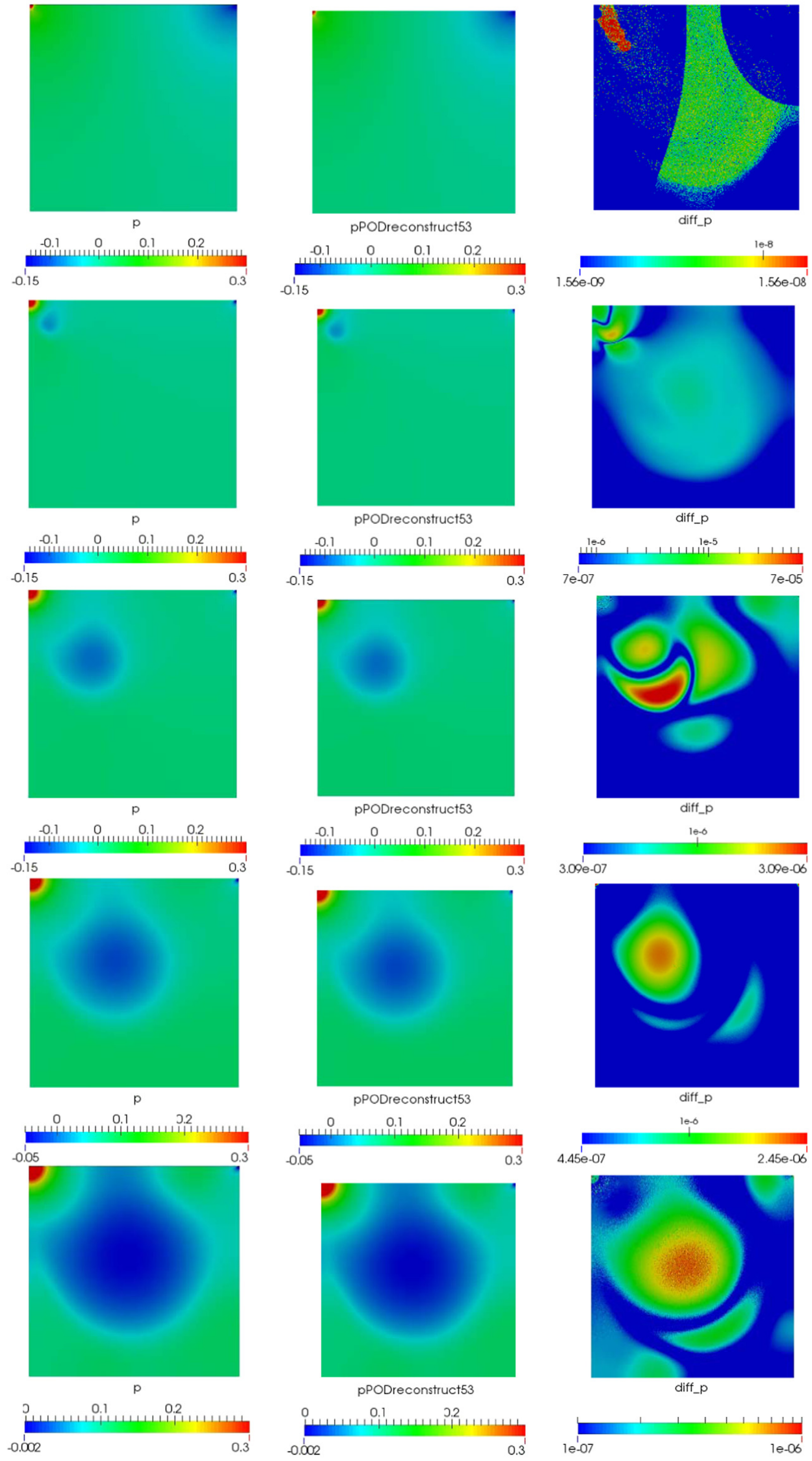


(a) FOM.

(b) ROM.

(c) Difference.

Fig. 5. Laminar case (Re $= 10^3$), contours of the velocity magnitude of the full order model (a), the reduced order model with $N_r = 53$ (b) and relative difference (c) at different times. From top to bottom 0.1, 1, 5, 10, 25 s.

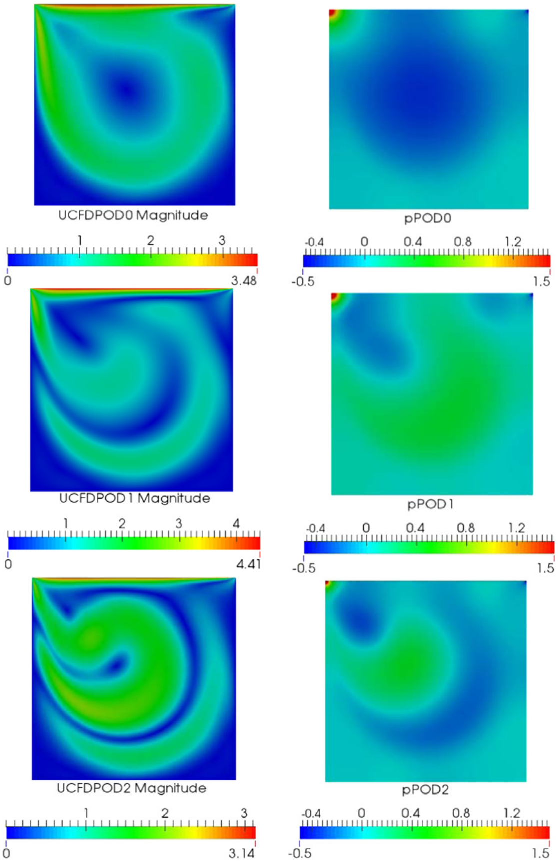


(a) FOM.

(b) ROM.

(c) Difference.

Fig. 6. Laminar case (Re $= 10^3$), contours of the pressure of the full order model (a), the reduced order model with $N_r = 53$ (b) and relative difference (c) at different times. From top to bottom 0.1, 1, 5, 10, 25 s.



(a) Velocity.
(b) Pressure.

Fig. 7. Laminar case (Re $= 10^3$), first three basis functions for the velocity (a) and pressure (b).

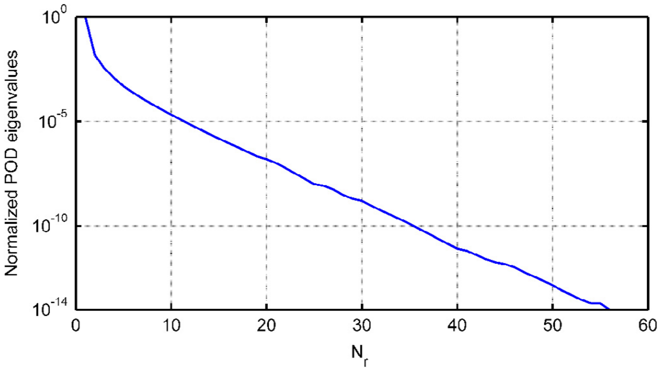


Fig. 8. Laminar case (Re $= 10^3$), normalized POD eigenvalues.

problems. A first check to assess the accuracy of the POD-FV-ROM is to calculate the relative error of the kinetic energy between FOM and ROM (in Fig. 9 this value is shown for different N_r). As for $N_r = 53$, the results are satisfactory being the error always lower than $4 \cdot 10^{-4}$. Moreover, in the second part of the transient, the error is positive indicating that the reduced order model is slightly underestimating the energy content [46–48]. This means that the POD-FV-ROM is not affected by the energy blow up mentioned in the Introduction, even if this problem is more related to turbulent flows than laminar ones.

