Computers and Fluids 246 (2022) 105604

Contents lists available at ScienceDirect

Computers and Fluids

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

Stabilized reduced-order models for unsteady incompressible flows in three-dimensional parametrized domains

Stefano Buoso [a,b,*], Andrea Manzoni [c], Hatem Alkadhi [b], Vartan Kurtcuoglu [a,d,e]

[a] The Interface Group, Institute of Physiology, University of Zurich, Zurich, Switzerland
[b] Institute of Diagnostic and Interventional Radiology, University Hospital Zurich, University of Zurich, Zurich, Switzerland
[c] MOX, Dipartimento di Matematica, Politecnico di Milano, Milano, Italy
[d] National Center of Competence in Research, Kidney.CH, Switzerland
[e] Zurich Center for Integrative Human Physiology, University of Zurich, Zurich, Switzerland

ARTICLE INFO

Keywords:
Computational fluid dynamics
Finite-elements
Proper orthogonal decomposition
Reduced order modeling
Stabilization techniques
Discrete empirical interpolation

ABSTRACT

In this work we derive a parametric reduced-order model (ROM) for the unsteady three-dimensional incompressible Navier–Stokes equations without additional pre-processing on the reduced-order subspaces. Concerning the high-fidelity, full-order model, we start from a streamline-upwind Petrov–Galerkin stabilized finite element discretization of the equations using $\mathbb{P}^1 - \mathbb{P}^1$ elements for velocity and pressure, respectively. We rely on Galerkin projection of the discretized equations onto reduced basis subspaces for the velocity and the pressure, respectively, obtained through Proper Orthogonal Decomposition on a dataset of snapshots of the full-order model. Both nonlinear and nonaffinely parametrized algebraic operators of the reduced-order system of nonlinear equations, including the projection of the stabilization terms, are efficiently assembled exploiting the Discrete Empirical Interpolation Method (DEIM), and its matrix version (MDEIM), thus obtaining an efficient *offline–online* computational splitting. We apply the proposed method to *(i)* a two-dimensional lid-driven cavity flow problem, considering the Reynolds number as parameter, and *(ii)* a three-dimensional pulsatile flow in stenotic vessels characterized by geometric and physiological parameter variations. We numerically show that the projection of the stabilization terms on the reduced basis subspace and their reconstruction using (M)DEIM allows to obtain a stable ROM with coupled velocity and pressure solutions, without any need for enriching the reduced velocity space, or further stabilizing the ROM. Additionally, we demonstrate that our implementation allows to compute the ROM solution about 20 times faster than the full order model.

1. Introduction

Haemodynamic models have the potential to provide essential information for the functional evaluation of clinical cases, offering useful parameters and biomarkers to support clinicians in diagnostic procedures and surgical planning [1]. The US Food and Drug administration has also recognized the benefits of complementing experiments with in-silico physiological models for the design of medical devices [2]. Such models are often based on the spatio-temporal discretization of physics-based models, usually expressed in terms of unsteady nonlinear partial differential equations (PDEs), for which the domain shape, the boundary/initial conditions and the model data (coefficients, source terms, etc.) are related to subject-specific anatomy and physiology. Computational requirements for the numerical simulation of these models are often out of reach in any practical clinical application where they are required to provide results quickly for a variety of scenarios [3]. A prime example is the patient-specific assessment of the pressure losses in lesioned vessels [4], which is currently done invasively, but could be replaces by a computational assessment. However, for such in-silico models to be adopted, their computational cost must be reduced.

Recently, several research efforts have been aimed at making in-silico models affordable in practical situations, in particular for haemodynamics simulations. For instance, a recent trend in the field of scientific machine learning consists of training neural networks to identify surrogate models to reproduce the input–output relationship of the systems of interest [5,6]. In this context, [7] have trained a network for the real-time estimation of the Fractional Flow Reserve (FFR) index, which is a clinical metric used to evaluate the pressure losses due to stenoses in patients' coronary arteries. Physics-informed neural networks have been exploited to assimilate flow measurements and computational flow dynamics models derived from first physical principles [8,9], or reconstruct high-resolution flow fields from limited flow information [10]. Deep learning (DL) algorithms have

* Corresponding author at: The Interface Group, Institute of Physiology, University of Zurich, Zurich, Switzerland.
E-mail address: stefano.buoso@uzh.ch (S. Buoso).

https://doi.org/10.1016/j.compfluid.2022.105604
Received 5 September 2021; Received in revised form 21 June 2022; Accepted 17 July 2022
Available online 21 July 2022

been recently exploited to build non-intrusive and extremely efficient ROMs for parameter-dependent unsteady problems in computational fluid dynamics [11–13] and continuum mechanics [14], ultimately enabling real-time fluid simulations in a parametrized context. The majority of these strategies do not rely on the solution of a system of equations arising from the discretization of mathematical models based on physical principles, and they are, to a certain extent, agnostic with regards to the physical principles driving system evolution,

A different class of approaches starts directly from the mathematical description of these principles to obtain a cheaper model for the evaluation of the fluid dynamics equations. These are known as projection-based ROMs, built, e.g., through the reduced basis (RB) method [15]. Essentially, they consist of projecting the full-order model (FOM) corresponding to a given, high-fidelity discretization of the PDE system, onto a low-dimensional subspace. This subspace is usually spanned by a set of basis functions built from a family of snapshots of the FOM, obtained for different input parameter values, at different times. The main advantage of ROMs is that, once trained *offline*, they can provide *online*, for any new unseen scenario, a reliable approximation of the solution field, thanks to the use of a (Petrov-)Galerkin projection onto low-dimensional subspaces. Our contribution addresses a possible way to build projection-based ROMs for the unsteady Navier–Stokes equations in domains with varying shape, focusing on a suitable strategy to ensure the stability of the ROM.

1.1. Existing strategies

RB methods have been applied to fluid dynamics problems starting mainly from FOMs based on the Finite Element (FE) Method to generate snapshots, and using strategies such as Proper Orthogonal Decomposition (POD) [16–23] or greedy algorithms [24–27]. Only few contributions investigated the use of different types of FOMs such as, e.g., ones based on the finite volume (FV) method [4,28–30]. The main focus of these works has been on incompressible fluids described by the Navier–Stokes (NS) equations. However, two relevant issues arising when dealing with fluid flows are related to the need for *(i)* preserving the pressure–velocity coupling at the ROM level, by relying on a mixed formulation, and *(ii)* fulfilling some stability constraints (regarding both the violation of the inf–sup condition and dominating convection).

Recently, we have proposed a RB framework based on the FV method that is able to describe systems with arbitrary parametrizations of geometry and flow conditions [4]. The pressure–velocity coupling is preserved by the exact projection of the coupling strategy adopted in the full order model (the pressure-implicit with splitting of operators (PISO) algorithm). An interesting feature of that framework is that no additional stabilization terms are required at the reduced level to ensure its stability, provided that the reduction is consistent with the exact formulation used in the original model.

In the case of high-fidelity FOMs built through the FE method, ensuring the stability of the ROM – in the sense of avoiding spurious pressure modes – when dealing with parametrized incompressible Navier–Stokes equations is not a trivial task, for which several approaches have been proposed in the past decade. Indeed, even though the velocity basis functions are obtained through a stable FOM, its Galerkin projection onto reduced spaces made by (stable) velocity and pressure fields computed for sampled values of the parameters does not ensure that the resulting ROM preserves the same stability properties. For instance:

- The velocity space can be augmented through a set of enriching basis functions, obtained by mapping the pressure POD modes through the so-called pressure supremizing operator, which depends on the divergence term. This yields a RB problem with additional degrees of freedom for the velocity field (as many as the pressure variable). To avoid the construction of the pressure supremizing operator online, for each parameter value, an offline enrichment (or approximate supremizer option) is usually performed [22,31].
- A Petrov–Galerkin RB method can be used to build an automatically stable RB problem, relying, for instance, on the least squares (LS) method. In this case, the resulting LS-RB method uses a test space that is obtained as the image of the trial RB space through a global supremizing operator involving both velocity and pressure fields, yielding an automatically stable RB problem [32,33].
- Pressure can be treated independently from velocity, reconstructing it by solving a Poisson equation. This strategy would yield a stable ROM, however, requiring divergence-free velocity basis functions [34]. Similarly, only the momentum equation could be used online, assigning the same temporal coefficients to both velocity and pressure in the ROM [18,29], although this approach seems to be valid only for simplified, parameter-independent geometries, and mild parametric variations.
- A stabilized FOM (like, e.g., a $\mathbb{P}^1$ - $\mathbb{P}^1$ Streamline Upwind Petrov–Galerkin (SUPG) FE discretization) instead of a stable FOM (like, e.g., a $\mathbb{P}^2$ - $\mathbb{P}^1$ FE discretization). In this case, neither an enrichment of the velocity space nor a LS-RB formulation is necessary to ensure the stability of the corresponding RB system, due to the fact that the additional terms stemming from the stabilization provide a contribution to the pressure/divergence equation, also at the reduced level, preventing the ROM system from becoming nonsingular. This is the option we investigate in this paper.

A second issue, dealing with the possible stability loss due to higher Reynolds numbers, has been instead taken into account to address turbulent flows through projection-based ROMs and avoid long-time instabilities. In this respect, several strategies have been proposed, including, e.g., dynamic subgrid-scale and variational multiscale models [35,36], constrained Galerkin projections [37] or minimum residual formulations [32,38,39] and, more recently, closure models using data-driven strategies [40,41] and neural networks [42]. We do not address this aspect in our work, since we only consider laminar flows, and are interested in the short- rather than the long-time flow regime.

1.2. Contribution and novelty of this paper

In this work we aim to show that, starting from a stabilized formulation of the three-dimensional NS equations, a canonical Galerkin projection of the equations onto velocity/pressure reduced spaces, spanned by POD modes, yields a ROM which is both inf–sup stable and able to preserve the pressure–velocity coupling. In particular, we numerically show that if the original FOM has been stabilized (e.g., by means of a $\mathbb{P}^1 - \mathbb{P}^1$ discretization of the NS equations with streamline upwind Petrov–Galerkin (SUPG) stabilization), we do not need special treatments of the ROM to ensure stability, thus avoiding the supremizer enrichment of the velocity space. We also show that this remains true even if the three-dimensional fluid domain undergoes parametric changes, and the flow involves parameter–dependent time-varying boundary conditions. Indeed, the construction of ROMs for the Navier–Stokes equations relying on a stabilized FOM had been formerly addressed in [43,44], however, *(i)* without taking into account either the parameter-dependent case, or a mixed formulation to ensure the pressure–velocity coupling in [43], and without *(ii)* considering parametrized geometries (and suitable hyper-reduction techniques to manage them) in [44].

We remark that the SUPG stabilized formulation we rely on derives from a multiscale splitting of the spaces for the velocity and pressure unknowns into the resolved scales (i.e., the FE part of the solution) and the unresolved subscales. Hence, stabilization terms are inserted into the equation for the resolved scales to account for the contribution of the unresolved ones, and depend on the residual, thus ending up with a residual-based, consistent stabilization. However, stabilization terms are highly nonlinear due to the presence of stabilization parameters, and are in our case also parameter-dependent because of the need to deal with parametrized geometries. Therefore, special attention must

be paid to the efficient treatment of stabilization terms through hyper-reduction, since their poor approximation might severely impact both the stability and the accuracy of the ROM. We remark a variational multiscale (VMS) formulation as the one employed in Stabile et al. [36] shares several features with the employed SUPG strategy. Indeed, in Stabile et al. [36], a projection of the VMS formulation of the two-dimensional Navier–Stokes equation onto a POD subspace for velocity and pressure has been considered, highlighting that retaining the VMS stabilization terms at the ROM level provides a stable ROM. However, their investigation did not consider three-dimensional problems, time-dependent boundary conditions, or geometric variations of the domain. In this latter case, the authors suggested that a supremizer enrichment might be necessary even when VMS terms are considered at ROM level: our findings, although obtained for a simpler SUPG scheme, show that a supremizer enrichment is not necessary.

To provide the most general formulation possible, we rely on the Discrete Empirical Interpolation Method (DEIM) [45–47], and its matrix version (MDEIM) [48] to approximate the dependence of the algebraic operators of the NS equations on the input parameters describing the problem. This approach has been already introduced by the authors in the context of FV [4] and will be here adapted for the FEM. We aim at showing that, even with variations in geometry and boundary conditions, ROM stability is granted simply by projecting the stabilized FOM onto the reduced spaces spanned by the velocity/pressure POD basis functions, without any supremizer enrichment, and that hyper-reduction only impacts the accuracy of the ROM solution, but not its stability. Importantly, we also aim to show that this approach allows to obtain considerable computational speedup when compared to the reference FEM formulation.

2. Methods

2.1. Problem description: parametrized unsteady Navier–Stokes equations

For the sake of testing, we consider in this paper two different problems: a two-dimensional, unsteady, lid-driven cavity flow, depending on the Reynolds number as parameter, and an unsteady, three-dimensional pulsatile flow into a cylindrical conduit with a prescribed section restriction, which shall be representative of a lesioned branch of the human coronary artery tree with respect to geometry and flow conditions. In this latter case, we restrict the variability of the problem to two parameters: a geometrical one (stenosis severity) and a physical one (inlet velocity profile). However, as we have shown previously, the approach can be easily generalized to an arbitrary set of parameters [4]. Since this latter test case is the more general one, involving three-dimensional flows, geometrical parameters and time-dependent boundary conditions, we frame the general problem description on it. Numerical results dealing with the three-dimensional pulsatile flow are reported in Section 3, where a thorough numerical analysis has been carried out on this test case. A more compact analysis of the lid-driven cavity flow is provided in the Appendix.

