

Article

Pressure Stabilization Strategies for a LES Filtering Reduced Order Model

Michele Girfoglio ¹, Annalisa Quaini ² and Gianluigi Rozza ^{1,*}
¹ Mathematics Area, mathLab, SISSA, International School for Advanced Studies, via Bonomea 265, 34136 Trieste, Italy; mgirfogl@sissa.it

² Department of Mathematics, University of Houston, Houston, TX 77204, USA; quaini@math.uh.edu

* Correspondence: grozza@sissa.it

Abstract: We present a stabilized POD–Galerkin reduced order method (ROM) for a Leray model. For the implementation of the model, we combine a two-step algorithm called Evolve-Filter (EF) with a computationally efficient finite volume method. In both steps of the EF algorithm, velocity and pressure fields are approximated using different POD basis and coefficients. To achieve pressure stabilization, we consider and compare two strategies: the pressure Poisson equation and the supremizer enrichment of the velocity space. We show that the evolve and filtered velocity spaces have to be enriched with the supremizer solutions related to both evolve and filter pressure fields in order to obtain stable and accurate solutions with the supremizer enrichment method. We test our ROM approach on a 2D unsteady flow past a cylinder at Reynolds number $0 \leq Re \leq 100$. We find that both stabilization strategies produce comparable errors in the reconstruction of the lift and drag coefficients, with the pressure Poisson equation method being more computationally efficient.

Keywords: large eddy simulation; reduced order modeling; pressure stabilization



Citation: Girfoglio, M.; Quaini, A.; Rozza, G. Pressure Stabilization Strategies for a LES Filtering Reduced Order Model. *Fluids* **2021**, *6*, 302. <https://doi.org/10.3390/fluids6090302>

Academic Editor: Robert Martinuzzi

Received: 28 June 2021

Accepted: 19 August 2021

Published: 25 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

For about a couple of decades, reduced order models (ROMs) have emerged as an efficient tool for the approximation of problems governed by parametrized partial differential equations. This success is owed to the fact that ROMs can significantly reduce the computational cost required by classical full order models (FOMs), e.g., finite element method or finite volume method, when several solutions associated to different parameter values are needed. The basic ROM framework consists of two steps. During a first phase (called *offline*), a database of several solutions is collected by solving a FOM of choice for different parameter values. Then, during a second phase (called *online*), the database collected in the offline phase is used to compute the solution for newly specified values of the parameters in a short amount of time. For a comprehensive review on ROMs, we refer the reader to, e.g., [1–8].

The particular ROM we consider in this paper is based on a POD–Galerkin approach, which consists in extracting of the most energetic modes representing the system dynamics and projecting the governing equations onto the space spanned by these modes. For the specific application that we target, i.e., an incompressible fluid flow at moderately high Reynolds number, and it is well known that standard POD–Galerkin models lead to instabilities [9,10]. A successful way to cure these instabilities in advection dominated flows is to adopt subgrid-scale closure models. See, e.g., [11,12]. Thus, we choose to work with a large eddy simulation (LES) approach.

We focus on a variant of the so-called Leray model [13], where the small-scale effects are described by a set of equations to be added to the discrete Navier–Stokes equations. This extra problem acts as a differential low-pass filter [14]. For its actual implementation, we use the Evolve-Filter (EF) algorithm [15–18]. One of the novelties of our approach is that we use a finite volume (FV) method [19,20], while the vast majority of the works on

Leray-type models use a finite element framework. Some of the reasons behind our interest in a FV approximation are the following: FV schemes are usually favored for practical, real-life applications; FV methods are considered robust, computationally efficient, and suitable when the conservation of the numerical flux is a crucial issue, as is the case in problems arising in fluid dynamics; nowadays many computational codes used in the industry, both commercial (e.g., CFX, FLUENT, STAR-CD) and open-source (e.g., OpenFOAM), are based on FV discretizations. Thus, with a FV approximation we hope to broaden the community of practitioners that use Leray-type models and ROMs. In the context of regularized ROMs, the Leray model and the EF algorithm have been thoroughly investigated. See, e.g., [21–25]. In all of these works, the filtering approach is only employed at reduced order level. In [26,27], for the first time the LES filtering is used also at the FOM level, i.e., to generate the snapshot data. Such an approach provides a ROM that is fully consistent with the FOM as the same mathematical framework is used during both the *offline* and *online* stages.