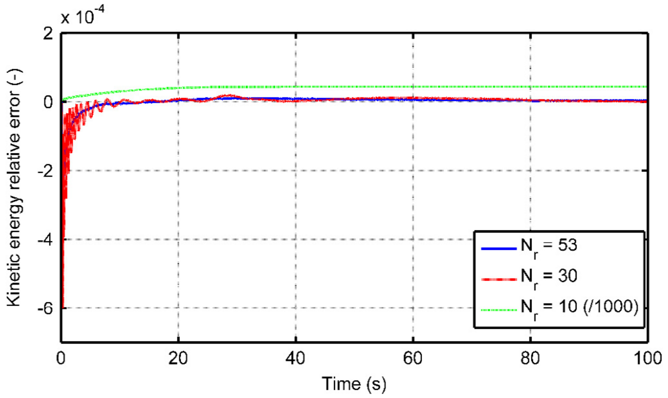


Fig. 9. Laminar case (Re $= 10^3$), kinetic energy relative difference between the FOM and ROM ($N_r = 53$, $N_r = 30$ and $N_r = 10$). The value for $N_r = 10$ is divided by 1000 for the sake of readability.

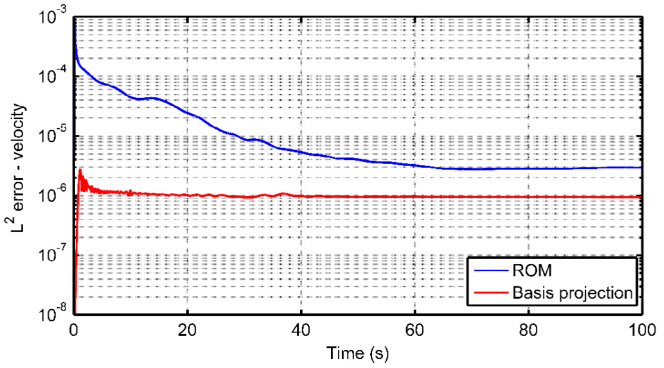


Fig. 10. Laminar case (Re $= 10^3$), L^2 error of the reconstructed velocity from ROM with $N_r = 53$ (blue line) and basis projection (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Figs. 5(b) and 6(b) present the velocity and the pressure field reconstructed starting from the POD time coefficients whereas their relative difference with respect to the FOM results is depicted in Figs. 5(c) and 6(c). In order to give a numerical estimation of the discrepancies introduced by the ROM, the L^2 error is introduced. For instance, for the velocity it reads:

$$\|e\|_{L^2} = \sqrt{\frac{\langle (\boldsymbol{u_{FOM}} - \boldsymbol{u}_r), (\boldsymbol{u_{FOM}} - \boldsymbol{u}_r) \rangle_{L^2}}{\langle \boldsymbol{u_{FOM}}, \boldsymbol{u_{FOM}} \rangle_{L^2}}}. \tag{51}$$

In Fig. 10, the L^2 error for the ROM velocity (blue line) is compared with the L^2 error of the velocity reconstructed from the coefficient given by the projection of the snapshots onto the POD functions (red line). The latter can be considered a “reference” value given a fixed number of basis functions. The error is always lower than 10^{-3}, being higher in the first part of the transient mainly because the flow starts at rest and the error is magnified by the small velocity magnitude. At the end of the transient, the error is set lower than 10^{-5}. The L^2 error of the ROM is satisfying considering the model order reduction purposes and the reduced computational cost of the ROM.

In addition, the capability to correctly reproduce the solution in time steps that are not sampled in the snapshots was studied. In Fig. 11, the L^2 error (between the ROM and FOM) in time steps that are not sampled in the snapshots (red cross) is compared to the error in sampled snapshots (blue circle — sampling time of 0.1 s). The results indicate a good prediction also in time steps that are not in the ensemble of the snapshots.

As for the pressure error (Fig. 12), a good result is obtained since the error is between 10^{-3} and 10^{-6}. It is interesting to notice that in the second part of the transient, the ROM pressure error is lower than the basis projection ones. This is not surprising considering that the time coefficients for ROM pressure are the same with respect to the ROM

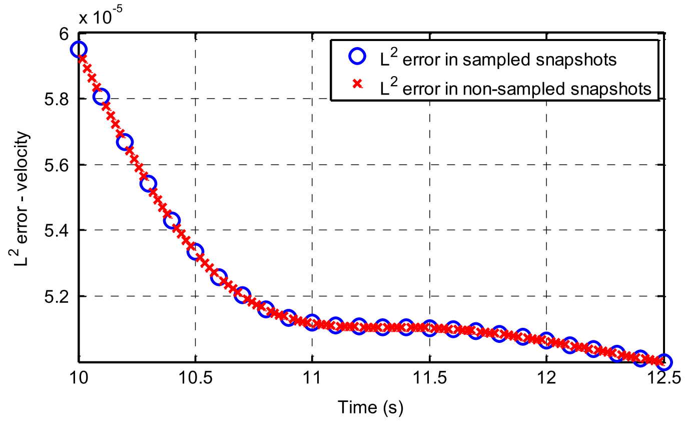


Fig. 11. Laminar case (Re $= 10^3$), L^2 error of the reconstructed velocity from ROM with $N_r = 53$. Comparison between error in time steps that are not sampled in the snapshots (red cross) and the error in time steps referring to the sampled snapshots (blue circle). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