2.1.1. Formulation

In the case of the three-dimensional pulsatile flow benchmark, the reference domain in our work, $\Omega_0 \subset \mathbb{R}^d$, $d = 3$, is a straight pipe with diameter and length d_0 and L_0, respectively. We define a Cartesian coordinate system with origin at the center of the inlet section and x_3 axis along the pipe length direction. The fluid domain is meshed with approximatively 68,000 tetrahedral elements. We prescribe a bell-shaped section restriction along x_3 on the reference geometry axis using the mapping function

$$x_i = x_{i,0}\left[1 - \mu_g \exp\left(-\frac{\left(x_3 - c_1\right)^2}{c_2}\right)\right] \qquad \text{and} \qquad i = 1, 2, \tag{1}$$

where x_i are the coordinates of a mesh point in the deformed configuration, $x_{i,0}$ are the coordinates of the corresponding mesh point in the reference configuration, μ_g, is the amplitude of the bell-shaped restriction and c_1 and c_2 are two constant parameters defining the throat position along x_3 and the length of the stenosis, respectively. We denote the domain with the new coordinates by $\Omega\left(\boldsymbol{\mu}_g\right)$.

Let $\Omega\left(\boldsymbol{\mu}_g\right) \subset \mathbb{R}^d$, $d = 3$, be an open bounded domain with (piecewise smooth) boundary $\partial\Omega\left(\boldsymbol{\mu}_g\right)$. The latter can be split into a Dirichlet boundary $\Gamma_D\left(\boldsymbol{\mu}_g\right)$ and a Neumann boundary $\Gamma_N\left(\boldsymbol{\mu}_g\right)$, respectively, such that $\partial\Omega\left(\boldsymbol{\mu}_g\right) = \Gamma_D\left(\boldsymbol{\mu}_g\right) \cup \Gamma_N\left(\boldsymbol{\mu}_g\right)$. In particular, the Dirichlet boundary can be decomposed into the union of the inlet boundary $\Gamma_{in}\left(\boldsymbol{\mu}_g\right)$ and the wall boundary $\Gamma_w\left(\boldsymbol{\mu}_g\right)$, such that $\Gamma_D\left(\boldsymbol{\mu}_g\right) = \Gamma_{in}\left(\boldsymbol{\mu}_g\right) \cup \Gamma_w\left(\boldsymbol{\mu}_g\right)$.

At this point, we can set the unsteady, incompressible parametrized NS equations to model the flow in the cylinder. The fluid is assumed to be Newtonian with constant and uniform density ρ, and kinematic viscosity ν. Under a *rigid wall* assumption, the system of parametrized PDEs we focus on reads as follows:

$$\begin{cases} \dfrac{\partial}{\partial t}\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{0} & \text{in } \Omega\left(\boldsymbol{\mu}_g\right)\times(0,T), \\ \nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega\left(\boldsymbol{\mu}_g\right)\times(0,T), \\ \boldsymbol{u} = \boldsymbol{g}_D\left(\boldsymbol{\mu}_p\right) & \text{in } \Gamma_D\left(\boldsymbol{\mu}_g\right)\times(0,T), \\ -p\boldsymbol{n} + \nu(\nabla\boldsymbol{u})\boldsymbol{n} = \boldsymbol{0} & \text{in } \Gamma_N\left(\boldsymbol{\mu}_g\right)\times(0,T), \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{in } \Omega\left(\boldsymbol{\mu}_g\right)\times\{t=0\}, \end{cases} \tag{2}$$

where $\boldsymbol{u}$ denotes the fluid velocity, p the fluid pressure (divided by the fluid density), $\boldsymbol{n}$ the boundary normal vector,

$$\boldsymbol{g}_D\left(\boldsymbol{\mu}_p\right) = \begin{cases} \boldsymbol{u}_{in}\left(\boldsymbol{\mu}_p\right) & \text{on } \Gamma_{in}\left(\boldsymbol{\mu}_g\right)\times(0,T), \\ \boldsymbol{0} & \text{in } \Gamma_w\left(\boldsymbol{\mu}_g\right)\times(0,T), \end{cases}$$

and $\boldsymbol{u}_{in}\left(\boldsymbol{\mu}_p\right)$ the Dirichlet data imposed at the inlet (Γ_D). On the lateral wall of the cylinder (Γ_w), we impose a no-slip condition, whereas a free-stress (homogeneous Neumann) boundary condition is imposed at the outlet (Γ_N).

For the sake of clarity, we have highlighted the dependence of the computational domain on the geometric parameters, $\boldsymbol{\mu}_g$, and of the boundary conditions on physical parameters, $\boldsymbol{\mu}_p$. Hereinafter, unless specifically stated, $\boldsymbol{\mu}$ will refer to the generic vector of input parameters, $\boldsymbol{\mu} = (\boldsymbol{\mu}_p, \boldsymbol{\mu}_g)$, belonging to the parameter space $\mathcal{P} \subset \mathbb{R}^p$.

In particular, at the inlet of our domain, we prescribe an oscillatory parabolic velocity profile $\boldsymbol{u}_{in} = \left(u_{1_{in}}, u_{2_{in}}, u_{3_{in}}\right)$, which has a non-zero component only in the x_3 direction, given by

$$u_{3_{in}}\left(\boldsymbol{\mu}_p\right) = 2\mu_p\left[1.0 + 0.25\sin(2\pi f t)\right]\left(1 - 4\frac{\left(x_1^2 + x_2^2\right)}{d_0^2}\right)\Big|_{x_3=0}. \tag{3}$$

Here x_1, x_2 and x_3 are the coordinates of a point of the deformed domain, d_0 is the inlet diameter, μ_p is the mean section inlet velocity, f is the oscillation frequency and t is the physical time. We will consider here a single parameter defining the inlet velocity profile. In case the inlet section is parameter-dependent, further geometrical parameters can be included in the boundary data definition, see, e.g., [4].

Let us now consider the weak formulation of the problem in (2), which provides the basis for the subsequent generation of the ROM. Let us introduce the following functional space:

$$V_D = \{\mathbf{v} \in H^1(\Omega)^d : \boldsymbol{v}|_{\Gamma_D} = \boldsymbol{g}_D\},$$

$$V = \{\mathbf{v} \in H^1(\Omega)^d : \boldsymbol{v}|_{\Gamma_D} = \boldsymbol{0}\},$$

$$Q = L^2(\Omega).$$

The weak formulation of (2) reads as follows: find $(\boldsymbol{u}(t), p(t)) \in V_D \times Q$ such that, for all $t \in (0,T)$,

$$\left(\frac{\partial}{\partial t}\boldsymbol{u}, \boldsymbol{v}\right) + (\boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}) + (\nu\nabla\boldsymbol{u}, \nabla\boldsymbol{v}) - (p, \nabla\cdot\boldsymbol{v}) + (\nabla\cdot\boldsymbol{u}, q) = 0 \tag{4}$$

for all $(\boldsymbol{v}, q) \in V \times Q$, with $\mathbf{u}(0) = \boldsymbol{u}_0$. In the equation above, $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product.

2.2. Full order model

We rely on the finite element method to build the FOM. Hence, we consider a FE partition, $\mathcal{T}_h$, of the domain $\Omega(\boldsymbol{\mu}_g)$, which we use to construct conforming finite element spaces $V_h \subset V$ and $Q_h \subset Q$ for velocity and pressure, respectively. Common choices for FE spaces are $\mathbb{P}^{k+1} - \mathbb{P}^k$, $k \geq 1$, in order to satisfy the inf–sup stability condition

$$\inf_{q_h \in Q_h} \sup_{\boldsymbol{w}_h \in V_h} \frac{(q_h, \nabla \cdot \boldsymbol{w}_h)}{\|\boldsymbol{w}_h\|_V \|q_h\|_Q} \geq \bar{\beta} > 0.$$

Approximation spaces such as $\mathbb{P}^2 - \mathbb{P}^1$ satisfy the condition above. However, they might entail overwhelming computational costs in the case of fine meshes because of the large number of degrees of freedom in the velocity space. An attractive alternative to mitigate the computational effort is provided by low-order approximation spaces such as $\mathbb{P}^1 - \mathbb{P}^1$ spaces, although they do not satisfy the inf–sup condition [49]. To overcome this issue, we rely on the streamline upwind Petrov–Galerkin (SUPG) stabilization [50] – formulated as in the Variational Multiscale framework [51] – that allows us not only to control the incompressibility constraint, but also to enhance stability with respect to the convective terms in the case of advection dominated flows. This approach also allows us to model the smaller flow scales not resolved by the mesh. This is comparable to sub-grid models in Large Eddy Simulation approaches [52].

2.2.1. Space discretization

Hence, we first introduce the FE space

$$X_h^r = \{w_h \in C^0(\bar{\Omega}) \ : w_h|_K \in \mathbb{P}^r \ \forall K \in \mathcal{T}_h\}$$

and define $V_h = V \cap [X_h^r]^d$, $V_{D,h} = V_D \cap [X_h^r]^d$, and $Q_h = Q \cap X_h^r$. We also introduce the strong residuals $\boldsymbol{r}_M(\boldsymbol{v}_h, p_h)$ and $\boldsymbol{r}_C(\boldsymbol{v}_h)$ of the momentum and conservation equations, respectively:

$$\boldsymbol{r}_M(\boldsymbol{v}_h, q_h) = \frac{\partial}{\partial t}\boldsymbol{v}_h + (\boldsymbol{v}_h \cdot \nabla)\boldsymbol{v}_h - \nu \Delta \boldsymbol{v}_h + \nabla q_h, \tag{5}$$

$$\boldsymbol{r}_C(\boldsymbol{v}_h) = \nabla \cdot \boldsymbol{v}_h. \tag{6}$$

The semi-discrete SUPG formulation of (4) becomes: for all $t \in (0, T)$, find $(\boldsymbol{u}_h(t), p_h(t)) \in V_{D,h} \times Q_h$ such that

$$\left(\frac{\partial}{\partial t}\boldsymbol{u}_h, \boldsymbol{v}_h\right) + (\boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h, \boldsymbol{v}_h) + (\nu \nabla \boldsymbol{u}_h, \nabla \boldsymbol{v}_h) - (p_h, \nabla \cdot \boldsymbol{v}_h) + (\nabla \cdot \boldsymbol{u}_h, q_h) + \sum_{K \in \mathcal{T}_h} \left(\tau_M \boldsymbol{r}_M(\boldsymbol{u}_h, p_h), \boldsymbol{u}_h \cdot \nabla \boldsymbol{v}_h + \nabla q_h\right)_K + \sum_{K \in \mathcal{T}_h} \left(\tau_c \boldsymbol{r}_c(\boldsymbol{u}_h), \nabla \cdot \boldsymbol{v}_h\right)_K = \boldsymbol{0} \tag{7}$$

for all $(\boldsymbol{v}_h, q_h) \in V_h \times Q_h$, with $\mathbf{u}_h(0) = \boldsymbol{u}_0$. Here, $\tau_M = \tau_M(\boldsymbol{\mu}_g, \boldsymbol{u}_h)$ and $\tau_C = \tau_C(\boldsymbol{\mu}_g, \boldsymbol{u}_h)$ are two stabilization parameters, which are calculated element-wise [51]. Several models have been proposed for the calculation of these parameters. In our work, their element-wise value is computed using the average element size, h, [53]. The expressions for τ_M and τ_C are thus [54]

$$\tau_M = \left(\frac{4}{\Delta t^2} + \frac{4\|\boldsymbol{u}_h\|^2}{h^2} + 9\left(\frac{2\mu}{\rho h^2}\right)^2\right)^{-\frac{1}{2}}, \qquad \tau_C = \frac{h^2}{32\tau_M}, \tag{8}$$

where Δt is the time step that is chosen for the time discretization, and $\|\boldsymbol{u}_h\|$ denotes the velocity magnitude.

2.2.2. Time discretization

Let us now introduce the time discretization of problem (7). To this end, we consider a first-order backward Euler scheme with semi-implicit treatment of the convective terms. By linearizing the convective term, we can mitigate the computational cost of a fully implicit scheme. Hence, we partition the time interval $[0, T]$ into N_t subintervals of size $\Delta t = T/N_t$, denoting by $t_n = n\Delta t$, $n = 0, \dots, N_t$ the instances of the time discretization, and by $\boldsymbol{u}_h^n$, p_h^n the approximations of $\boldsymbol{u}_h(t^n)$ and $p_h(t^n)$, respectively.

The fully discrete semi-implicit SUPG approximation of the unsteady Navier–Stokes equations thus reads: given $\boldsymbol{u}_h^n$, find $(\boldsymbol{u}_h^{n+1}, p_h^{n+1}) \in V_h \times Q_h$ such that

$$\begin{aligned}\left(\frac{\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n}{\Delta t}, \boldsymbol{v}_h\right) &+ (\boldsymbol{u}_h^n \cdot \nabla \boldsymbol{u}_h^{n+1}, \boldsymbol{v}_h) + (\nu \nabla \boldsymbol{u}_h^{n+1}, \nabla \boldsymbol{v}_h) - (p_h^{n+1}, \nabla \cdot \boldsymbol{v}_h) \\ &+ (\nabla \cdot \boldsymbol{u}_h, q_h) \\ &+ \sum_{K \in \mathcal{T}_h} \left(\tau_M^n \tilde{\boldsymbol{r}}_M(\boldsymbol{u}_h^{n+1}, p_h^{n+1}), \boldsymbol{u}_h^n \cdot \nabla \boldsymbol{v}_h + \nabla q_h\right)_K \\ &+ \sum_{K \in \mathcal{T}_h} \left(\tau_c^n \boldsymbol{r}_c(\boldsymbol{u}_h^{n+1}), \nabla \cdot \boldsymbol{v}_h\right)_K = G_u(\boldsymbol{v}_h; t^{n+1}) \\ &+ G_p(q_h; t^{n+1})\end{aligned} \tag{9}$$

for all $(\boldsymbol{v}_h, q_h) \in V_h \times Q_h$, with $\mathbf{u}_h(0) = \boldsymbol{u}_0$ where

$$\tilde{\boldsymbol{r}}_M(\boldsymbol{u}_h^{n+1}, p_h^{n+1}) = \frac{\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n}{\Delta t} + (\boldsymbol{u}_h^n \cdot \nabla)\boldsymbol{u}_h^{n+1} - \nu \Delta \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1},$$

and $G_u(\boldsymbol{v}_h; t^{n+1})$, $G_p(q_h; t^{n+1})$ are the contributions of the non-homogeneous Dirichlet boundary conditions on Γ_D at time $t = t^{n+1}$ on both equations (see, e.g., [23] for further details). Moreover,

$$\tau_M^n = \left(\frac{4}{\Delta t^2} + \frac{4\|\boldsymbol{u}_h^n\|^2}{h^2} + 9\left(\frac{2\mu}{\rho h^2}\right)^2\right)^{-\frac{1}{2}}, \qquad \tau_C^n = \frac{h^2}{32\tau_M^n}. \tag{10}$$