This paper could be seen as an extension of [26]. Therein, we used a pressure Poisson Equation (PPE) in the online stage [28–30] as a pressure stabilization technique. Here, we compare the PPE method with the supremizer enrichment of the velocity space [29,31–33], i.e., another technique that provides pressure stability. The main objective of this work is to test the accuracy and efficiency of these two methods within our LES filtering approach. We show that adapting the supremizer enrichment to the EF algorithm is not a trivial exercise. Indeed, the supremizer method becomes accurate only when the evolve and filtered velocity spaces are enriched by the supremizer solutions associated to both evolve and filter pressure fields. Since we are interested in the reconstruction of the pressure field at reduced order level, time is the only parameter we consider. We vary no physical and/or geometrical parameters. We test our framework on 2D flow past a cylinder with time-dependent Reynolds number $0 \leq Re(t) \leq 100$ [34,35].

The work is organized as follows. Section 2 describes the full order model and the numerical method we use for it. In Section 3, we describe the reduced order model. The numerical examples are reported in Section 4. Finally, Section 5 provides conclusions and perspectives.

2. The Full Order Model

In this section, we describe our Full Order Model (FOM). We consider a fixed domain $\Omega \subset \mathbb{R}^D$ with $D = 2, 3$ over a time interval of interest $(t_0, T) \subset \mathbb{R}^+$. The so-called *Leray model* couples the Navier–Stokes equations (NSE) with a differential filter as follows:

$$\rho \partial_t \mathbf{u} + \rho \nabla \cdot (\bar{\mathbf{u}} \otimes \mathbf{u}) - 2\mu \Delta \mathbf{u} + \nabla p = 0 \quad \text{in } \Omega \times (t_0, T), \quad (1)$$

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

$$-2\alpha^2 \Delta \bar{\mathbf{u}} + \bar{\mathbf{u}} + \nabla \lambda = \mathbf{u} \quad \text{in } \Omega \times (t_0, T), \quad (3)$$

$$\nabla \cdot \bar{\mathbf{u}} = 0 \quad \text{in } \Omega \times (t_0, T), \quad (4)$$

where ρ is the fluid density, μ is the dynamic viscosity, $\mathbf{u}$ is the fluid velocity, p is the fluid pressure, $\bar{\mathbf{u}}$ is the *filtered* velocity, and variable λ is a Lagrange multiplier to enforce the incompressibility constraint for $\bar{\mathbf{u}}$. Parameter α in Equation (3) can be interpreted as a *filtering radius* (that is, the radius of the neighborhood where the filter extracts information from the unresolved scales). In this paper, α will be constant in space and time. More sophisticated choices are possible, but will not be considered here. Problem (1)–(4) is endowed with suitable boundary conditions

$$\mathbf{u} = \mathbf{u}_D \quad \text{on } \partial\Omega_D \times (t_0, T), \quad (5)$$

$$(2\mu \nabla \mathbf{u} - p \mathbf{I}) \mathbf{n} = \mathbf{0} \quad \text{on } \partial\Omega_N \times (t_0, T), \quad (6)$$

$$\bar{\mathbf{u}} = \mathbf{u}_D \quad \text{on } \partial\Omega_D \times (t_0, T), \quad (7)$$

$$(2\alpha^2 \nabla \bar{\mathbf{u}} - \lambda \mathbf{I}) \mathbf{n} = \mathbf{0} \quad \text{on } \partial\Omega_N \times (t_0, T). \quad (8)$$

and the initial data $\mathbf{u} = \mathbf{u}_0$ in $\Omega \times \{t_0\}$. Here, $\overline{\partial\Omega_D} \cup \overline{\partial\Omega_N} = \overline{\partial\Omega}$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. In addition, $\mathbf{u}_D$ and $\mathbf{u}_0$ are given. Note that we restrict our attention to homogeneous Neumann boundary conditions. Of course, the methodology we propose can be extended to non-homogeneous Neumann conditions, as well as other boundary conditions (e.g., Robin conditions) [36].