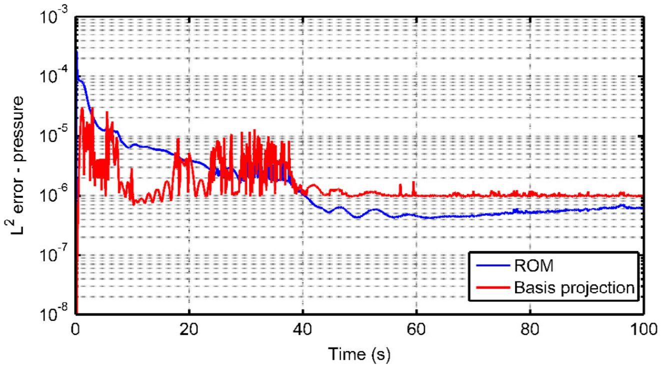


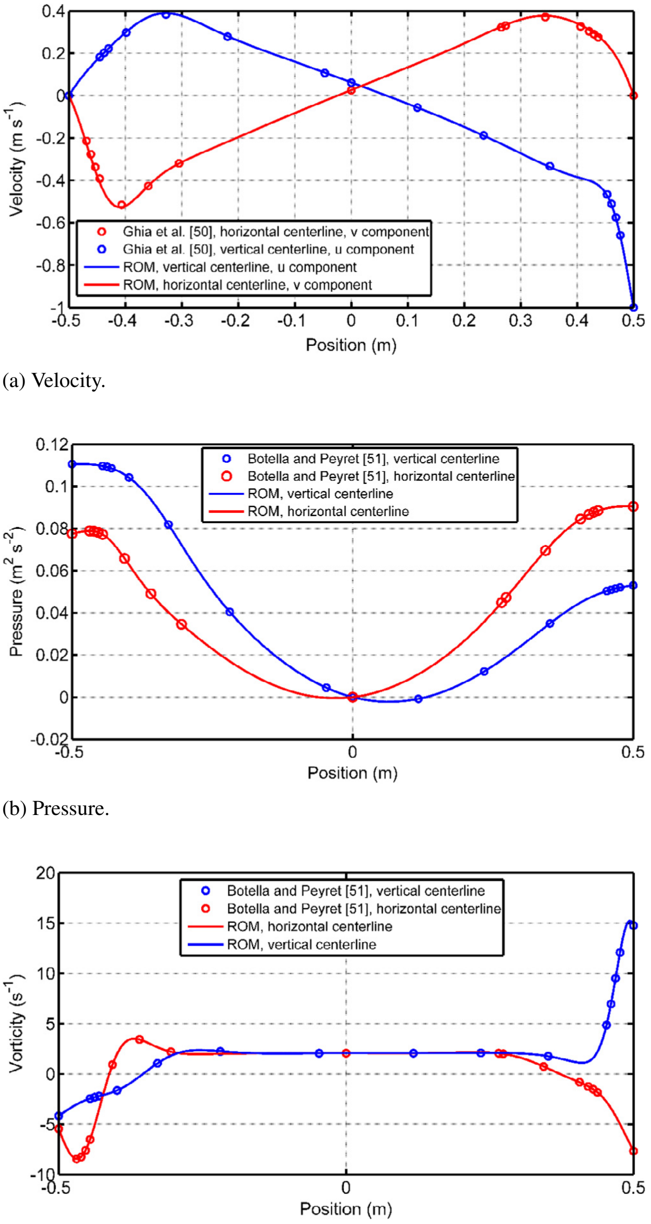
Fig. 12. Laminar case (Re $= 10^3$), L^2 error of the reconstructed pressure from ROM with $N_r = 53$ (blue line) and basis projection (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Stream function values shown in Fig. 14 [51].

Stream function							
Value	$1.175 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	10^{-1}	$9 \cdot 10^{-2}$	$7 \cdot 10^{-2}$	$5 \cdot 10^{-2}$
Label	a		b		c		d
Value	$3 \cdot 10^{-2}$	10^{-2}	10^{-4}	10^{-5}	10^{-10}	0	-10^{-6}
Label		e	f				
Value	-10^{-5}	$-5 \cdot 10^{-5}$	-10^{-4}	$-2.5 \cdot 10^{-4}$	$-5 \cdot 10^{-4}$	-10^{-3}	$-1.5 \cdot 10^{-3}$
Label	g		h		i		j

velocity and the spatial basis for pressure is calculated from the eigenvectors of the correlation matrix of the velocity (see Eqs. (21) and (24)).

As last comparison, a benchmark with literature data [50,51] is also undertaken to prove the accuracy of the reduced order model. In particular, the ROM velocity at the end of the transient is compared with the steady-state solution of the lid-driven cavity at Re $= 10^3$. Fig. 13 presents the comparison through the horizontal and the vertical centreline of the cavity for the velocity, the pressure and the vorticity. As for the streamlines and the vorticity, the contours of these quantities are compared in Figs. 14 and 15, respectively (see Tables 2 and 3 for the contour values). The almost perfect agreement between the literature data and the ROM results confirm the reliability of the POD-FV-ROM procedure for the laminar case.



(a) Velocity.

(b) Pressure.

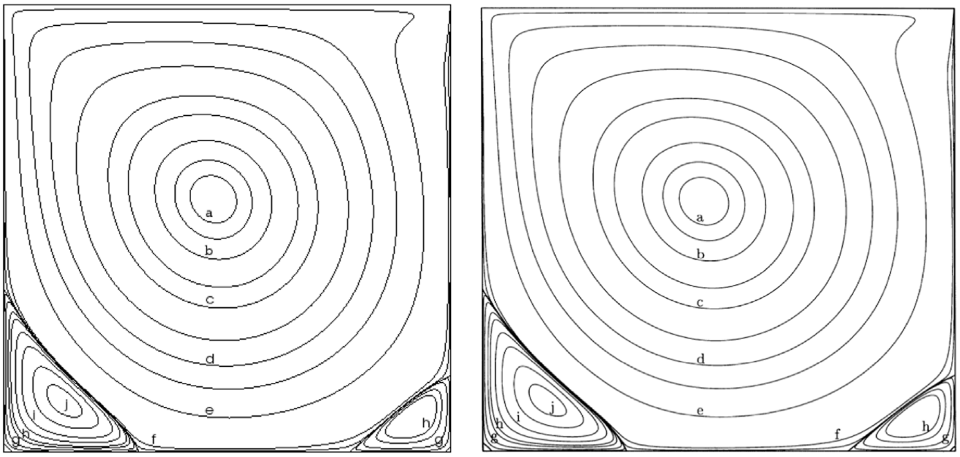
(c) Vorticity.

Fig. 13. Laminar case (Re = 10^3), comparison of velocity, pressure and vorticity between ROM results and the literature data presented in Ghia's and Botella's work [50,51] along horizontal and vertical centrelines.

Table 3
Vorticity contours shown in Fig. 15 [51].

Vorticity											
Value	5	4	3	2	1	0.5	0	−0.5	−1	−2	−3
Label	a	b	c	d	e	f	g	h	i	j	k

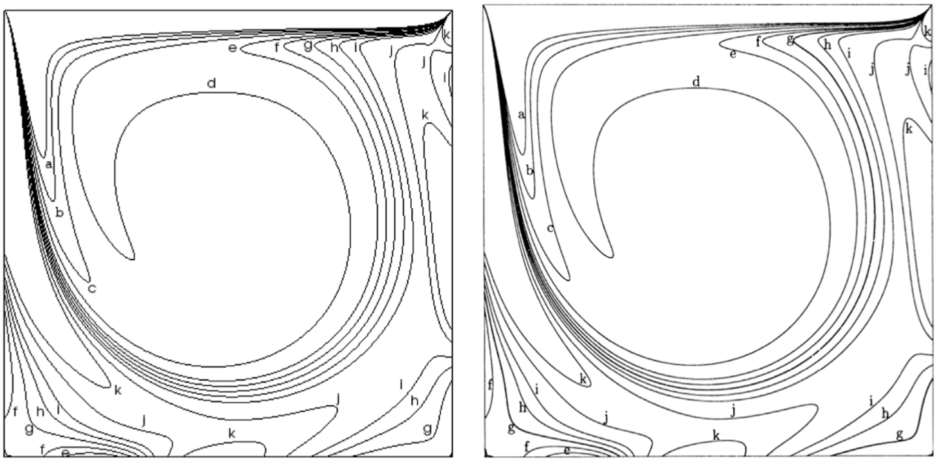
Finally, for the full order model, a computational time of 2590 cpu-hours is required for the simulation of $T = 100$ s. On the other hand, for the ROM, only a computational time of 5 s is needed to perform the same



(a) Present work.

(b) Botella’s work [51].

Fig. 14. Laminar case (Re $= 10^3$), streamline contours (see Table 2 for stream function values) for the present work and Botella’s work [51].



(a) Present work.

(b) Botella’s work [51].

Fig. 15. Laminar case (Re $= 10^3$), vorticity contours (see Table 3 for stream function values) for the present work and Botella’s work [51].

simulation. For the sake of completeness, for the POD generation and the ROM matrix calculation of the offline step, the procedure takes 150 and 180 min, respectively.