2.2.3. Algebraic formulation

Problem (9) can be equivalently rewritten in algebraic form. Let us denote by $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_{h,u}}$ and $\{\eta_k\}_{k=1}^{N_{h,p}}$ two (Lagrangian) finite element bases for V_h and Q_h, respectively, and by $\mathbf{u}_h^n \in \mathbb{R}^{N_{h,u}}$, $\mathbf{p}_h^n \in \mathbb{R}^{N_{h,p}}$ the coefficient vectors in the expansions of the discrete functions $\boldsymbol{v}_h^n$ and p_h^n onto the FE bases. The algebraic form of problem (9) reads: given $\mathbf{u}_h^n$, for $n = 0, \dots, N_t - 1$, find $(\mathbf{u}_h^{n+1}, \mathbf{p}_h^{n+1}) \in \mathbb{R}^{N_{h,u}} \times \mathbb{R}^{N_{h,p}}$ such that

$$\begin{bmatrix} \frac{1}{\Delta t}\boldsymbol{M}_h + \boldsymbol{C}_h(\mathbf{u}_h^n) + \boldsymbol{L}_h & \boldsymbol{B}_h^T \\ -\boldsymbol{B}_h & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_h^{n+1} \\ \mathbf{p}_h^{n+1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{S}_{uu,h}(\mathbf{u}_h^n) & \boldsymbol{S}_{up,h}(\mathbf{u}_h^n) \\ \boldsymbol{S}_{pu,h}(\mathbf{u}_h^n) & \boldsymbol{S}_{pp,h}(\mathbf{u}_h^n) \end{bmatrix} \begin{bmatrix} \mathbf{u}_h^{n+1} \\ \mathbf{p}_h^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{u,h}(\mathbf{u}_h^n) \\ \boldsymbol{f}_{p,h}(\mathbf{u}_h^n) \end{bmatrix} \tag{11}$$

where:

- $\boldsymbol{M}_h \in \mathbb{R}^{N_{h,u} \times N_{h,u}}$ and $\boldsymbol{L}_h \in \mathbb{R}^{N_{h,u} \times N_{h,u}}$ are the velocity mass and stiffness matrices, respectively,

  $$(\boldsymbol{M}_h)_{ij} = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i), \qquad (\boldsymbol{K}_h)_{ij} = \nu(\nabla \boldsymbol{\varphi}_j, \nabla \boldsymbol{\varphi}_i), \qquad i, j = 1, \dots, \mathbb{R}^{N_{h,u}};$$

- $\boldsymbol{B}_h \in \mathbb{R}^{N_{h,p} \times N_{h,p}}$ is the pressure-divergence matrix,

  $$(\boldsymbol{B}_h)_{ij} = (\nabla \cdot \boldsymbol{\varphi}_j, \varphi_i), \qquad i = 1, \dots, \mathbb{R}^{N_{h,p}}, j = 1, \dots, \mathbb{R}^{N_{h,u}};$$

- $\boldsymbol{C}_h(\mathbf{u}_h^n) \in \mathbb{R}^{N_{h,u} \times N_{h,u}}$ is the matrix corresponding to the nonlinear convective term,

  $$(\boldsymbol{C}_h(\mathbf{u}_h^n))_{ij} = (\mathbf{u}_h^n \cdot \nabla \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i), \qquad i, j = 1, \dots, \mathbb{R}^{N_{h,u}};$$

- the blocks of the matrix corresponding to the SUPG stabilization term are given by

  $$(\boldsymbol{S}_{uu,h}(\mathbf{u}_h^n))_{ij} = \left(\tau_M^n\left(\frac{1}{\Delta t}\boldsymbol{\varphi}_j + \mathbf{u}_h^n \cdot \nabla \boldsymbol{\varphi}_j - \nu \Delta \boldsymbol{\varphi}_j\right), \mathbf{u}_h^n \cdot \nabla \boldsymbol{\varphi}_i\right) + \left(\tau_C^n \nabla \cdot \boldsymbol{\varphi}_j, \nabla \cdot \boldsymbol{\varphi}_i\right),$$

  $$(\boldsymbol{S}_{pu,h}(\mathbf{u}_h^n))_{kj} = \left(\tau_M^n\left(\frac{1}{\Delta t}\boldsymbol{\varphi}_j + \mathbf{u}_h^n \cdot \nabla \boldsymbol{\varphi}_j - \nu \Delta \boldsymbol{\varphi}_j\right), \nabla \eta_k\right),$$

  $$(\boldsymbol{S}_{up,h}(\mathbf{u}_h^n))_{ik} = (\tau_M^n \nabla \eta_k, \mathbf{u}_h^n \cdot \nabla \boldsymbol{\varphi}_i), \qquad (\boldsymbol{S}_{pp,h}(\mathbf{u}_h^n))_{kl} = (\tau_M^n \nabla \eta_l, \nabla \eta_k),$$

  for $i, j = 1, \dots, \mathbb{R}^{N_{h,u}}$, $k, l = 1, \dots, \mathbb{R}^{N_{h,p}}$;

- at the right hand side, there are the contributions of both the SUPG stabilization $\mathbf{f}_{u,s}$, $\mathbf{f}_{p,s}$, and the Dirichlet data $\mathbf{g}_u(t_{n+1})$, $\mathbf{g}_p(t_{n+1})$,

$$\boldsymbol{f}_{u,h}\left(\boldsymbol{u}_h^n\right)=\frac{1}{\Delta t}\boldsymbol{M}_h\mathbf{u}_h^n+\mathbf{g}_u(t_{n+1})+\mathbf{f}_{u,s}(\boldsymbol{u}_h^n),\qquad \boldsymbol{f}_{p,h}\left(\boldsymbol{u}_h^n\right)=\mathbf{g}_p(t_{n+1})+\mathbf{f}_{p,s}(\boldsymbol{u}_h^n),$$

where

$$(\mathbf{f}_{u,s}(\boldsymbol{u}_h^n))_i=\left(\tau_M^n\frac{1}{\Delta t}\boldsymbol{u}_h^n,\boldsymbol{u}_h^n\cdot\boldsymbol{\varphi}_i\right),\qquad (\mathbf{f}_{p,s}(\boldsymbol{u}_h^n))_k=\left(\tau_M^n\frac{1}{\Delta t}\boldsymbol{u}_h^n,\nabla\eta_k\right),$$

for $i=1,\dots,\mathbb{R}^{N_{h,u}}$, $k=1,\dots,\mathbb{R}^{N_{h,p}}$.

Note that considering a semi-implicit treatment of the nonlinear convective terms, the fully-discrete problem (11) yields a linear system to be solved at each time t_n. Even if not highlighted for the sake of notation, all the matrices and vectors above depend on the geometrical parameter vector $\boldsymbol{\mu}_g$ and, ultimately, on $\boldsymbol{\mu}$. In contrast, only the vector corresponding to the boundary data depends on the physical parameters $\boldsymbol{\mu}_p$. We also highlight that the $\boldsymbol{S}_{pp,h}$ block appearing in the SUPG matrix is nonzero, and is nonsingular, thus making the saddle point system to be solved at each time step well-posed. Indeed, the saddle point matrix features the same structure as the one obtained with the finite volume method [4], for which the (2,2) block was filled by the Laplacian of the pressure projection term.

2.3. Reduced order model

In this section, we describe the construction of reduced order models for parametrized fluid flows relying on a POD-Galerkin framework. To make the paper self-contained, we report some basic notions about proper orthogonal decomposition (POD), Galerkin projection and hyper reduction through the discrete empirical interpolation method. For the sake of simplicity, we will rely on a fully algebraic construction.

2.3.1. Proper orthogonal decomposition

In this paper, we employ POD both for the construction of RB spaces and for the efficient approximation of parameter-dependent arrays. For the sake of simplicity, we tailor our description to the former case. Let $\mathbf{s}(\boldsymbol{\tau})$ be the discrete representation of a field variable, such as the discrete flow velocity $\mathbf{u}_h$ or the pressure $\mathbf{p}_h$, at a given time, with $\boldsymbol{\tau}=(t,\boldsymbol{\mu})\in\mathcal{D}\subset\mathbb{R}^{p+1}$, and let $\mathcal{M}_z=\{\mathbf{z}(\boldsymbol{\tau})\in\mathbb{R}^h\mid\boldsymbol{\tau}\in\mathcal{D}\}$ be the manifold to be approximated. The goal of POD exploiting the so-called *snapshot method* is to approximate $\mathcal{M}_z$ with a low-dimensional optimal linear space of $\mathbb{R}^{N_h}$ starting from a set of n_s *snapshots* sampled from $\mathcal{M}_z$, $\mathbf{S}=[\mathbf{s}_1,\mathbf{z}_2,\dots,\mathbf{s}_{n_s}]\in\mathbb{R}^{N_h\times n_s}$, with $\boldsymbol{\tau}_i\in\mathcal{D}$, $i=1,\dots,n_s$. For any prescribed dimension N, POD provides the N-dimensional subspace, spanned by the columns of $\mathbf{V}=[\zeta_1|\dots|\zeta_N]\in\mathbb{R}^{N_h\times N}$, which best approximates $\{\mathbf{s}_i\}_{i=1}^{n_s}$ among all possible N-dimensional subspaces. To this end, POD computes the singular value decomposition (SVD) $\mathbf{X}^{\frac{1}{2}}\mathbf{S}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^T$ of the matrix $\mathbf{S}$ with respect to a scalar product induced by a symmetric positive definite matrix $\mathbf{X}$, where $\mathbf{U}\in\mathbb{R}^{N_h\times N_h}$ and $\mathbf{Z}\in\mathbb{R}^{n_s\times n_s}$ denote orthogonal matrices, and $\boldsymbol{\Sigma}\in\mathbb{R}^{N_h\times n_s}$ is a diagonal matrix containing the singular values $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_{n_s}\geq 0$. The matrix $\mathbf{V}$ is obtained by retaining the first N columns of $\mathbf{U}$ and represents, by construction, an $\mathbf{X}$-orthonormal basis of the best N-dimensional subspace approximating the snapshot set. In particular, we have

$$\sum_{i=1}^{n_s}\|\mathbf{s}_i-\mathbf{V}\mathbf{V}^T\mathbf{X}\mathbf{s}_i\|_{\mathbf{X}}^2=\min_{\mathbf{W}\in\mathcal{V}_N}\sum_{i=1}^{n_s}\|\mathbf{s}_i-\mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{s}_i\|_{\mathbf{X}}^2=\sum_{i=N+1}^{n_s}\sigma_i^2,$$

where $\mathcal{V}_N=\{\mathbf{W}\in\mathbb{R}^{N_h\times N}:\mathbf{W}^T\mathbf{X}\mathbf{W}=\mathbf{I}_N\}$ and $\mathbf{I}_N$ is the N-dimensional identity matrix. Moreover, the discarded singular values provide an estimate of the relative error, since

$$\frac{\sum_{i=1}^{n_s}\|\mathbf{s}_i-\mathbf{V}\mathbf{V}^T\mathbf{X}\mathbf{s}_i\|_{\mathbf{X}}^2}{\sum_{i=1}^{n_s}\|\mathbf{s}_i\|_{\mathbf{X}}^2}=\frac{\sum_{i=N+1}^{n_s}\sigma_i^2}{\sum_{i=1}^{n_s}\sigma_i^2}.\tag{12}$$

For a given tolerance ε_{POD}, (12) is employed to control the relative error on the approximation of the snapshots and to select N basis functions. In particular, we can set the bases dimension N as the minimum integer such that

$$\frac{\sum_{i=1}^{N}\sigma_i^2}{\sum_{i=1}^{n_s}\sigma_i^2}\geq 1-\varepsilon_{POD}.\tag{13}$$

In our case, snapshots of velocity and pressure solution fields are obtained by sampling the parameter space defining the problem at different instances in time. Then, POD is performed separately on each set of (velocity and pressure) snapshots $\mathbf{S}_u$ and $\mathbf{S}_p$ collected for $\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_{ns}$ at times $t^1,\dots,t^{N_t}$. To limit the computational cost of the calculation, the POD basis could be built sequentially by performing at first POD with respect to the time trajectory (for a fixed parameter value) and, after collecting this information, POD with respect to the parametric dependence. See, e.g., [23] for further details.

2.3.2. Galerkin projection

The RB approximation of velocity and pressure fields at time t_n can then be expressed as a linear combination of the RB basis functions,

$$\mathbf{u}_h^{n+1}(\boldsymbol{\mu})\approx\mathbf{V}_u\mathbf{u}_N^{n+1}(\boldsymbol{\mu}),\qquad \mathbf{p}_h^{n+1}(\boldsymbol{\mu})\approx\mathbf{V}_p\mathbf{p}_N^{n+1}(\boldsymbol{\mu}),\tag{14}$$

where $\mathbf{V}_v\in\mathbb{R}^{N_{h,u}\times N_u}$ and $\mathbf{V}_p\in\mathbb{R}^{N_{h,p}\times N_p}$ denote the matrices whose columns are the vectors of degrees of freedom of the basis functions for the velocity and the pressure RB spaces, respectively.