2.1. The Evolve-Filter Algorithm

First, we address the time discretization of model (1)–(4). Let $\Delta t \in \mathbb{R}$, $t^n = t_0 + n\Delta t$, with $n = 0, \dots, N_T$ and $T = t_0 + N_T\Delta t$, and let y^n denote the approximation of a generic quantity y at the time t^n .

With the aim of decoupling the Navier–Stokes system (1) and (2) from the filter system (3) and (4) at each time step, we adopt the Evolve-Filter (EF) algorithm [15–17]. This algorithm reads as follows: given $\mathbf{u}^{n-1}$ and $\mathbf{u}^n$, at time t^{n+1} :

- *Evolve*: find intermediate velocity and pressure $(\mathbf{v}^{n+1}, q^{n+1})$ such that

$$\rho \frac{3}{2\Delta t} \mathbf{v}^{n+1} + \rho \nabla \cdot (\mathbf{u}^* \otimes \mathbf{v}^{n+1}) - 2\mu \Delta \mathbf{v}^{n+1} + \nabla q^{n+1} = \mathbf{b}^{n+1}, \quad (9)$$

$$\nabla \cdot \mathbf{v}^{n+1} = 0, \quad (10)$$

with boundary conditions

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

$$(2\mu \nabla \mathbf{v}^{n+1} - q^{n+1} \mathbf{I}) \mathbf{n} = \mathbf{0} \quad \text{on } \partial\Omega_N \times (t_0, T), \quad (12)$$

and initial condition $\mathbf{v}^0 = \mathbf{u}_0$ in $\Omega \times \{t_0\}$. In Equation (9), we set $\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ and $\mathbf{b}^{n+1} = (4\mathbf{u}^n - \mathbf{u}^{n-1}) / (2\Delta t)$. Moreover, we approximated the Eulerian time derivative with a Backward Differentiation Formula of order 2 (BDF2) [37].

- *Filter*: find $(\mathbf{u}^{n+1}, \lambda^{n+1})$ such that

$$-\alpha^2 \Delta \mathbf{u}^{n+1} + \mathbf{u}^{n+1} + \nabla \lambda^{n+1} = \mathbf{v}^{n+1}, \quad (13)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (14)$$

with boundary conditions

$$\mathbf{u}^{n+1} = \mathbf{u}_D^{n+1} \quad \text{on } \partial\Omega_D \times (t_0, T), \quad (15)$$

$$(2\alpha^2 \nabla \mathbf{u}^{n+1} - \lambda^{n+1} \mathbf{I}) \mathbf{n} = \mathbf{0} \quad \text{on } \partial\Omega_N \times (t_0, T). \quad (16)$$

So, we accept $\mathbf{u}^{n+1}$ and q^{n+1} as the approximation of the fluid velocity and pressure at the time t^{n+1} , respectively.

Remark 1. Problem (13)–(14) can be seen as a generalized Stokes problem. In fact, by multiplying Equation (13) by $\rho / \Delta t$ and with a rearrangement of the terms we get:

$$\frac{\rho}{\Delta t} \mathbf{u}^{n+1} - \bar{\mu} \Delta \mathbf{u}^{n+1} + \nabla \bar{q}^{n+1} = \frac{\rho}{\Delta t} \mathbf{v}^{n+1}, \quad \bar{\mu} = \rho \frac{\alpha^2}{\Delta t}, \quad (17)$$

where variable $\bar{q}^{n+1} = \rho \lambda^{n+1} / \Delta t$ can be regarded as a filtered pressure. Problem (17), (14) can be seen as a time dependent Stokes problem with viscosity $\bar{\mu}$ and the Eulerian time derivative approximated by the Backward Euler (or BDF1) scheme. Thus, problem (17) and (14) can be solved by adapting a standard linearized Navier–Stokes solver.

Remark 2. It is known that the EF algorithm is over-diffusive, also when combined with a FV method. In [19], we showed that the Evolve-Filter-Relax algorithm [18] with nonlinear differential filters is a better choice, especially for realistic problems. Both the EF and the EFR algorithm require pressure stabilization. Thus, in this manuscript, we prefer to focus on the EF algorithm, which

does not have the complication of non-linearity. For the application of our ROM approach with pressure stabilization to nonlinear LES filtering and associated assessment for 2D and 3D problems, the reader is referred to [27].