4.3. Turbulent case

In the second test simulation, an unsteady incompressible turbulent flow at Re $= 10^5$ (i.e., $\upsilon = 10^{-5}$) is considered. According to Section 2.3, any RANS approach with turbulent viscosity model can be applied to the POD-FV-ROM for turbulent flows. Nevertheless, for this simulation, the Shear Stress Transport (SST) formulation of the $k - \omega$ modelling [70,71] is selected for two main reasons. The first one is that SST $k - \omega$ is a low-Reynolds model and the mesh can be refined without compromising the solution stability. In this way, we provide the ROM with a fine mesh solution. The second reason to select SST k-ω relies on the fact that this model blends the k-ω features near the wall and the k-ε behaviour in the bulk flow. Let $\Omega = (0, 1) \times (0, 1)$ and $T > 0$ the simulation time, the governing equations read (for further information about the model parameter and functions, please refer to [49,70,71])

$$
\begin{cases}
\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = \nabla\cdot\left[-p\boldsymbol{I} + (\nu+\nu_t)\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) - \dfrac{2}{3}k\boldsymbol{I}\right] & \text{on } [0,T]\ x\ \varOmega \\
\nabla\cdot\boldsymbol{u} = 0 & \text{on } [0,T]\ x\ \varOmega \\
\boldsymbol{u} = (-1,0) & \text{on } [0,T]\ x\ \varGamma_1 \\
\boldsymbol{u} = (0,0) & \text{on } [0,T]\ x\ \varGamma_{2,3,4} \\
\boldsymbol{u} = (0,0) & \text{on } [0]\ x\ \varOmega \\
\dfrac{\partial k}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,k = \nabla\cdot\left[(\nu+\nu_T\alpha_k)\,\nabla k\right] - \beta^* k\omega + P & \text{on } [0,T]\ x\ \varOmega \\
\dfrac{\partial\omega}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\omega = \nabla\cdot\left[(\nu+\nu_T\alpha_\omega)\,\nabla\omega\right] & \\
\quad -\beta\omega^2 + \dfrac{\gamma}{\nu_T}P + 2\,(F_1-1)\,\dfrac{\alpha_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega & \text{on } [0,T]\ x\ \varOmega.
\end{cases}
\tag{52}
$$

The full order simulation is performed until $T = 100$ s with a constant time step of $4\cdot 10^{-4}$ s. A total number of 1000 snapshots were considered, sampling the velocity, the pressure and the eddy viscosity each 0.1 s. Figs. 16(a), 17(a) and 18(a) present the velocity, the pressure and the eddy viscosity of the full order simulation at different times. During the offline phase, the basis for the velocity, the face flux, the pressure and the eddy viscosity is calculated following the procedure described in Section 2.3.

As for the POD bases creation, the first three functions of the velocity, pressure and turbulent viscosity bases are shown in Fig. 19 and the decay of the normalized POD eigenvalues is given in Fig. 20. Considering also the decay trend, $N_r = 85$ basis functions are used for the turbulent case hereinafter in order to reach a good compromise between accuracy and computational requirements. The number of the basis functions is increased with respect to the laminar case as expected due to the increase of the Reynolds number and the flow complexity.

Once having calculated the matrices of Eq. (44), the set of the ODEs is simulated in the Dymola environment. The penalizing value τ is fixed at 10^{-2}. Due to the energy blow up mentioned in the Introduction afflicting the turbulent flows, the relative error of the kinetic energy between FOM and ROM could be significant (in Fig. 21, this value is shown for different N_r). As for $N_r = 85$, except in the very initial moment of the transient, the relative error is around 10^{-3}, indicating that the ROM is not affected by energy blow up. In particular, in the second part of the transient, even if the error is negative (i.e., the energy in ROM becomes slightly greater than the energy in FOM), the energy difference is reduced along the simulation time. As expected, the kinetic energy relative error is increased of one order of magnitude with respect to the laminar case due to the increase of the Reynolds number. Nonetheless, the value is acceptable and the boundedness of the error can give some good insight about the stability of the reduced order model.

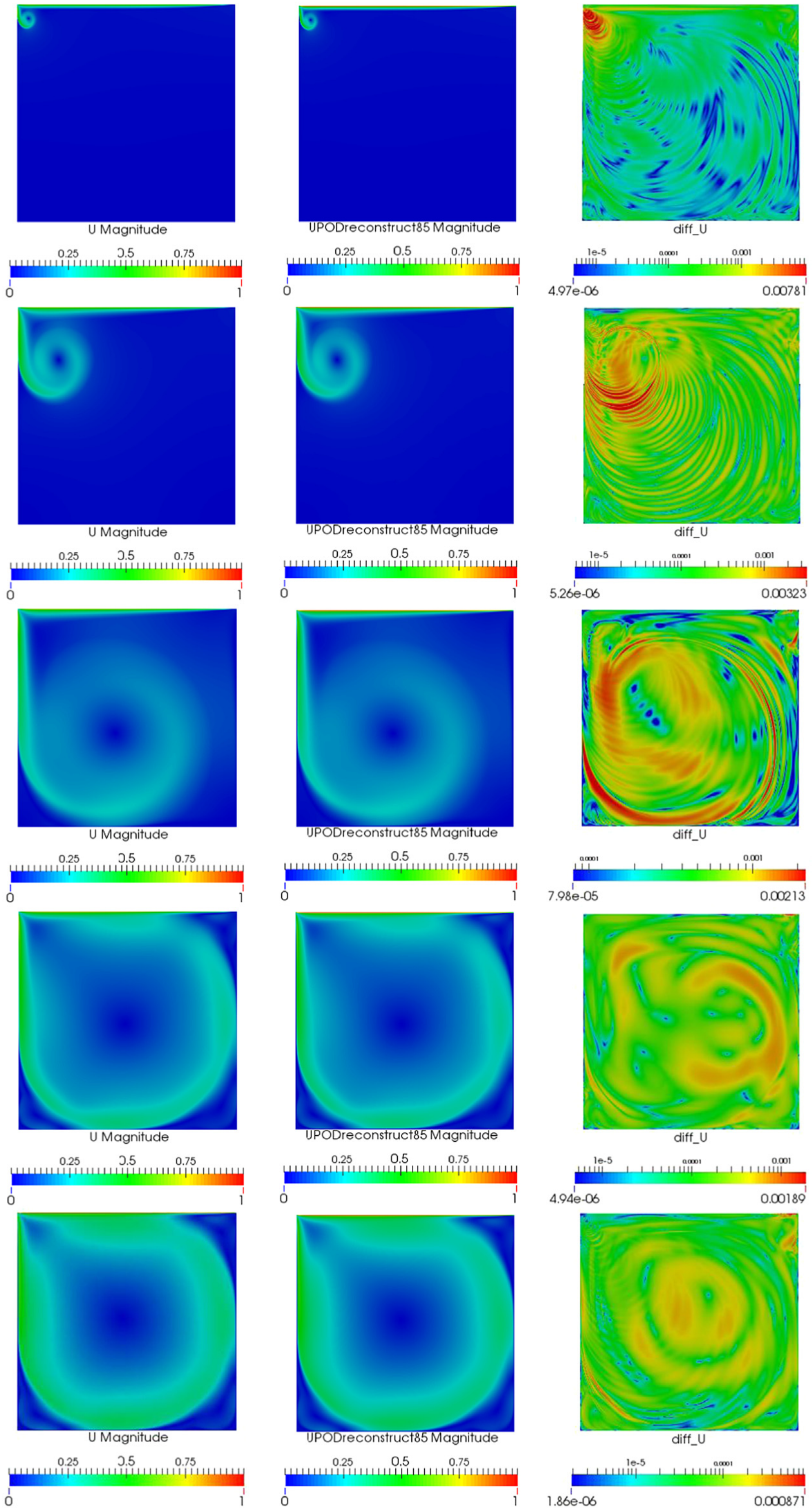
Figs. 16(b), 17(b) and 18(b) present the velocity, the pressure and the turbulent viscosity field reconstructed starting from the POD time coefficients. The relative difference with respect to the FOM results is depicted in Figs. 16(c), 17(c) and 18(c).