Substituting (14) into (11) and performing a Galerkin projection, we obtain the following POD-Galerkin problem (denoted as *parametrized ROM* (PROM) from hereon): given $\mathbf{u}_N^n$, for $n=0,\dots,N_t-1$, find $(\mathbf{u}_N^{n+1},\mathbf{p}_N^{n+1})\in\mathbb{R}^{N_u}\times\mathbb{R}^{N_p}$ such that

$$\begin{bmatrix}\frac{1}{\Delta t}\boldsymbol{M}_N+\boldsymbol{C}_N\left(\mathbf{u}_N^n\right)+\boldsymbol{L}_N & \boldsymbol{B}_N^T\\ -\boldsymbol{B}_N & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{u}_N^{n+1}\\ \mathbf{p}_N^{n+1}\end{bmatrix}+\begin{bmatrix}\boldsymbol{S}_{uu,N}\left(\mathbf{u}_N^n\right) & \boldsymbol{S}_{up,N}\left(\mathbf{u}_N^n\right)\\ \boldsymbol{S}_{pu,N}\left(\mathbf{u}_N^n\right) & \boldsymbol{S}_{pp,N}\left(\mathbf{u}_N^n\right)\end{bmatrix}\begin{bmatrix}\mathbf{u}_N^{n+1}\\ \mathbf{p}_N^{n+1}\end{bmatrix}=\begin{bmatrix}\boldsymbol{f}_{N,u}\left(\boldsymbol{u}_N^n\right)\\ \boldsymbol{f}_{N,p}\left(\boldsymbol{u}_N^n\right)\end{bmatrix},\tag{15}$$

where all the RB arrays appearing in the equation above are obtained by projecting onto the RB spaces the corresponding blocks defined in (11). In particular:

- the reduced-order mass, diffusion and pressure-divergence operators are given by

$$\boldsymbol{M}_N=\mathbf{V}_u^T\boldsymbol{M}_h\mathbf{V}_u,\qquad \boldsymbol{L}_N=\mathbf{V}_u^T\boldsymbol{L}_h\mathbf{V}_u,\qquad \boldsymbol{B}_N=\mathbf{V}_u^T\boldsymbol{B}_h\mathbf{V}_p;$$

- the reduced-order convective operator is given by

$$\boldsymbol{C}_N\left(\mathbf{u}_N^n\right)=\mathbf{V}_u^T\boldsymbol{C}_h\left(\mathbf{V}_u\mathbf{u}_N^n\right)\mathbf{V}_u;$$

- the reduced-order SUPG stabilization terms are given by

$$\boldsymbol{S}_{uu,N}\left(\mathbf{u}_N^n\right)=\mathbf{V}_u^T\boldsymbol{S}_{uu,h}\left(\mathbf{V}_u\mathbf{u}_N^n\right)\mathbf{V}_u,$$
$$\boldsymbol{S}_{up,N}\left(\mathbf{u}_N^n\right)=\mathbf{V}_u^T\boldsymbol{S}_{up,h}\left(\mathbf{V}_u\mathbf{u}_N^n\right)\mathbf{V}_p,$$
$$\boldsymbol{S}_{pu,N}\left(\mathbf{u}_N^n\right)=\mathbf{V}_p^T\boldsymbol{S}_{pu,h}\left(\mathbf{V}_u\mathbf{u}_N^n\right)\mathbf{V}_u,$$
$$\boldsymbol{S}_{pp,N}\left(\mathbf{u}_N^n\right)=\mathbf{V}_p^T\boldsymbol{S}_{pp,h}\left(\mathbf{V}_u\mathbf{u}_N^n\right)\mathbf{V}_p;$$

- finally, the reduced-order right-hand sides, also involving a SUPG contribution, are given by

$$\boldsymbol{f}_{N,u}\left(\boldsymbol{u}_N^n\right)=\mathbf{V}_u^T\boldsymbol{f}_{h,u}\left(\mathbf{V}_u\boldsymbol{u}_N^n\right),\qquad \boldsymbol{f}_{N,p}\left(\boldsymbol{u}_N^n\right)=\mathbf{V}_p^T\boldsymbol{f}_{h,p}\left(\mathbf{V}_u\boldsymbol{u}_N^n\right).$$

The dimension of the system defined by Eq. (15) is now equal to N_u+N_p, and it is much lower than the FOM dimension, $N_{h,u}+N_{h,p}$ of Eq. (11). Hence, solving the Navier–Stokes system has become computationally less expensive. However, without any further development, the PROM still requires the calculation of the reduced operators involving the projection of the corresponding FOM operators onto the POD bases. Additionally, some of the operators in Eq. (15) depend on the FOM solution, so that they would need to be recomputed at each iteration using the FOM, and then projected onto the POD

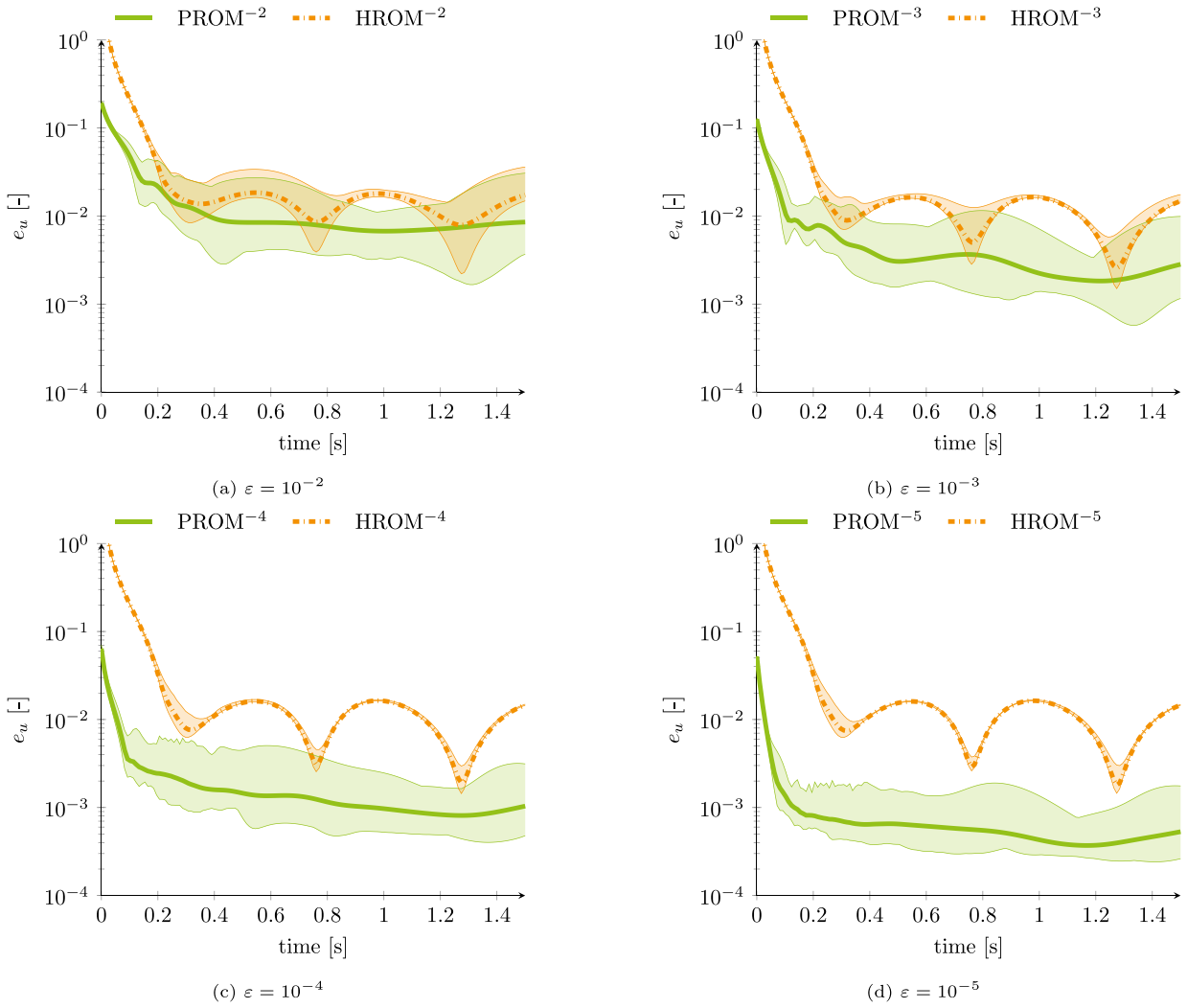


Fig. 1. Comparison between FOM, PROM and HROM predictions in terms of velocity relative error for reducing values of ε in the POD method and constant reconstruction tolerance for (M)DEIM. Each panel shows the error over time in the velocity field between the FOM and the PROM (green) or between the FOM and the HROM (orange). Solid lines show the mean values over time, computed over the training dataset, while the shaded area represents the maximum and minimum errors obtained over the training dataset.

subspace. This would only yield a negligible computational speedup, thus preventing the ROM from being a feasible alternative. For this reason, we need to rely on a further *hyper-reduction* stage to efficiently reconstruct these solution-dependent operators at each iteration: this is the goal of the following section.

We highlight that a Galerkin projection might yield a singular algebraic system (15) depending on the chosen high-fidelity FOM approximation. For instance, if stable velocity/pressure spaces were used, as in the case of a $\mathbb{P}^2$ - $\mathbb{P}^1$ FE discretization, the first matrix appearing at the left-hand side of (15) would be singular, because the POD velocity basis is divergence-free (in a discrete sense), thus violating the inf–sup condition and not allowing for the solution of the pressure based on the mass conservation principle. To overcome this, several strategies have been proposed, such as the augmentation of the velocity space through a set of enriching basis functions involving the so-called supremizing operator, the use of a Petrov–Galerkin projection, or the independent reconstruction of pressure from velocity; see Section 1.1 and references therein. The approach we have pursued, instead, hinges upon velocity POD basis functions that fulfill a stabilized problem, in which the additional stabilization terms avoid the violation of the inf–sup condition. For this reason, the ROM is automatically stable – that is, the sum of the two matrices appearing in (15) cannot be singular – and no further treatment is required in its construction.

2.3.3. (Matrix) discrete empirical interpolation method

For the problem at hand, both nonlinearities and parameter dependence make assembling the ROM arrays a computational bottleneck. For the sake of clarity, we restore the explicit parameter dependence.

Linear, time-invariant matrices such as $\mathbf{M}_N$ and $\mathbf{L}_N$ can be assembled at time $t = 0$, but they depend on geometrical parameters in a nonaffine way. In applications with fixed domain geometries, the matrix $\mathbf{C}_N$ could be ideally assembled exploiting the quadratic nonlinearity of the convective term.

For these reasons, the generation of the ROM arrays (to be used to assemble the ROM for any new parameter value during the online stage) is done using a hyper-reduction technique such as the discrete empirical interpolation method (DEIM) [45,47,55] when dealing with vector operators (such as the ones appearing at the right-hand side of (15)) or its matrix version (MDEIM) [48], similarly to the strategy proposed in [23].

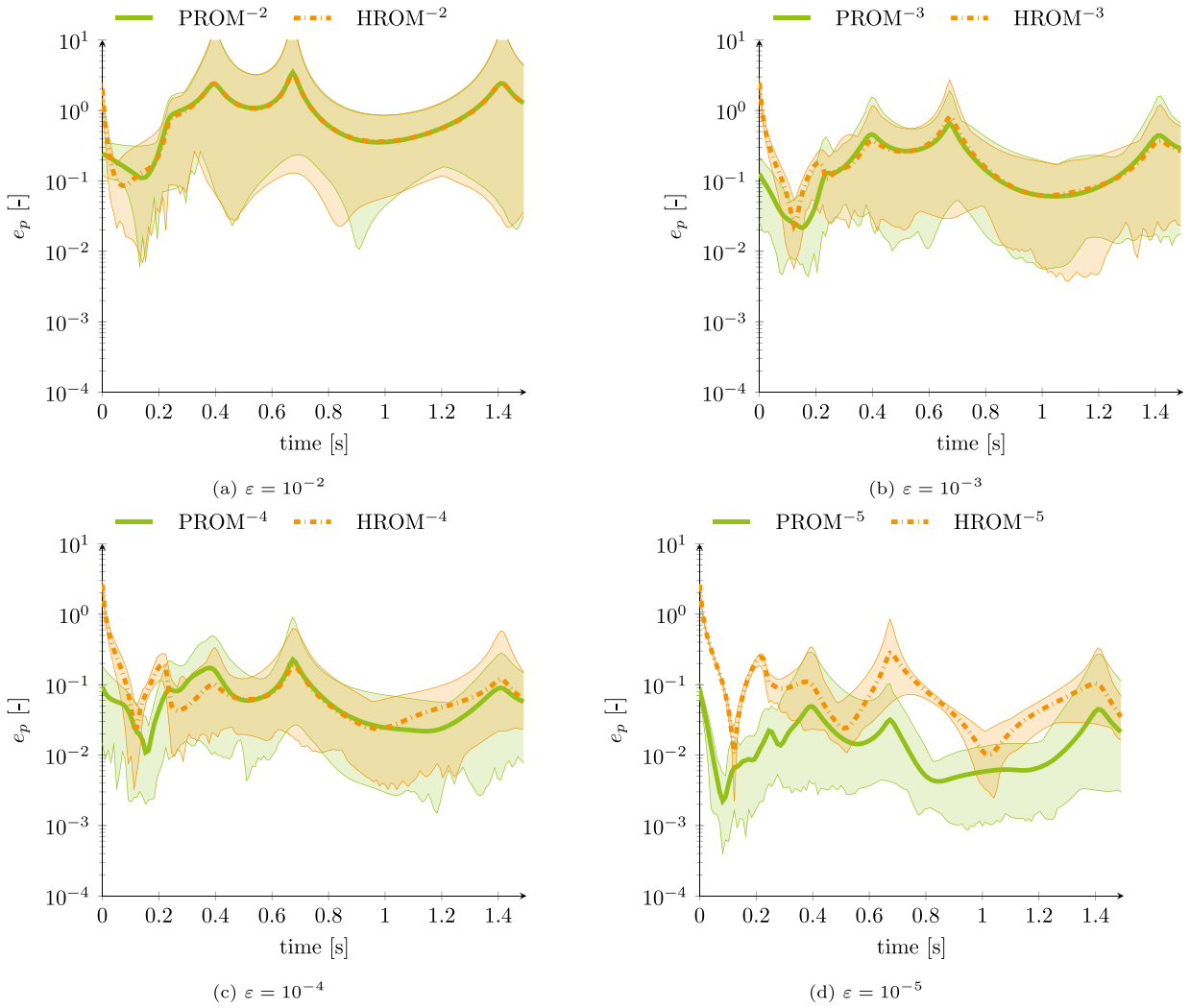


Fig. 2. Comparison between FOM, PROM and HROM predictions in terms of pressure relative error for reducing values of ε in the POD method and constant reconstruction tolerance for (M)DEIM. Each panel shows the error over time in the pressure field between the FOM and the PROM (green) or between the FOM and the HROM (orange). Solid and dashed lines show the mean values over time, computed over the training dataset, while the shaded area represents the maximum and minimum errors obtained over the training dataset.