2.2. Space Discrete Problem by a Finite Volume Method

Concerning the space discretization, we use a FV method for both problem (9) and (10) and problem (17) and (14). We partition the computational domain Ω into cells or control volumes Ω_i , with $i = 1, \dots, N_c$, where N_c is the total number of cells in the mesh. Let $\mathbf{A}_j$ be the surface vector of each face of the control volume (i.e., a vector that is perpendicular to the surface and has magnitude equal to the surface area), with $j = 1, \dots, M$.

Once fully discretized, the *Evolve* problem (9) and (10) reads

$$\rho \frac{3}{2\Delta t} \mathbf{v}_i^{n+1} + \rho \sum_j \varphi_j^* \mathbf{v}_{i,j}^{n+1} - 2\mu \sum_j (\nabla \mathbf{v}_i^{n+1})_j \cdot \mathbf{A}_j + \sum_j q_{i,j}^{n+1} \mathbf{A}_j = \mathbf{b}_i^{n+1}, \quad (18)$$

$$\sum_j (\nabla q^{n+1})_j \cdot \mathbf{A}_j = \sum_j (\mathbf{H}(\mathbf{v}_i^{n+1}))_j \cdot \mathbf{A}_j, \quad (19)$$

with $\varphi_j^* = \mathbf{u}_j^* \cdot \mathbf{A}_j$ and

$$\mathbf{H}(\mathbf{v}_i^{n+1}) = -\rho \sum_j \varphi_j^* \mathbf{v}_{i,j}^{n+1} + 2\mu \sum_j (\nabla \mathbf{v}_i^{n+1})_j \cdot \mathbf{A}_j + \mathbf{b}_i^{n+1}. \quad (20)$$

In (18)–(20), $\mathbf{v}_i^{n+1}$ and $\mathbf{b}_i^{n+1}$ denote the average velocity and source term in control volume Ω_i , respectively. Moreover, we denote with $\mathbf{v}_{i,j}^{n+1}$ and $q_{i,j}^{n+1}$ the velocity and pressure associated to the centroid of face j normalized by the volume of Ω_i .

As for the filter problem (17) and (14), once fully discretized it reads:

$$\frac{\rho}{\Delta t} \mathbf{u}_i^{n+1} - \bar{\mu} \sum_j (\nabla \mathbf{u}_i^{n+1})_j \cdot \mathbf{A}_j + \sum_j \bar{q}_{i,j}^{n+1} \mathbf{A}_j = \frac{\rho}{\Delta t} \mathbf{v}_i^{n+1}, \quad (21)$$

$$\sum_j (\nabla \bar{q}_i^{n+1})_j \cdot \mathbf{A}_j = \sum_j (\bar{\mathbf{H}}(\mathbf{u}_i^{n+1}))_j \cdot \mathbf{A}_j, \quad (22)$$

with

$$\bar{\mathbf{H}}(\mathbf{u}_i^{n+1}) = \bar{\mu} \sum_j (\nabla \mathbf{u}_i^{n+1})_j \cdot \mathbf{A}_j + \frac{\rho}{\Delta t} \mathbf{v}_i^{n+1}. \quad (23)$$

In (21)–(23), we denoted with $\mathbf{u}_i^{n+1}$ the average filtered velocity in control volume Ω_i , while $\bar{q}_{i,j}^{n+1}$ is the auxiliary pressure at the centroid of face j normalized by the volume of Ω_i .

More details on the full discretization of both problems (18)–(20) and (21)–(23) are available in [19].

The EF algorithm has been implemented within the C++ finite volume library OpenFOAM® [38]. The linear system related to the problem (18) and (19) has been solved by using the PISO algorithm [39]. On the other hand, for problem (21) and (22) the SIMPLEC algorithm [40] (i.e., a slightly modified version of the SIMPLE algorithm [41]) has been chosen. We remark that both PISO and SIMPLEC are partitioned algorithms that perform a decoupling of the computation of the pressure from the computation of the velocity.