In Fig. 22, the L^2 error for the ROM velocity (blue line) is compared with the L^2 error of the velocity reconstructed from the coefficient given by the projection of the snapshots onto the POD functions (red line). Also in this case, the error is higher in the first part of the transient since the velocity magnitude starts at rest (see initial condition, Eq. (52)). After 15 s, the error is lower than $6\cdot 10^{-3}$ and at the end of the transient, it is set to $3.5\cdot 10^{-4}$.

Similar results are obtained for the pressure error (Fig. 23). The error in the first instances is $4\cdot 10^{-2}$, but just after some seconds is kept lower than $8\cdot 10^{-3}$ for the entire transient.

The error for the turbulent viscosity is shown in Fig. 24. The results seem to be less satisfactory with respect to the velocity and pressure but it should consider that, in this work, as for the pressure, the spatial basis for the turbulent viscosity is calculated from the eigenvectors of the correlation matrix of the velocity and therefore they are not tailored for the turbulent viscosity. In this way, only the momentum equation is projected in the spatial basis, disregarding the complicated equations for the turbulent quantities. Nevertheless, other options can be undertaken as the expansion of k and ω along with the projection of all the equations constituting the turbulence modelling. Notwithstanding the maximum L^2 error is 10^{-1} at the beginning of the transient and after some seconds it sets to 10^{-2} for the rest of the transient.

Finally, for the full order model, a computational time of 4462 cpu-hours is required for the simulation of $T = 100$ s. On the other hand, for the ROM, only a computational time of 45 s is needed to perform the same

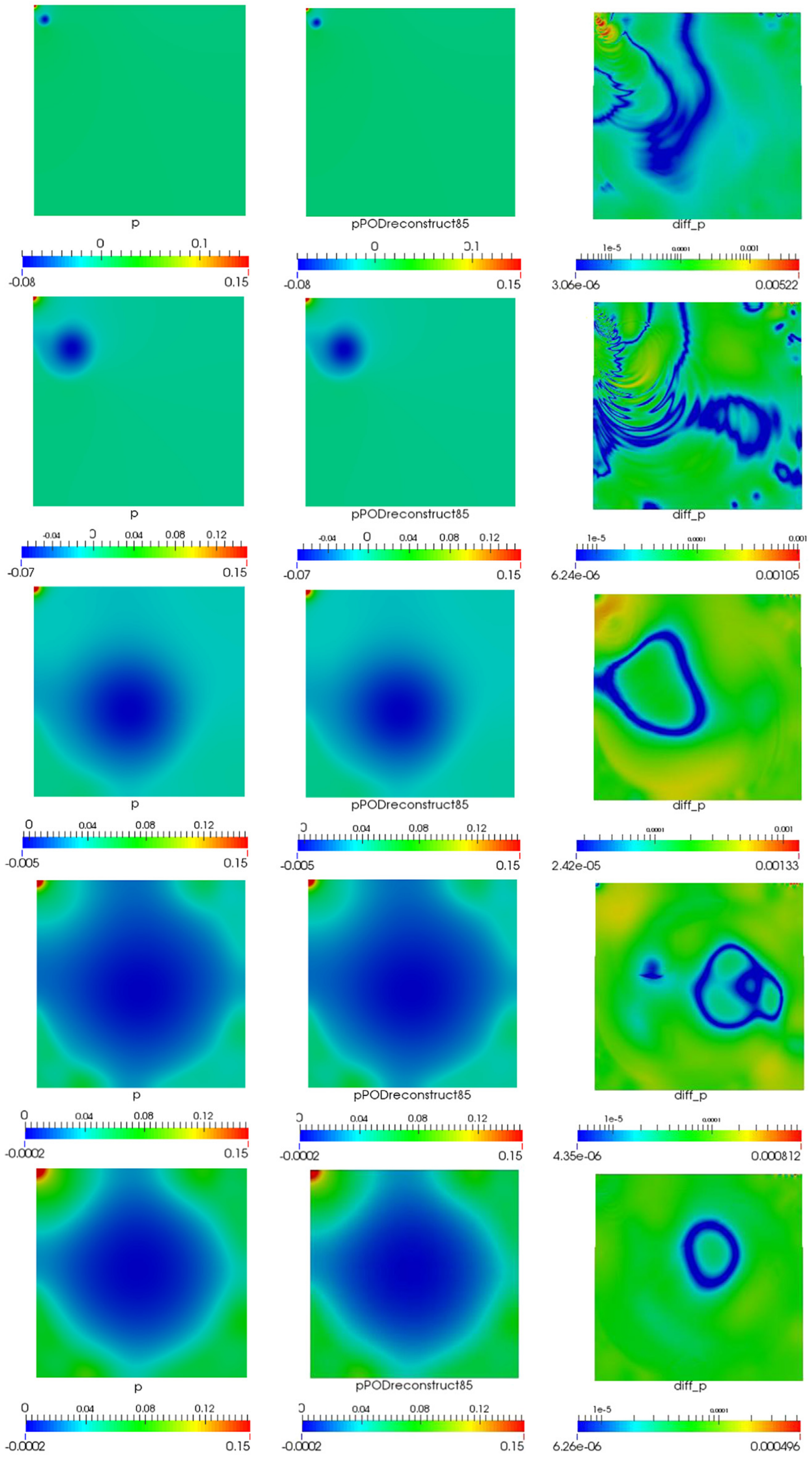


(a) FOM.

(b) ROM.

(c) Difference.

Fig. 16. Turbulent case (Re $= 10^5$), contours of the velocity magnitude of the full order model (a), the reduced order model with $N_r = 85$ (b) and relative difference (c) at different times. From top to bottom 1, 5, 20, 40, 100 s.