For reference, we show here how MDEIM works, for the approximation of a generic matrix $\mathbf{A}_h(\mathbf{u}_h(\boldsymbol{\tau});\boldsymbol{\tau})$, where $\mathbf{u}_h(\boldsymbol{\tau})$ specifies the (nonlinear) dependence on the solution field and $\boldsymbol{\tau}$ the time-dependence and the (nonaffine) dependence on the input parameters – with $\boldsymbol{\tau} = (t,\boldsymbol{\mu}) \in \mathcal{D} \subset \mathbb{R}^{p+1}$. This strategy then applies to all the matrices appearing in (15) and also to the right-hand side vectors. Hence, we seek to approximate $\mathbf{A}_h(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})$ as a linear combination of m_A $(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})$-independent matrices $\mathbf{A}_1,\dots,\mathbf{A}_{m_A}$, and corresponding $(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})$-dependent weights, $\{\theta_i^A(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})\}_{i=1}^{m_A}$,

$$\begin{aligned}\mathbf{A}_h(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau}) \approx \widetilde{\mathbf{A}}_h(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau}) &= \sum_{i=1}^{m_A} \theta_i^A(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})\mathbf{A}_i \\ &= \boldsymbol{\Phi}_A \boldsymbol{\theta}_A(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau}),\end{aligned} \tag{16}$$

where $\boldsymbol{\theta}_A(\mathbf{u}_h(\boldsymbol{\tau});\boldsymbol{\tau}) \in \mathbb{R}^{m_A}$ is the vector of the coefficients to be determined and

$$\boldsymbol{\Phi}_A = \left[\boldsymbol{\phi}_1^A \mid \dots \mid \boldsymbol{\phi}_{m_A}^A\right] = [\mathrm{vec}(\mathbf{A}_1) \mid \dots \mid \mathrm{vec}(\mathbf{A}_{m_A})] \in \mathbb{R}^{N_h^2 \times m_A}$$

is a POD-basis spanning a subspace of

$$\mathcal{M}_A = \left\{ \mathrm{vec}(\mathbf{A}_h(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})) \ : \boldsymbol{\tau} = (t,\boldsymbol{\mu}) \in \{t_1,\dots,t_{N_t}\} \times \mathcal{P} \right\} \subset \mathbb{R}^{N_h^2}.$$

Such a basis can then be computed during the *offline* phase by performing POD on a set of snapshots of (vectorized matrices) $\mathbf{A}_h(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$ as described in Section 2.3. The coefficient vector $\boldsymbol{\theta}_A(\mathbf{u}_h(\boldsymbol{\tau});\boldsymbol{\tau})$ can be evaluated for each new value of $\boldsymbol{\tau}$ by imposing m_A interpolation constraints on a subset $\wp = [\wp_1,\dots,\wp_{m_A}]$ of entries of $\mathbf{A}_h(\mathbf{u}_h(\boldsymbol{\tau});\boldsymbol{\tau})$ (the so-called magic points, see e.g. [55]) selected by the DEIM algorithm, that is,

$$\boldsymbol{\theta}_A(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau}) = (\boldsymbol{\Phi}_A|_{\wp})^{-1}(\mathrm{vec}(\mathbf{A}_h(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau}))|_{\wp}).$$

Therefore, during the *online* phase, it is sufficient to calculate the algebraic operators only at the magic points, i.e., to assemble the FOM operators only on those elements of the computational mesh which contribute to the entries of $\wp$, in order to quickly reconstruct their formulation in the ROM. Finally, we can approximate $\mathbf{A}_N(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})$

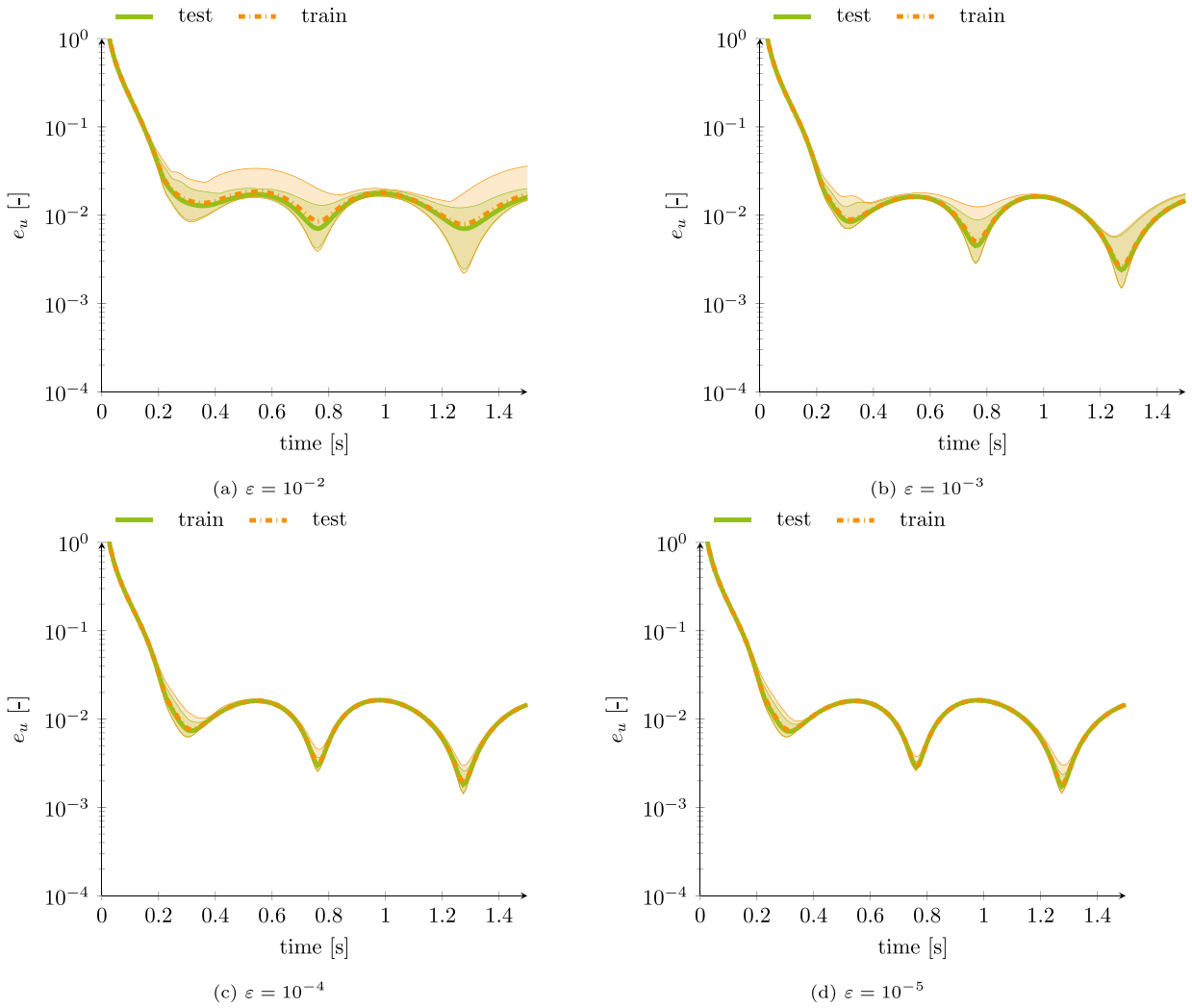


Fig. 3. Comparison between FOM and HROM predictions for training and test datasets for reducing values of ε in the POD method and constant reconstruction tolerance for (M)DEIM. Each panel shows the error over time in the velocity field for test (green) training (orange) cases. Solid and dashed lines show the mean values over time, computed over the training dataset, while the shaded area represents the maximum and minimum errors obtained over the training dataset.

by

$$\widetilde{\mathbf{A}}_N(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau}) = \mathbf{V}^T\widetilde{\mathbf{A}}_h(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})\mathbf{V} = \sum_{i=1}^{m_A}\theta_i^A(\mathbf{V}\mathbf{u}_N(\boldsymbol{\tau});\boldsymbol{\tau})\mathbf{V}^T A_i\mathbf{V},$$

hence by relying on the matrices $\mathbf{V}^T A_i\mathbf{V}$, $i = 1,\dots,m_A$, that can be precomputed once for all and stored. See, e.g., [48,56] for further details.

Replacing all the ROM arrays appearing in the PROM (15) with their (M)DEIM approximations, we obtain the following POD-Galerkin hyper-ROM (HROM): given $\mathbf{u}_{N,m}^n$, for $n = 0,\dots,N_t-1$, find $(\mathbf{u}_{N,m}^{n+1}, \mathbf{p}_N^{n+1}) \in \mathbb{R}^{N_u}\times\mathbb{R}^{N_p}$ such that

$$\begin{bmatrix} \frac{1}{\Delta t}\widetilde{\boldsymbol{M}}_N + \widetilde{\boldsymbol{C}}_N\left(\mathbf{u}_{N,m}^n\right) + \widetilde{\boldsymbol{L}}_N & \widetilde{\boldsymbol{B}}_N^T \\ -\widetilde{\boldsymbol{B}}_N & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{N,m}^{n+1} \\ \mathbf{p}_{N,m}^{n+1} \end{bmatrix} + \begin{bmatrix} \widetilde{\boldsymbol{S}}_{uu,N}\left(\mathbf{u}_{N,m}^n\right) & \widetilde{\boldsymbol{S}}_{up,N}\left(\mathbf{u}_{N,m}^n\right) \\ \widetilde{\boldsymbol{S}}_{pu,N}\left(\mathbf{u}_{N,m}^n\right) & \widetilde{\boldsymbol{S}}_{pp,N}\left(\mathbf{u}_{N,m}^n\right) \end{bmatrix} \begin{bmatrix} \mathbf{u}_{N,m}^{n+1} \\ \mathbf{p}_{N,m}^{n+1} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{f}}_{N,u}\left(\boldsymbol{u}_{N,m}^n\right) \\ \widetilde{\boldsymbol{f}}_{N,p}\left(\boldsymbol{u}_{N,m}^n\right) \end{bmatrix}, \tag{17}$$

where we have also denoted the dependence of the solution on the hyper-reduced operators.

Note that the snapshots required to determine the reduced bases to approximate the ROM arrays depending on $\mathbf{V}_u\mathbf{u}_N$ through hyper-reduction are generated by solving the parametrized ROM (15), rather than the FOM. In the case where the HROM arrays only depend on time and parameters (as in the case of linear mass, diffusion and pressure/divergence operators), the assembling of the matrix snapshots does not require to solve any further problem.

3. Results

We considered as reference geometry a cylindrical conduit (diameter $d_0 = 10.0$ mm, length $L_0 = 100.0$ mm) with a prescribed section restriction of severity $\mu_g \in [0.1, 0.4]$, see Section 2.1. We only considered the restriction severity as input parameter, keeping the restriction position as fixed, by choosing $c_1 = 0.01$ and $c_2 = 0.02$, respectively. The time averaged mean inlet blood velocity, μ_p, was chosen to yield peak inlet averaged Reynolds numbers between 300 and 500. We selected an oscillation frequency of the input velocity profile, $f = 1.5$ Hz. These parameters define realistic stenosis severity and physiological flow velocity of interest for clinical diagnosis of coronary

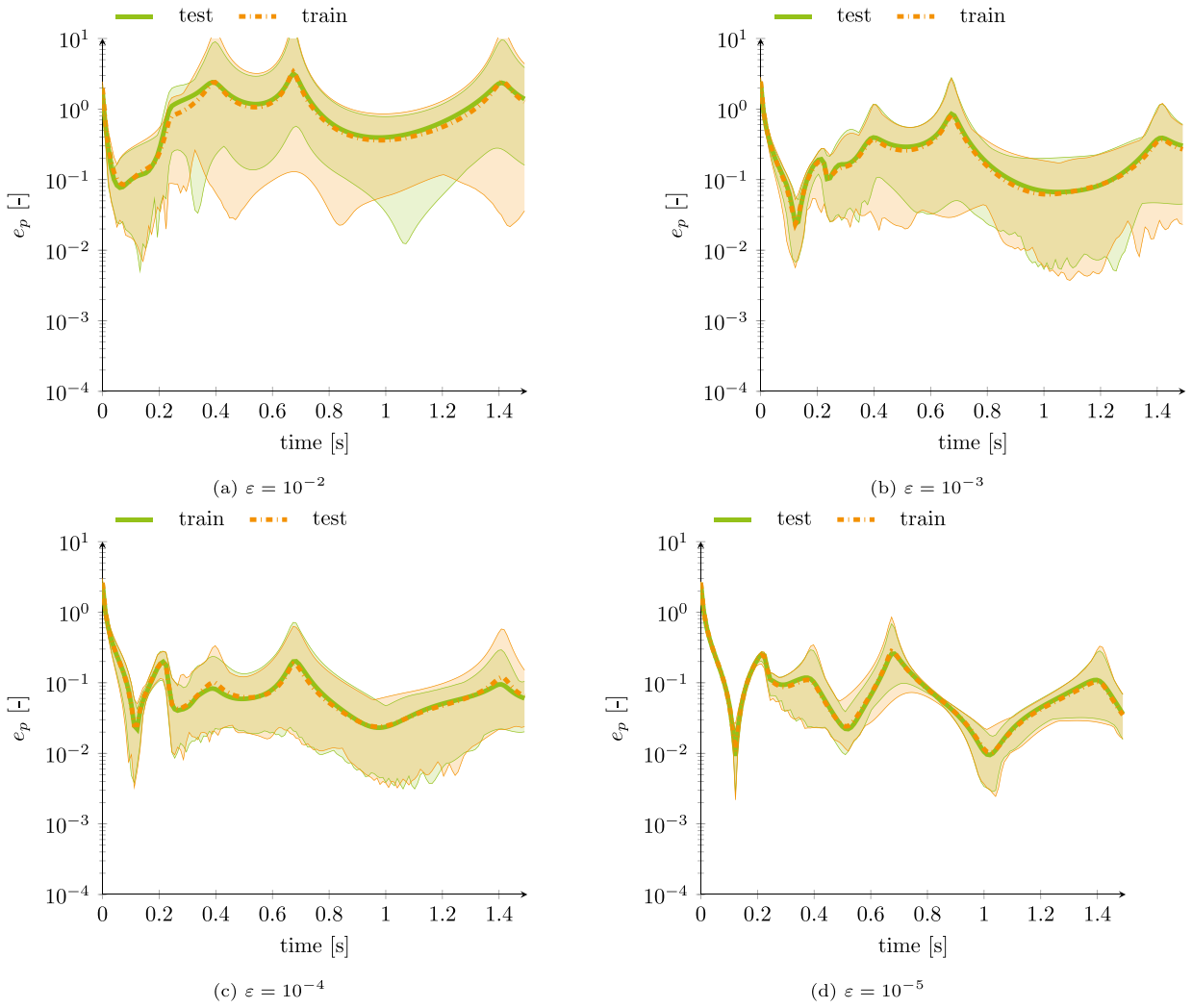


Fig. 4. Comparison between FOM and HROM predictions for training and test datasets for reducing values of ε in the POD method and constant reconstruction tolerance for (M)DEIM. Each panel shows the error over time in the pressure field for test (green) training (orange) cases. Solid and dashed lines show the mean values over time, computed over the training dataset, while the shaded area represents the maximum and minimum errors obtained over the training dataset.