3. The Reduced Order Model

The Reduced Order Model (ROM) we propose draws inspiration from the framework introduced in [26]. In Section 3.1, we provide a description of the procedure used to build a POD–Galerkin ROM, whilst in Section 3.2, we propose two different strategies for pressure

stabilization at reduced order level. Finally, Section 3.3 is focused on the method we apply to impose non-homogeneous Dirichlet boundary conditions (5) and (7) at the reduced order level. All the ROM computations are performed using ITHACA-FV [42], an in-house open source C++ library.

3.1. A POD–Galerkin Projection Method

We assume that the velocity fields $\mathbf{v}$ and $\mathbf{u}$ and pressure fields q and $\bar{q}$ can be approximated by linear combinations of the dominant modes (basis functions) that are dependent on space variables only and multiplied by scalar coefficients that depend on the time:

$$\mathbf{v} \approx \mathbf{v}_r = \sum_{i=1}^{N_{v_r}} \beta_i(t) \boldsymbol{\varphi}_i(\mathbf{x}), \quad q \approx q_r = \sum_{i=1}^{N_{q_r}} \gamma_i(t) \psi_i(\mathbf{x}), \quad (24)$$

$$\mathbf{u} \approx \mathbf{u}_r = \sum_{i=1}^{N_{u_r}} \bar{\beta}_i(t) \bar{\boldsymbol{\varphi}}_i(\mathbf{x}), \quad \bar{q} \approx \bar{q}_r = \sum_{i=1}^{N_{\bar{q}_r}} \bar{\gamma}_i(t) \bar{\psi}_i(\mathbf{x}). \quad (25)$$

In (24) and (25), N_{Φ_r} denotes the cardinality of a reduced basis for the space field Φ belongs to. Note that Φ could be either a scalar or a vector field. Using (24) to approximate $\mathbf{v}^{n+1}$ and q^{n+1} in (9) and (10), we obtain

$$\rho \frac{3}{2\Delta t} \mathbf{v}_r^{n+1} + \rho \nabla \cdot (\mathbf{u}_r^* \otimes \mathbf{v}_r^{n+1}) - 2\mu \Delta \mathbf{v}_r^{n+1} + \nabla q_r^{n+1} = \mathbf{b}_r^{n+1}, \quad (26)$$

$$\nabla \cdot \mathbf{v}_r^{n+1} = 0, \quad (27)$$

where $\mathbf{u}_r^* = 2\mathbf{u}_r^n - \mathbf{u}_r^{n-1}$ and $\mathbf{b}_r^{n+1} = (4\mathbf{u}_r^n - \mathbf{u}_r^{n-1}) / (2\Delta t)$. Then, using (25) to approximate $\mathbf{u}^{n+1}$ and $\bar{q}^{n+1}$ in (17) and (14) we get:

$$\frac{\rho}{\Delta t} \mathbf{u}_r^{n+1} - \bar{\mu} \Delta \mathbf{u}_r^{n+1} + \nabla \bar{q}_r^{n+1} = \frac{\rho}{\Delta t} \mathbf{v}_r^{n+1}, \quad (28)$$

$$\nabla \cdot \mathbf{u}_r^{n+1} = 0. \quad (29)$$

The reader is referred, e.g., to [2,6,43–48] for an overview of several techniques proposed in the literature to generate the reduced basis spaces. Among these, we mention the Proper Orthogonal Decomposition (POD), the Proper Generalized Decomposition (PGD) and the Reduced Basis (RB) with a greedy sampling strategy. We opt for the method of snapshots. To this aim, we solve the FOM described in Section 2 for each time $t^k \in \{t^1, \dots, t^{N_s}\} \subset (t_0, T]$. The snapshots matrices are obtained from the full-order snapshots:

$$\mathcal{S}_{\Phi} = [\Phi(t^1), \dots, \Phi(t^{N_s})] \in \mathbb{R}^{N_{\Phi_h} \times N_s} \quad \text{for } \Phi = \{\mathbf{v}, \mathbf{u}, q, \bar{q}\}, \quad (30)$$