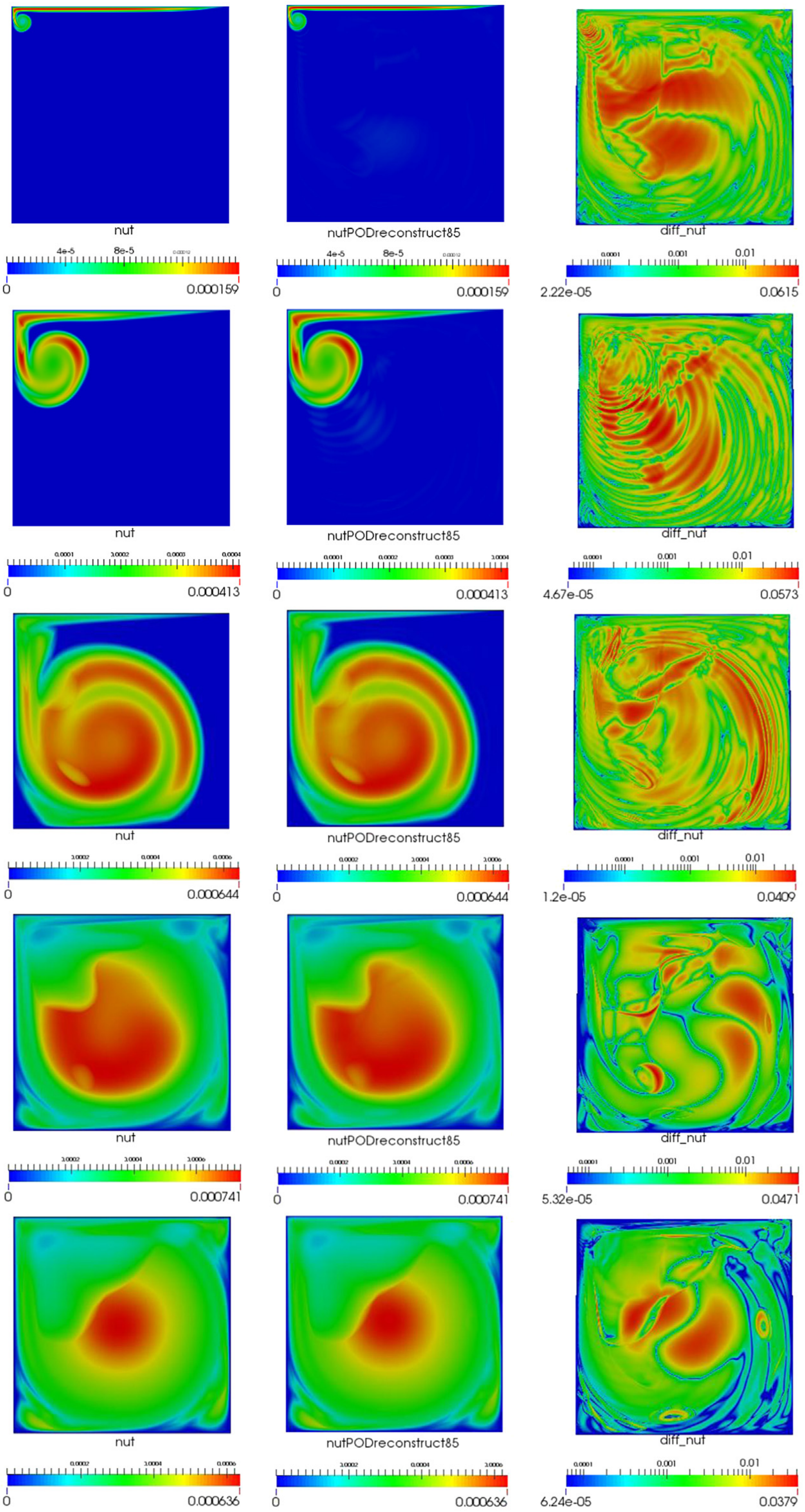


(a) FOM.

(b) ROM.

(c) Difference.

Fig. 17. Turbulent case (Re $= 10^5$), contours of the pressure of the full order model (a), the reduced order model with $N_r = 85$ (b) and relative difference (c) at different times. From top to bottom 1, 5, 20, 40, 100 s.

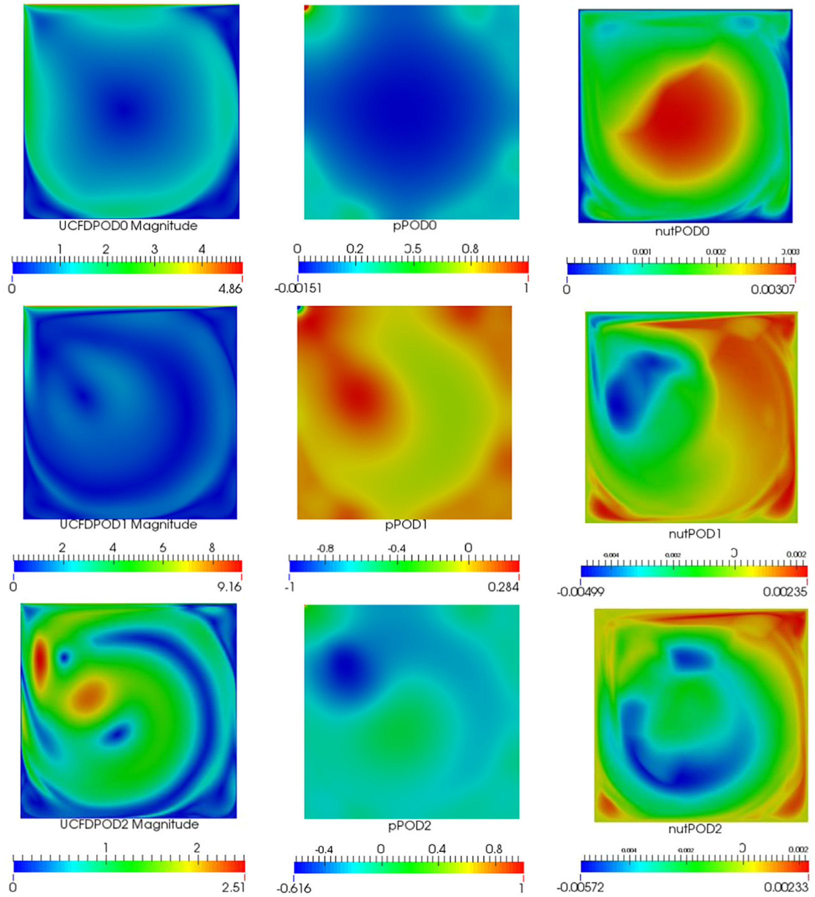


(a) FOM.

(b) ROM.

(c) Difference.

Fig. 18. Turbulent case (Re $= 10^5$), contours of the eddy viscosity of the full order model (a), the reduced order model with $N_r = 85$ (b) and relative difference (c) at different times. From top to bottom 1, 5, 20, 40, 100 s.



(a) Velocity.

(b) Pressure.

(c) Turbulent viscosity.

Fig. 19. Turbulent case (Re $= 10^5$), first three basis functions for the velocity (a), pressure (b) and turbulent viscosity (c).

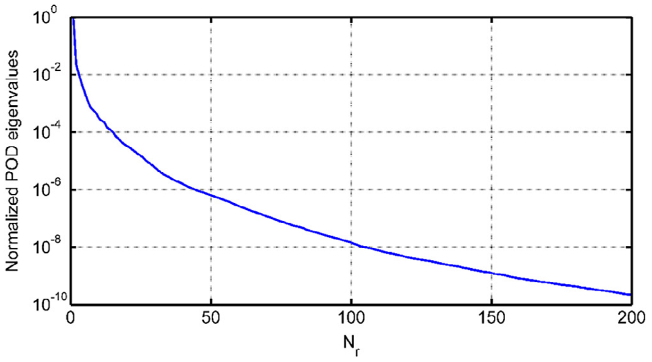


Fig. 20. Turbulent case (Re $= 10^5$), normalized POD eigenvalues.

simulation. For the sake of completeness, for the POD generation and the ROM matrix calculation of the offline step, the procedure takes 3 cpu-hours and 760 cpu-hours, respectively.

5. Conclusion

In this work, a POD-Galerkin Method for Finite Volume Approximation of Navier–Stokes and RANS equations is presented. In particular, the aim of the paper is both to extend the classic POD-Galerkin-ROM approach to the Finite

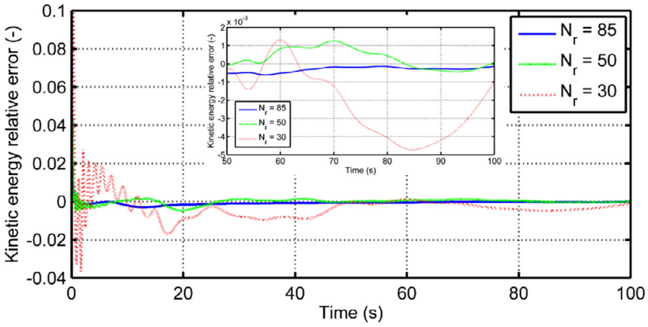


Fig. 21. Turbulent case (Re $= 10^5$), kinetic energy relative difference between the FOM and ROM ($N_r = 85$, $N_r = 50$ and $N_r = 30$).