Table 1
Number of POD bases for velocity, N_v, and pressure, N_p approximations in PROMs and HROMs.

ROM	ε	N_v	N_p
PROM^{-2}/HROM^{-2}	10^{-2}	5	3
PROM^{-3}/HROM^{-3}	10^{-3}	13	5
PROM^{-4}/HROM^{-4}	10^{-4}	28	9
PROM^{-5}/HROM^{-5}	10^{-5}	49	17

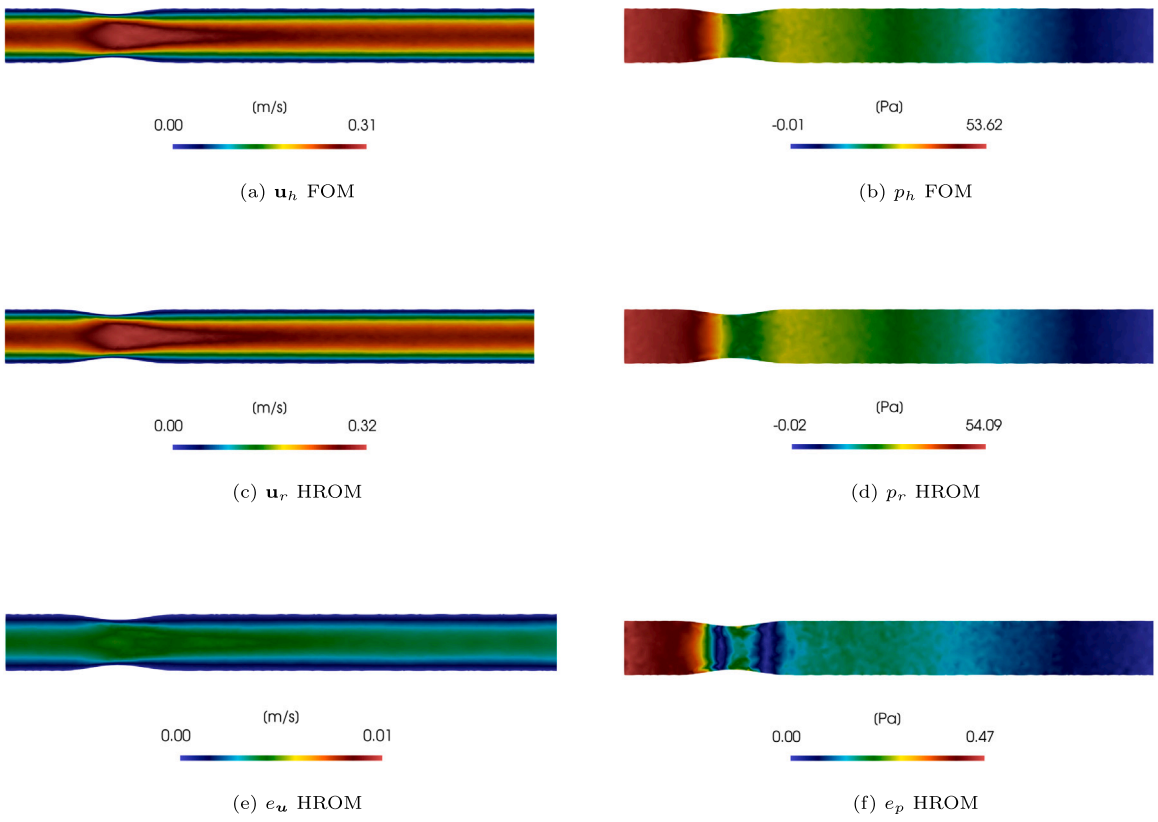
artery disease. We modeled blood as an incompressible Newtonian fluid with density, $\rho = 1060$ kg m^{-3}, and kinematic viscosity, $\nu = 0.004$ Pa·s. We solved the discretized algebraic NS equations described in (11) using the open-source finite element library `FEniCS` [57,58].

Training datasets were generated by sampling the parametric space (μ_p, μ_g) with a 9×7 grid, resulting in 63 cases. We ran simulations for a total time of $T = 1.5$ s and with a time step $\Delta t = 10^{-4}$ s. The inlet velocity profile was initially ramped to its maximum value in the oscillatory cycle for t ∈ [0,0.2] s before the oscillatory component was added. Snapshots for velocity and the algebraic forms of the NS operators were saved at time intervals $\Delta t = 2 \times 10^{-3}$ s.

We compare the results obtained with either the PROM or the HROM against the ones computed through the high-fidelity FOM. Recall that in the PROM, the algebraic operators are obtained by projecting the FOM arrays onto the POD velocity/pressure spaces and then solved for the reduced approximation (the amplitudes of the POD modes). In the HROM, instead, the system algebraic operators are approximated through (M)DEIM for the sake of computational efficiency. We ran the PROM and HROM for different values of the POD tolerance, ε_{POD}, chosen equal to $\left[10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right]$ and keeping the value of the (M)DEIM tolerance for each approximated operator at 10^{-12}. Therefore, we obtained 5 different PROMs and HROMs, which are identified with a superscript reporting the corresponding POD tolerance (e.g., PROM^{-5} for a POD tolerance of 10^{-5}). Table 1 summarizes the resulting number of POD velocity and pressure basis functions, for either the PROMs or the HROMs. Table 2 reports instead the resulting number of basis for the reconstruction of the operators in the HROMs, when different tolerances are used to stop the (M)DEIM approximation of the ROM operators. As expected, the most difficult terms to be approximated are the ones related to either convective or SUPG terms.

Table 2
Number of DEIM bases for the reconstruction of the algebraic operators: mass $\widetilde{\boldsymbol{M}}_N$; convective $\widetilde{\boldsymbol{C}}_N(\mathbf{u}^n_{N,m})$; diffusion $\widetilde{\boldsymbol{L}}_N$; pressure/divergence $\widetilde{\boldsymbol{B}}_N$; stabilization terms $\widetilde{\boldsymbol{S}}_{uu,N}(\mathbf{u}^n_{N,m})$, $\widetilde{\boldsymbol{S}}_{up,N}(\mathbf{u}^n_{N,m})$, $\widetilde{\boldsymbol{S}}_{pu,N}(\mathbf{u}^n_{N,m})$, $\widetilde{\boldsymbol{S}}_{pp,N}(\mathbf{u}^n_{N,m})$; right hand sides $\widetilde{\boldsymbol{f}}_{N,u}(\boldsymbol{u}^n_{N,m})$, $\widetilde{\boldsymbol{f}}_{N,p}(\boldsymbol{u}^n_{N,m}t)$. For their definition we refer the reader to Section 2.3.2.

ROM	$\widetilde{\boldsymbol{M}}_N$	$\widetilde{\boldsymbol{L}}_N$	$\widetilde{\boldsymbol{B}}_N$	$\widetilde{\boldsymbol{C}}_N\left(\mathbf{u}^n_{N,m}\right)$	$\widetilde{\boldsymbol{S}}_{uu,N}$	$\widetilde{\boldsymbol{S}}_{up,N}$	$\widetilde{\boldsymbol{S}}_{pu,N}$	$\widetilde{\boldsymbol{S}}_{pp,N}$	$\widetilde{\boldsymbol{f}}_{N,u}\left(\boldsymbol{u}^n_{N,m}\right)$	$\widetilde{\boldsymbol{f}}_{N,p}\left(\boldsymbol{u}^n_{N,m}\right)$
HROM^{-2}	2	4	2	16	12	17	15	5	37	16
HROM^{-3}	2	4	2	35	17	35	31	5	115	37
HROM^{-4}	2	4	2	65	25	61	49	5	225	68
HROM^{-5}	2	4	2	96	27	87	70	5	280	99



(a) $\mathbf{u}_h$ FOM (b) p_h FOM

(c) $\mathbf{u}_r$ HROM (d) p_r HROM

(e) $e_{\boldsymbol{u}}$ HROM (f) e_p HROM

Fig. 5. Comparison of velocity (left column) and pressure (right column) for $(\mu_p, \mu_g) = (500, 0.1)$. (a) and (b) show the FOM solutions at the end of the simulation (peak velocity instant), (c) and (d) the corresponding solution with HROM^{-5} and (e) and (f) the absolute difference between the two.

After the generation of all the terms necessary for building PROMs and HROMS at the online phases, we use both models on 48 test cases corresponding to parameter values (μ_p, μ_g) not included in the 63 training 63. Solution accuracy is evaluated by considering the relative error in the prediction of both velocity and pressure fields over time. For each value of $\boldsymbol{\mu} \in \mathcal{P}$, the (instantaneous) relative velocity/pressure errors are computed as

$$e_u(t) = \frac{\|\mathbf{u}_r(\boldsymbol{\mu}) - \mathbf{u}_h(\boldsymbol{\mu})\|_2}{\|\mathbf{u}_h(\boldsymbol{\mu})\|_2}, \qquad e_p(t) = \frac{\|\mathbf{p}_r(\boldsymbol{\mu}) - \mathbf{p}_h(\boldsymbol{\mu})\|_2}{\|\mathbf{p}_h(\boldsymbol{\mu})\|_2},$$

where $\mathbf{u}_r$, $\mathbf{p}_r$ and $\mathbf{u}_h$, $\mathbf{p}_h$ refer to the velocity/pressure field in the PROM/HROM and the FOM, respectively.

Figs. 1 and 2 show the average of the instantaneous errors e_u and e_p, over time, computed over the training datasets for the PROMs and HROMs. Each panel corresponds to a fixed POD tolerance, ε.

In Figs. 3 and 4, we compare the reconstruction errors computed over the training and the test datasets for each of the values of ε considered.

Finally, Figs. 5 and 6 show the comparison of the velocity magnitude and the pressure field predicted by the FOM and the HROM^{-5} at the end of the simulation, for which we also are at the peak value of the inlet velocity profile. In particular, the panels show the contour plot on a section of the computational domain.

We used multiple reconstruction tolerances for the selection of the number of the POD bases for the reconstruction of the pressure and velocity fields: from 5 to 49 bases for the velocity and 3 to 17 for pressure when moving from $\varepsilon = 10^{-2}$ to 10^{-5}. By increasing the number of POD bases, we observe an increase in the number of reconstruction terms for (M)DEIM, in particular for the convective term and the stabilization contributions at the right-hand side of Eq. (15). At the initial time steps, we observe the largest relative errors, of the order 10^{-1} for the velocity reconstruction for all PROMs, due to the low

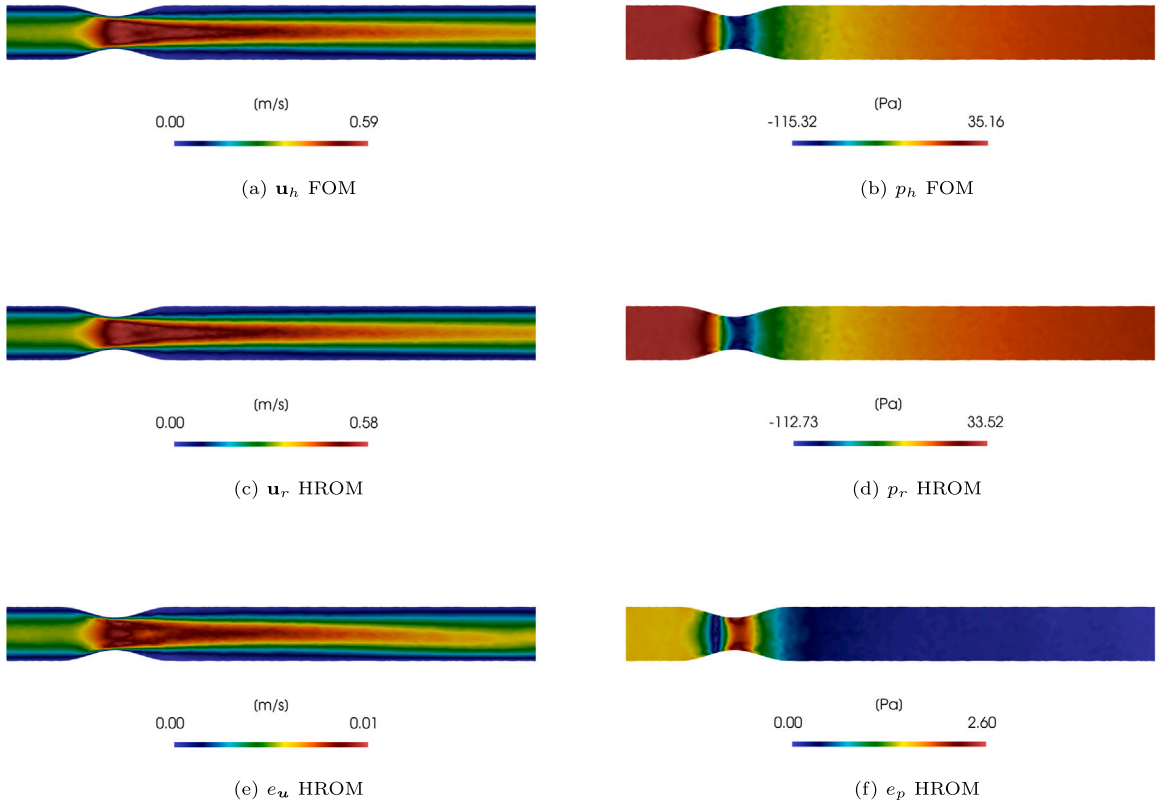


Fig. 6. Comparison of velocity (left column) and pressure (right column) for (μ_p, μ_g) = (500, 0.4). (a) and (b) show the FOM solutions at the end of the simulation (peak velocity instant), (c) and (d) the corresponding solution with HROM^{-5} and (e) and (f) the absolute difference between the two.

velocity values in the reference velocity fields. We further observe that the error quickly decreases over time to average values from 10^{-2} to 10^{-3} for PROM^{-2} and PROM^{-5}, respectively. Pressure reconstruction errors are significantly larger than those of the velocity for low numbers of bases, but decrease to average values below 10^{-2} for the more strict tolerance requirement (PROM^{-5}). For the PROMs, we observe in general that an increase in the number of basis functions leads to a reduction of the solution error for both velocity and pressure, as one may expect. Even more importantly, no degradation of flow stability was observed when solving the PROM.

When considering the HROMs, approximating the algebraic operators using (M)DEIM, we observe a degradation of the solution accuracy for both velocity and pressure. As in the PROMs, the velocity error is higher in the initial part of the time interval, then it rapidly decreases, reaching an almost constant value. In contrast to the PROMs, in the HROMs we observe limited benefits for the velocity solution from the increase of the number of the bases. The approximation with (M)DEIM increases the mean velocity error up to an order of magnitude as compared to the corresponding PROMs. The pressure solution with HROM, instead, benefits from an increase in the number of bases. Relative pressure solution errors, in fact, decrease from approximately 1 (with HROM^{-2}) to about 10^{-1} for HROM^{-4}. Errors then stabilize and the increase in the number of bases only reduces the error range around the mean value. The error in the prediction of the pressure drop follows the trend of the pressure error and presents similar mean values (10^{-1}).

4. Discussion

In this study, we have presented a way to build a stable and accurate reduced-order model for the efficient approximation of parametrized three-dimensional, incompressible unsteady Navier–Stokes equations. The model is based on a stabilized finite element high-fidelity full-order model involving the streamline upwind Petrov–Galerkin (SUPG) method, and the subsequent projection onto low-dimensional subspaces for velocity and pressure obtained applying Proper Orthogonal Decomposition (POD) to a set of velocity/pressure snapshots computed for selected parameter values. We have used a tensorial approach to sample the parameter space of interest and generate the projection subspaces. The efficient reconstruction of the algebraic Navier–Stokes operators and stabilization terms is obtained via the Discrete Empirical Interpolation Method (DEIM) and its matrix version MDEIM. This choice allows to accommodate the nonaffine parameter dependence of each array of the ROM on the parameter-dependent geometries. The framework is applied to the pulsative flow into an idealized stenotic vessel with varying degree of stenosis and inlet mass flow. Overall, we observe that the hyper-reduction through (M)DEIM of the stabilization terms allows to obtain a stable pressure–velocity coupling in the ROM without the need of any additional enrichment of the velocity subspace through, e.g., supremizing basis functions. This has been observed for unsteady pulsatile cases, with geometric variations of the fluid domain and physical variations of the inlet boundary conditions.

The errors of the ROMs (PROMs and HROMs) need to be considered in relation to the reduction of the computational cost. In our application, we observe only a marginal speedup from the FOM to the PROM. For instance, in the case of the PROM^{-5}, the speedup is only of about 1.5 times. However, when considering the HROMs, we observe a reduction of the computational cost by a factor of 8, corresponding to a reduction of the computing time from 0.8 s for a single iteration of the FOM to 0.1 s for the HROM^{-5}, on a single CPU core (Intel©Xeon©E5-1630). This 8-fold speedup can be further increased to a factor of 16, if

the HROM^{-4} is considered, for which solution errors are indeed similar to the HROM^{-5}.

As a final remark, we point out that we could obtain stable results only with DEIM reconstruction tolerances of 10^{-12}. Larger values of this tolerance that might yield even more substantial speedups, do not allow the correct approximation of the operators, ultimately making the HROM solution quickly divergent. For smaller tolerances, numerical errors in the reconstruction of the calculation of the bases were found to reduce the accuracy of the solution. This very narrow range of tolerances for which DEIM provides stable solutions is found to be consistent with previous work on reduced-order modeling [59], making the use of hyper-reduction techniques an expensive task and a delicate issue in the context at hand – where instabilities might arise for different reasons, such as, e.g., moderate/high Reynolds numbers, or the pressure treatment. Also for this reason, more recently, deep learning (DL)-based ROMs have been proposed to learn in a non-intrusive way both the nonlinear trial manifold and the reduced dynamics, by relying on deep neural networks after performing a former dimensionality reduction through POD [12]. This framework has been recently applied to parametrized fluid dynamics problems in [13], ultimately yielding the possibility to perform real-time fluid flow simulations. A similar combination of dimensionality reduction through POD and neural networks has been applied to problems in continuum mechanics in [14]. Nevertheless, the approach proposed in this work provides a reliable, physics-based ROM, whose training time is comparable to the one needed for the training of a neural network architecture, almost the same amount of training data, and a better versatility with respect to existing projection-based ROMs due to the automatic generation of stabilization terms in the ROM.

5. Conclusion

We have presented a methodology for generating a parametrized reduced-order model from a stabilized Finite Element discretization of the three-dimensional, incompressible unsteady Navier–Stokes equations. The resulting ROM resolves the pressure–velocity coupling by preserving the effect of the SUPG stabilization of the full order model, without the use of supremizer basis functions in the velocity subspace for the stabilization at the ROM level. We investigated the effect of increasing the number of bases for the reconstruction of velocity and pressure fields, and of the approximation of the algebraic operators of the NS equations through (M)DEIM. For the best trade-off between accuracy and speedup (HROM^{-4}), we observe average relative prediction errors of the full velocity and pressure fields of less than 10%, accompanied by a speedup of about 16 times compared to the conventional full order model.

CRediT authorship contribution statement

Stefano Buoso: Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Andrea Manzoni:** Conceptualization, Methodology, Writing – review & editing, Visualization. **Hatem Alkadhi:** Supervision, Funding acquisition. **Vartan Kurtcuoglu:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

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.

Acknowledgments

The authors acknowledge the financial support of the University of Zurich, Switzerland through the Forschungskredit Postdoc Fellowship (FK-18-043) and of the Swiss National Science Foundation through NCCR Kidney.CH.

Appendix

The methodology presented in this manuscript has also been applied, for the sake of testing, to one of the standard benchmark cases for fluid dynamics: the two dimensional lid-driven cavity flow (Fig. 7). The geometry is a square of dimensions L=1 m and D = 1 m. A Cartesian coordinate system is centered at the bottom-left corner with axes aligned with the edges of the domain. The cavity is meshed with a uniform grid with 50 nodes on each edge. Homogeneous Dirichlet boundary conditions are prescribed at the walls of the cavity and a uniform velocity, $\boldsymbol{u}_{lid} = (u, 0)$, is set at the top edge. The Reynolds number is defined as $\mathrm{Re} = \frac{\rho u D}{\mu}$. The simulation is run with the same FOM described in Section 2.2 for a total time of $T = 10$ s and a time step of dt=5 ms. We considered 10 training cases with Re equally spaced in the range [200-2000] and saved 200 snapshots for both velocity and pressure fields. We evaluated HROMs performance for POD tolerances, ϵ_{POD}, chosen equal to 10^{-2} and 10^{-4} with (M)DEIM fixed to 10^{-12}. Tables 3 and 4 summarize the resulting number of POD velocity and pressure bases and (M)DEIM bases for the two cases. As the geometry is fixed, the mass, diffusion and divergence operators are constant for all cases and do not need to be approximated using (M)DEIM. Test cases are generated considering velocity values obtained from the mid points of the intervals generated from the training samples to maximize the distance between training and test cases. Prediction errors over time of velocity and pressure fields between HROM and FOM are shown in Figs. 8 and 9. The HROMs are stable for both POD tolerances, with a decrease in mean error when increasing the number of POD bases. Further increase of velocity and pressure modes did not provide substantial benefits in error reduction. Fig. 10 shows the vortex structures at three instants of the simulation for FOM and HROMs and demonstrates the good agreement between the prediction of the solution fields.

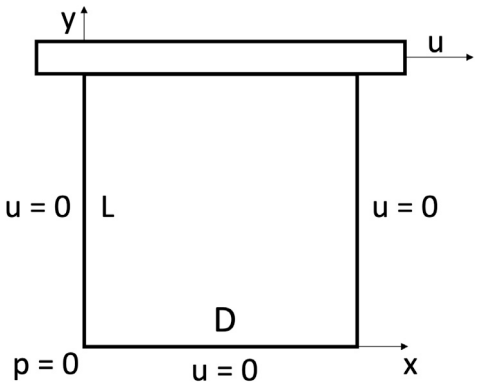


Fig. 7. Geometry of the lid-driven cavity flow benchmark.

Table 3
Number of POD bases for velocity, N_v, and pressure, N_p approximations in PROMs and HROMs for the lid-driven cavity flow problem.

ROM	ϵ	N_v	N_p
HROM^{-2}	10^{-2}	10	9
HROM^{-4}	10^{-4}	29	25

Table 4
Number of DEIM bases for the reconstruction of the algebraic operators in the lid-driven cavity flow problem: convective $\widetilde{\boldsymbol{C}}_N(\mathbf{u}^n_{N,m})$; stabilization terms $\widetilde{\boldsymbol{S}}_{uu,N}(\mathbf{u}^n_{N,m})$, $\widetilde{\boldsymbol{S}}_{up,N}(\mathbf{u}^n_{N,m})$, $\widetilde{\boldsymbol{S}}_{pu,N}(\mathbf{u}^n_{N,m})$, $\widetilde{\boldsymbol{S}}_{pp,N}(\mathbf{u}^n_{N,m})$; right hand sides $\widetilde{\boldsymbol{f}}_{N,u}(\boldsymbol{u}^n_{N,m})$, $\widetilde{\boldsymbol{f}}_{N,p}(\boldsymbol{u}^n_{N,m}t)$. For their definition we refer the reader to Section 2.3.2.

ROM	$\widetilde{\boldsymbol{C}}_N\left(\mathbf{u}^n_{N,m}\right)$	$\widetilde{\boldsymbol{S}}_{uu,N}$	$\widetilde{\boldsymbol{S}}_{up,N}$	$\widetilde{\boldsymbol{S}}_{pu,N}$	$\widetilde{\boldsymbol{S}}_{pp,N}$	$\widetilde{\boldsymbol{f}}_{N,u}\left(\boldsymbol{u}^n_{N,m}\right)$	$\widetilde{\boldsymbol{f}}_{N,p}\left(\boldsymbol{u}^n_{N,m}\right)$
HROM^{-2}	11	33	10	11	1	44	10
HROM^{-4}	30	55	29	30	1	99	29

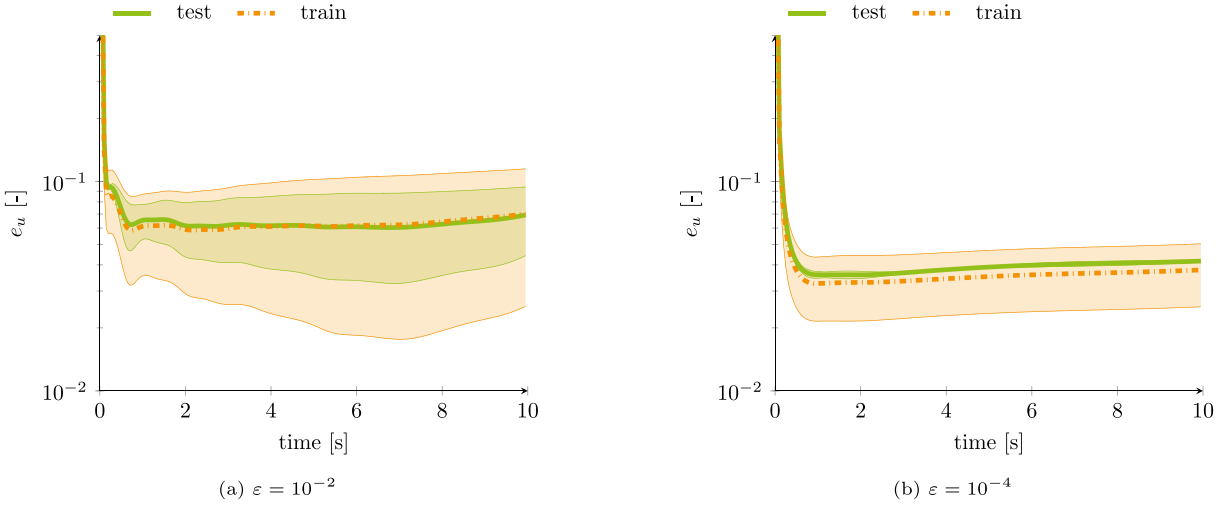


Fig. 8. Comparison between FOM and HROM predictions for training and test datasets of the lid-driven cavity flow for reducing values of ε in the POD method and constant reconstruction tolerance for (M)DEIM. Each panel shows the error over time in the velocity field for test (green) training (orange) cases. Solid and dashed lines show the mean values over time, computed over the training dataset, while the shaded area represents the maximum and minimum errors obtained over the training dataset.