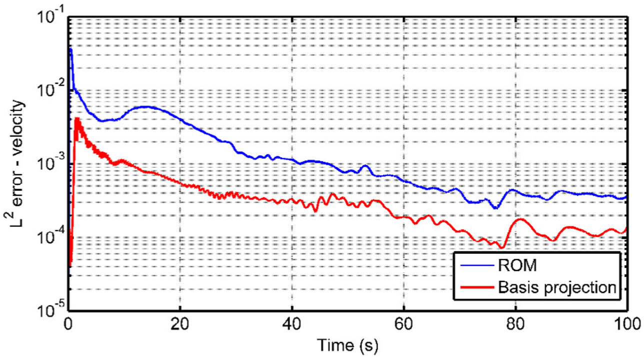


Fig. 22. Turbulent case (Re $= 10^5$), L^2 error of the reconstructed velocity from ROM with $N_r = 85$ (blue line) and basis projection (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

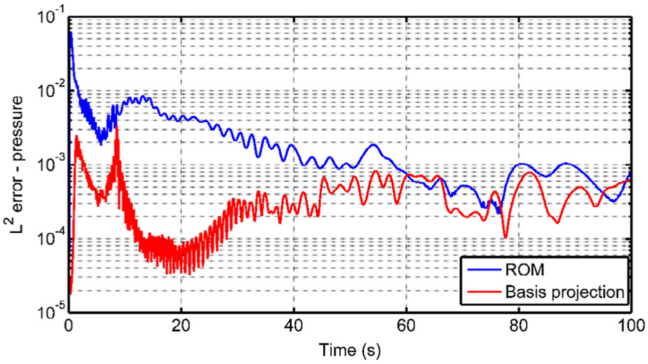


Fig. 23. Turbulent case (Re $= 10^5$), L^2 error of the reconstructed pressure from ROM with $N_r = 85$ (blue line) and basis projection (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Volume approximation of the Navier–Stokes equations and to build a ROM which is capable to handle turbulent flows modelled through the RANS equations.

The POD-FV-ROM procedure is described starting from the modifications to the classic POD-G-ROM that should be adopted in a FV framework. A different approach from the literature cases that usually employ LES and DNS simulations for the snapshots creation is adopted to treat the turbulence. The eddy viscosity is expanded as linear combination of spatial modes, but the time coefficients are the same for velocity, face flux, pressure, and turbulent viscosity, and the spatial basis is calculated from the eigenvectors of the correlation matrix of the velocity. In this way, only the momentum equation is projected in order to obtain the set of ROM equations. Moreover, this procedure is not tailored to a specific turbulent model and can be applied to any eddy viscosity model, for all the different equations related to the turbulent quantities.

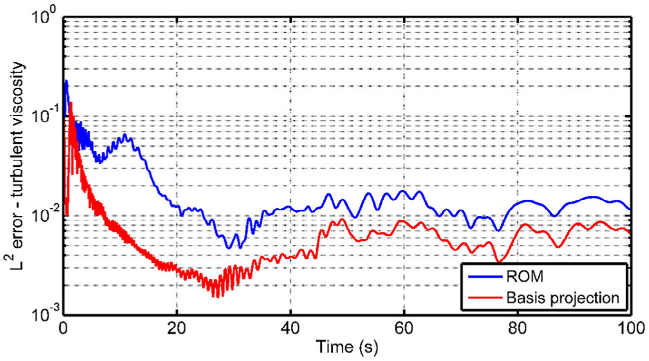


Fig. 24. Turbulent case ($\mathrm{Re} = 10^5$), L^2 error of the reconstructed turbulent viscosity from ROM with $N_r = 85$ (blue line) and basis projection (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The POD-FV-ROM has been tested in the classic benchmark of the numerical simulation of the 2D lid-driven cavity. In particular, two simulations at $\mathrm{Re} = 10^3$ and $\mathrm{Re} = 10^5$ have been considered in order to assess both laminar and turbulent flows. The results have turned out to be very satisfactory. As for the laminar case, the ROM with 53 spatial modes reproduces very accurate results for both velocity and pressure, keeping the error well below 10^{-3} in any times of the transient. The time performance is also good with a simulation time of 5 s with respect to the 2590 cpu-hours of the FOM simulation. On the other hand, for the turbulent case, a higher number of spatial modes ($N_r = 85$) is required as expected due to the higher Reynolds number. Considering the engineering-oriented purposes of the work, the outcomes are satisfactory being the error of the main variables of interest, i.e., velocity and pressure, almost always below 10^{-2} during the transient. Moreover, as the study of the kinetic energy has proved, the POD-FV-ROM is not affected by the energy blow up issue characteristic of the classic turbulent ROM [46–48]. Even if a higher number of spatial modes are needed to obtain a fair reproduction of the transient, the ROM simulation time is 45 s against a FOM time of 4462 cpu-hours. This is very important since in many applications the real-time response of the model is required.

As future developments of this work, the coupling with the energy equation is envisaged in order to extend the possible application of this approach also to multi-physics problem involving for example power production. In particular, the development of a ROM of a nuclear reactor pool is foreseen, which is intended to be employed in a control-oriented simulator. In the light of this development, it is relevant to develop a parametrized ROM [1], both in physical and geometrical frameworks. For this purpose, it could be necessary employing stabilization techniques also for laminar flows [72]. In addition, the predictive capability of the POD-FV-ROM will be the subject of our future work, assessing this aspect also in case of parametric POD (different Reynolds numbers, geometry parameters or boundary conditions).

Acknowledgements

The work has been carried out in the framework of a collaboration between Politecnico di Milano and the SISSA (International School for Advanced Studies), including a detachment of the first author at the SISSA—mathLab Department. The authors are grateful to F. Ballarin, G. Pitton, A. Sartori (SISSA, Trieste) for the fruitful conversation and suggestions about the reduced order modelling. Acknowledgement is also due to the SISSA—MathLab Department people for their contributions in many different topics.

The computing resources have been provided by the Sis14_COGESTA and Sis15_COGESTA cpu time grant allocation at CINECA supercomputing center, Bologna, Italy.

References

[1] 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 (2008) 229–275.

[2] A. Quarteroni, G. Rozza, A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, J. Math. Ind. 1 (2011) 3.

[3] M.D. Gunzburger, Perspectives in Flow Control and Optimization in Advances in Design and Control, SIAM, 2003.

[4] F. Chinesta, A. Huerta, G. Rozza, K. Wilcox, Model order reduction, in: Erwin Stein, Ren´e de Borst, Tom Hughes (Eds.), Encyclopedia of Computational Mechanics, second ed., John Wiley & Sons, Ltd., New York, 2016.

[5] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, in: SpringerBriefs in Mathematics, Springer International Publishing, 2016.

[6] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives, in: A. Quarteroni, G. Rozza (Eds.), Reduced Order Methods for Modeling and Computational Reduction, in: MS&A Series, vol. 9, Springer Milano, 2013, pp. 235–274.

[7] A. Manzoni, A. Quarteroni, G. Rozza, Computational reduction for parametrized PDEs: Strategies and applications, Milan J. Math. 80 (2) (2012) 283–309.

[8] N.H. El-Farra, P.D. Christofides, Hybrid control of parabolic PDE systems, in: Proceedings of the 41st IEEE Conference on Decision and Control, 2002.

[9] A. Barbagallo, D. Sipp, P.J. Schmid, Input-output measures for model reduction and closed-loop control: application to global modes, J. Fluid Mech. 685 (2011) 23–53.

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

[11] T. Lassila, G. Rozza, Parametric free-form shape design with PDE models and reduced basis method, Comput. Methods Appl. Mech. Engrg. 199 (23–24) (2010) 1583–1592.

[12] I. Oliveira, A. Patera, Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations, Optim. Eng. 8 (1) (2007) 43–65.

[13] K. Carlberg, C. Farhat, A compact proper orthogonal decomposition basis for optimization-oriented reduced-order models, in: Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2008.

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

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

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

[17] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1993) 539–575.

[18] J.L. Lumley, The structure of inhomogeneous turbulent flows, in: Atmospheric Turbulence and Radio Wave Propagation, 1967, pp. 166–178.

[19] W. Cazemier, R.W.C.P. Verstappen, A.E.P. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Phys. Fluids 10 (1998) 1685–1699.

[20] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2003) 492–515.

[21] J. Weller, E. Lombardi, M. Bergmann, A. Iollo, Numerical methods for low-order modeling of fluid flows based on POD, Internat. J. Numer. Methods Fluids 63 (2) (2010) 249–268.

[22] 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.

[23] T. Iliescu, Z. Wang, Variational multiscale proper orthogonal decomposition: Navier-Stokes equations, Numer. Methods Partial Differential Equations 30 (2014) 641–663.

[24] R. Eymard, T. Gallouët, R. Herbin, The finite volume method, in: Ph. Ciarlet, J.L. Lions (Eds.), Handbook for Numerical Analysis, North Holland, 2000, pp. 715–1022.

[25] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics, the Finite Volume Method, Pearson Education Limited, 2007.

[26] C.A.J. Fletcher, Computational Techniques for Fluid Dynamics, Vol. 1: Fundamental and General Techniques, Springer, 1996.

[27] G. Iaccarino, Predictions of a turbulent separated flow using commercial CFD codes, J. Fluids Eng. 123 (2001) 819–828.

[28] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (1998) 620–631.

[29] B. Haasdonk, M. Ohlberger, Adaptive basis enrichment for the reduced basis method applied to finite volume schemes, in: Proceedings of 5th Internatonal Symposum on Finite Volumes for Complex Applications, 2008.

[30] M. Drohmann, B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximation of evolution equations on parametrized geometries, in: Proceedings of Algoritmy 2008.

[31] B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, ESAIM-Math. Model. Numer. 42 (2008) 277–302.

[32] B. Haasdonk, M. Ohlberger, Reduced basis method for explicit finite volume approximations of nonlinear conservation laws, in: Proceedings of Symposia in Applied Mathematics, 2009.

[33] B. Haasdonk, M. Ohlberger, G. Rozza, A reduced basis method for evolution schemes with parameter-dependent explicit operators, Electron. Trans. Numer. Anal. 32 (2008) 145–161.

[34] J. Östh, B.R. Noack, S. Krajnovi¢, 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) 515–544.

[35] J. Rambo, Reduced-order modeling of multiscale turbulent convection: application to data center thermal management (Ph.D. thesis), Georgia Institute of Technology, 2006.

[36] J. Rambo, Y. Joshi, Reduced-order modeling of turbulent forced convection with parametric conditions, Int. J. Heat Mass Transfer 50 (2007) 539–551.

[37] J. Rambo, Y. Joshi, Reduced order modeling of steady turbulent flows using the POD, in: Proceedings of the ASME Summer Heat Transfer Conference, Vol. 3, 2005, pp. 837–846.

[38] S.B. Pope, Turbulent Flows, Cambridge University Press, 2000.

[39] 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.

[40] O. San, T. Iliescu, Proper orthogonal decomposition closure models for fluid flows: Burgers equation, Int. J. Numer. Anal. Model. B 5 (3)(2013) 217–237.

[41] 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.

[42] L. Cordier, B.R. Noack, G. Tissot, G. Lehnasch, J. Delville, M. Balajewicz, G. Daviller, R.K. Niven, Identification strategies for model-based control, Exp. Fluids 54 (2013) 1580.

[43] M. Balajewicz, E.H. Dowell, Stabilization of projection-based reduced order models of the Navier–Stokes, Nonlinear Dynam. 70 (2012) 1619–1632.

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

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

[46] 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.

[47] L.F. Richardson, Weather Prediction by Numerical Process, Cambridge University Press, 1922.

[48] A.N. Kolmogorov, The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers, Proc. USSR Acad. Sci. 32 (1941) 16–18. (in Russian). Reprinted in English: Proceedings of the Royal Society A 434 (1991) 9–13.

[49] OpenFOAM, Version 2.3, 2014. www.openfoam.org.

[50] U. Ghia, K.N. Ghia, C.T. Shin, High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

[51] O. Botella, R. Peyret, Benchmark spectral results on the Lid-driven cavity flow, Comput. & Fluids 27 (4) (1998) 421–433.

[52] C.H. Bruneau, M. Saad, The 2D lid-driven cavity problem revisited, Comput. & Fluids 35 (2006) 326–348.

[53] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, A reduced order model for investigating the dynamics of the Gen-IV LFR coolant pool, Appl. Math. Model. (2016) (revised manuscript submitted for publication).

[54] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2) (2013) 492–515.

[55] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.

[56] B. Noack, P. Papas, P. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, J. Fluid Mech. 523 (1) (2005) 339–365.

[57] H. Jasak, Error analysis and estimation in the Finite Volume method with applications to fluid flows (Ph.D. thesis), Imperial College, University of London, 1996.

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

[59] S. Sirisup, G.E. Karniadakis, Stability and accuracy of periodic flow solutions obtained by a POD-penalty method, Physica D 202 (2005) 218–237.

[60] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM-CMBS, Philadelphia, 1977.

[61] K. Bizon, G. Continillo, Reduced order modelling of chemical reactors with recycle by means of POD-penalty method, Comput. Chem. Eng. 39 (2012) 22–32.

[62] D. Rempfer, F.H. Fasel, Evolution of three-dimensional coherent structures in a flat-plate boundary-layer, J. Fluid Mech. 260 (1994) 351–375.

[63] E. Erturk, Discussions on driven cavity flow, Internat. J. Numer. Methods Fluids 60 (2009) 275–294.

[64] H. Jasak, A. Jemcov, Z. Tukovic, OpenFOAM: A C++ library for complex physics simulations, in: Proceedings of the International Workshop on Coupled Methods in Numerical Dynamics, 2007.

[65] P. Fritzson, Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, Wiley-IEEE Press, New York, NY, 2004.

[66] The Modelica Association. Modelica 3.2.2 Language Specification, 2014. Available online at: http://www.modelica.org/.

[67] H. Elmqvist, F.E. Cellier, M. Otter, Object-oriented modeling of hybrid systems, in: Proceedings of the European Simulation Symposium, 1993.

[68] DYMOLA, 2015. Dassault Systèmes, France, 2015. http://www.3ds.com/products-services/catia/products/dymola.

[69] L.R. Petzold, A description of DASSL: A differential/algebraic system solver, in: Proceedings of the IMACS World Congress, 1982.

[70] F.R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (8) (1994) 1598–1605.

[71] F.R. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, in: Proceeding of the 4th International Symposium on Turbulence, Heat and Mass Transfer, 2003.

[72] 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 (5) (2015) 1136–1161.