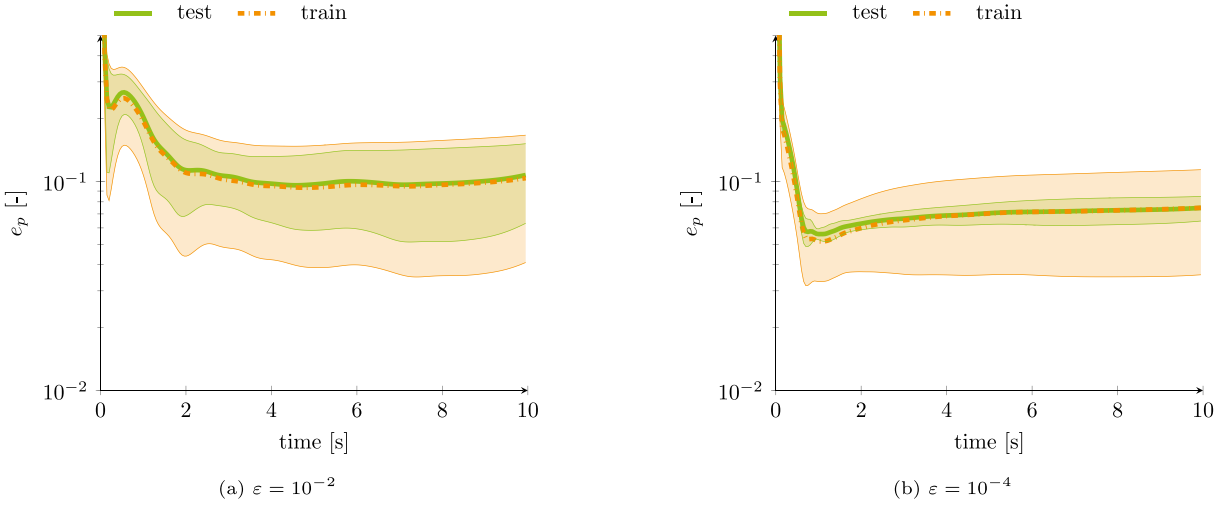


Fig. 9. Comparison between FOM and HROM predictions for training and test datasets of the lid-driven cavity flow for reducing values of ε in the POD method and constant reconstruction tolerance for (M)DEIM. Each panel shows the error over time in the pressure field for test (green) training (orange) cases. Solid and dashed lines show the mean values over time, computed over the training dataset, while the shaded area represents the maximum and minimum errors obtained over the training dataset.

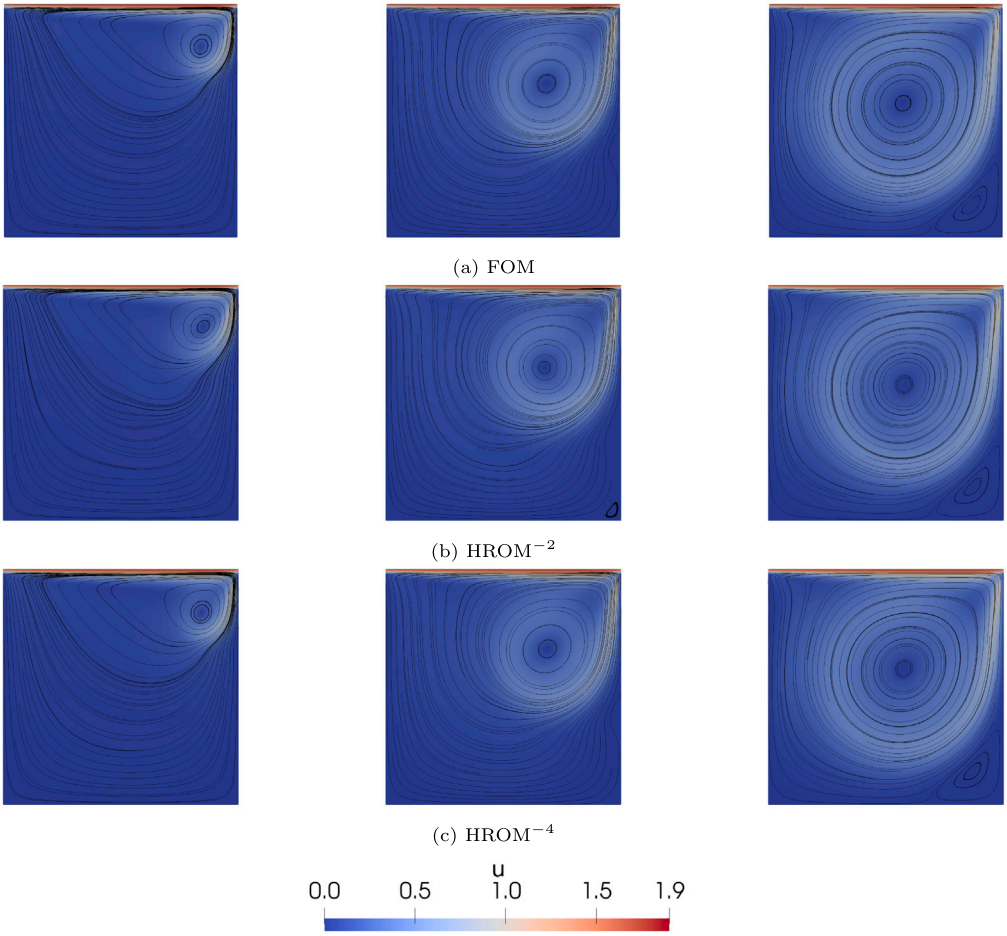


Fig. 10. Comparison of velocity fields for the lid-driven cavity flow for the test case with Re= 1900. (a), (b) and (c) show predicted solutions at three time instant of the simulation for the FOM, HROM^{-2} and HROM^{-4} models, respectively.

References

[1] Bove EL, de Leval MR, Migliavacca F, Guadagni G, Dubini G. Computational fluid dynamics in the evaluation of hemodynamic performance of cavopulmonary connections after the norwood procedure for hypoplastic left heart syndrome. J Thorac Cardiovasc Surg 2003;126(4):1040–7.

[2] Malinauskas RA, Hariharan P, Day SW, Herbertson LH, Buesen M, Steinseifer U, Aycock KI, Good BC, Deutsch S, Manning KB, Craven BA. FDA benchmark medical device flow models for CFD validation. ASAIO J 2017;63.

[3] Quarteroni A, Manzoni A, Vergara C. The cardiovascular system: mathematical modelling, numerical algorithms and clinical applications. Acta Numer 2017;26:365–590.

[4] Buoso S, Manzoni A, Alkadhi H, Plass A, Quarteroni A, Kurtcuoglu V. Reduced-order modeling of blood flow for noninvasive functional evaluation of coronary artery disease. Biomech Model Mechanobiol 2020;18(6):1867–81.

[5] Brunton S, Noack B, Koumoutsakos P. Machine learning for fluid mechanics. Annu Rev Fluid Mech 2020;52:477–508.

[6] Kutz J. Deep learning in fluid dynamics. J Fluid Mech 2017;814:1–4.

[7] Itu L, Rapaka S, Passerini T, Georgescu B, Schwemmer C, Schoebinger M, Flohr T, Sharma P, Comaniciu D. A machine-learning approach for computation of fractional flow reserve from coronary computed tomography. J Appl Physiol 2016;121:42–52.

[8] Raissi M, Perdikaris P, Karniadakis G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.

[9] Kissas G, Yang Y, Hwuang E, Witschey W, Detre J, Perdikaris P. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. Comput Methods Appl Mech Engrg 2020;358:112623.

[10] Raissi M, Yazdani A, Karniadakis G. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science 2020;367(6481):1026–30.

[11] Fresca S, Dedè L, Manzoni A. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. J Sci Comput 2021;87(2):1–36.

[12] Fresca S, Manzoni A. POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. 2021, arXiv preprint arXiv:2101.11845.

[13] Fresca S, Manzoni A. Real-time simulation of parameter-dependent fluid flows through deep learning-based reduced order models. 2021, arXiv preprint arXiv:2106.05722.

[14] Buoso S, Joyce T, Kozerke S. Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks. Med Image Anal 2021;102066.

[15] Quarteroni A, Manzoni A, Negri F. Reduced basis methods for partial differential equations: an introduction. UNITEXT - la matematica per il 3+2, vol. 92, Springer International Publishing; 2016.

[16] Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J Numer Anal 2002;40(2):492–515.

[17] Gunzburger MD, Peterson JS, Shadid JN. Reducer-order modeling of time-dependent PDEs with multiple parameters in the boundary data. Comput Methods Appl Mech Engrg 2007;196:1030–47.

[18] Bergmann M, Bruneau C-H, Iollo A. Enablers for robust POD models. J Comput Phys 2009;228(2):516–38.

[19] Weller J, Lombardi E, Bergmann M, Iollo A. Numerical methods for low-order modeling of fluid flows based on POD. Internat J Numer Methods Fluids 2010;63(2):249–68.

[20] Buoso S, Palacios R. Electro-aeromechanical modelling of integrally actuated membrane wings. J Fluids Struct 2015;58:188–202.

[21] Buoso S, Dickinson B, Palacios R. Bat-inspired integrally actuated membrane wings with leading edge sensing. Bioinspiration Biomim 2018;16(1):016013.

[22] Ballarin F, Manzoni A, Quarteroni A, Rozza G. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. Internat J Numer Methods Engrg 2015;102(5):1136–61.

[23] Dal Santo N, Manzoni A. Hyper-reduced order models for parametrized unsteady Navier-Stokes equations on domains with variable shape. Adv Comput Math 2019;45(5):2463–501.

[24] Veroy K, Patera A. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. Int J Numer Methods Fluids 2005;47(8-9):773–88.

[25] Deparis S. Reduced basis error bound computation of parameter-dependent Navier–Stokes equations by the natural norm approach. SIAM J Numer Anal 2008;46(4):2039–67.
[26] Manzoni A. An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier-Stokes flows. ESAIM Math Model Numer Anal 2014;48(4):1199–226.
[27] Yano M. A space-time Petrov–Galerkin certified reduced basis method: Application to the Boussinesq equations. SIAM J Sci Comput 2014;36(1):A232–66.
[28] Buoso S, Palacios R. On-demand aerodynamics of integrally actuated membranes with feedback control. AIAA J 2017;55(2):377–88.
[29] Lorenzi S, Cammi A, Luzzi L, Rozza G. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. Comput Methods Appl Mech Engrg 2016;311:151–79.
[30] Stabile G, Rozza G. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier Stokes equations. Comput & Fluids 2018;173:273–84.
[31] Rozza G, Huynh D, Manzoni A. Reduced basis approximation and error bounds for Stokes flows in parametrized geometries: roles of the inf–sup stability constants. Numer Math 2013;125(1):115–52.
[32] Carlberg K, Bou-Mosleh C, Farhat C. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. Internat J Numer Methods Engrg 2011;86(2):155–81.
[33] Dal Santo N, Deparis S, Manzoni A, Quarteroni A. An algebraic least squares reduced basis method for the solution of parametrized Stokes equations. Comput Methods Appl Mech Engrg 2019;344:186–208.
[34] Caiazzo A, Iliescu T, John V, Schyschlowa S. A numerical investigation of velocity–pressure reduced order models for incompressible flows. J Comput Phys 2014;259:598–616.
[35] Wang Z, Akhtar I, Borggaard J, Iliescu T. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. Comput Methods Appl Mech Engrg 2012;237:10–26.
[36] Stabile G, Ballarin F, Zuccarino G, Rozza G. A reduced order variational multiscale approach for turbulent flows. Adv Comput Math 2018;45:2349–68.
[37] Fick L, Maday Y, Patera AT, Taddei T. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. J Comput Phys 2018;371:214–43.
[38] Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. J Comput Phys 2013;242:623–47.
[39] Carlberg K, Barone M, Antil H. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. J Comput Phys 2017;330:693–734.
[40] Xie X, Mohebujjaman M, Rebholz LG, Iliescu T. Data-driven filtered reduced order modeling of fluid flows. SIAM J Sci Comput 2018;40(3):B834–57.
[41] Hijazi S, Stabile G, Mola A, Rozza G. Data-driven POD-Galerkin reduced order model for turbulent flows. J Comput Phys 2020;416:109513.
[42] Xie X, Webster C, Iliescu T. Closure learning for nonlinear model reduction using deep residual neural network. Fluids 2020;5(1):39.
[43] Baiges J, Codina R, Idelsohn S. Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations. Internat J Numer Methods Fluids 2013;72(12):1219–43.
[44] Negri F. Efficient reduction techniques for the simulation and optimization of parametrized systems: analysis and applications (Ph.D. thesis), EPFL Lausanne; 2015.
[45] Barrault M, Maday Y, Nguyen N, Patera A. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C R Acad Sci Paris Sér I Math 2004;339(9):667–72.
[46] Maday Y, Nguyen NC, Patera AT, Pau SH. A general, multipurpose interpolation procedure: the magic points. Commun Pure Appl Anal 2009;8(1):383–404.
[47] Chaturantabut S, Sorensen D. Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 2010;32(5):2737–64.
[48] Negri F, Manzoni A, Amsallem D. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. J Comput Phys 2015;303:431–54.
[49] Boffi D, Brezzi F, Fortin M. Mixed finite element methods and applications. 1st ed.. Berlin: Springer; 2013.
[50] Brooks A, Hughes T. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput Methods Appl Mech Engrg 1982;32(1–3):199–259.
[51] Bazilevs Y, Calo V, Cottrell J, Hughes T, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput Methods Appl Mech Engrg 2007;197(1):173–201.
[52] Forti D, Dedè L. Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a high performance computing framework. Comput & Fluids 2015;117:168–82.
[53] Tezduyar T. In: Hutchinson JW, Wu TY, editors. Stabilized finite element formulations for incompressible flow computations. Advances in applied mechanics, vol. 28, Elsevier; 1991, p. 1–44.
[54] Ahmed N, Rubino S. Numerical comparisons of finite element stabilized methods for a 2D vortex dynamics simulation at high Reynolds number. Comput Methods Appl Mech Engrg 2019;349:191–212.
[55] Maday Y, Nguyen NC, Patera TA, Pau SH. A general multipurpose interpolation procedure: the magic points. Commun Pure Appl Anal 2009;8:383.
[56] Farhat C, Grimberg S, Manzoni A, Quarteroni A. Computational bottlenecks for PROMs: Pre-computation and hyperreduction. In: Benner P, Grivet-Talocia S, Quarteroni A, Rozza G, Schilders W, Silveira L, editors. Model order reduction. Vol. 2: Snapshot-based methods and algorithms. Berlin: De Gruyter; 2020, p. 181–244.
[57] Alnæs MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes ME, Wells GN. The FEniCS project version 1.5. Arch Numer Softw 2015;3(100).
[58] Logg A, Mardal K-A, Wells GN, et al. Automated solution of differential equations by the finite element method. Springer; 2012.
[59] Peherstorfer B, Drmač Z, Gugercin S. Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points. SIAM J Sci Comput 2020;42(5):A2837–64.