where the subscript h denotes a solution computed with the FOM and N_{Φ_h} is the dimension of the space field Φ in the FOM. Then, for each value of the dimension of the POD space $N_{POD} = 1, \dots, N_s$ one has to find the scalar coefficients $a_1^1, \dots, a_1^{N_s}, \dots, a_{N_s}^1, \dots, a_{N_s}^{N_s}$ and functions $\zeta_1, \dots, \zeta_{N_s}$ that minimize the error between the snapshots and their projection onto the POD basis. In the L^2 -norm, we have

$$E_{N_{POD}} = \arg \min \sum_{i=1}^{N_s} \|\Phi_i - \sum_{k=1}^{N_{POD}} a_i^k \zeta_k\| \quad \forall N_{POD} = 1, \dots, N_s$$

with $(\zeta_i, \zeta_j)_{L_2(\Omega)} = \delta_{i,j} \quad \forall i, j = 1, \dots, N_s.$ (31)

Problem (31) is equivalent to the following eigenvalue problem:

$$\begin{aligned} \mathcal{C}^\Phi Q^\Phi &= Q^\Phi \Lambda^\Phi, \\ \mathcal{C}_{ij}^\Phi &= (\Phi(t^i), \Phi(t^j))_{L_2(\Omega)} \quad \text{for } i, j = 1, \dots, N_s, \end{aligned} \quad (32)$$

where $\mathcal{C}^\Phi$ is the correlation matrix computed from the snapshot matrix $\mathcal{S}_\Phi$, Q^Φ is the matrix of eigenvectors and Λ^Φ is a diagonal matrix whose diagonal entries are the eigenvalues of $\mathcal{C}^\Phi$ [49]. Once problem (32) is solved, the basis functions are obtained as follows:

$$\zeta_i = \frac{1}{N_s \Lambda_i^\Phi} \sum_{j=1}^{N_s} \Phi_j Q_{ij}^\Phi, \quad (33)$$

and the POD modes are:

$$L_\Phi = [\zeta_1, \dots, \zeta_{N_{\Phi_r}}] \in \mathbb{R}^{N_{\Phi_h} \times N_{\Phi_r}}, \quad (34)$$

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

Let

$$M_{rij} = (\varphi_i, \varphi_j)_{L_2(\Omega)}, \quad \tilde{M}_{rij} = (\varphi_i, \overline{\varphi_j})_{L_2(\Omega)}, \quad A_{rij} = (\varphi_i, \Delta \varphi_j)_{L_2(\Omega)}, \quad (35)$$

$$B_{rij} = (\varphi_i, \nabla \psi_j)_{L_2(\Omega)}, \quad P_{rij} = (\psi_i, \nabla \cdot \varphi_j)_{L_2(\Omega)}, \quad (36)$$

where φ_i and ψ_i are the basis functions in (24). At time t^{n+1} , the reduced algebraic system for problem (26) and (27) is:

$$\rho \frac{3}{2\Delta t} M_r \beta^{n+1} + \rho G_r(\bar{\beta}^n, \bar{\beta}^{n-1}) \beta^{n+1} - 2\mu A_r \beta^{n+1} + B_r \gamma^{n+1} = \frac{\rho}{\Delta t} \tilde{M}_r \left(2\bar{\beta}^n - \frac{1}{2}\bar{\beta}^{n-1} \right), \quad (37)$$

$$P_r \beta^{n+1} = 0, \quad (38)$$

where vectors β^{n+1} and γ^{n+1} contain the values of coefficients β_i and γ_i in (24) at time t^{n+1} . The term $G_r(\bar{\beta}^n, \bar{\beta}^{n-1}) \beta^{n+1}$ in (37) is related to the non-linear convective term:

$$\left(G_r(\bar{\beta}^n, \bar{\beta}^{n-1}) \beta^{n+1} \right)_i = (2\bar{\beta}^n - \bar{\beta}^{n-1})^T \mathcal{G}_{r_i..} \beta^{n+1} \quad (39)$$

where $\mathcal{G}_r$ is a third-order tensor defined as follows

$$\mathcal{G}_{r_{ijk}} = (\varphi_i, \nabla \cdot (\varphi_j \otimes \overline{\varphi_k}))_{L_2(\Omega)}. \quad (40)$$

See [37,50] for more details.

In order to write the reduced algebraic system for problem (28) and (29), let

$$\overline{M}_{rij} = (\overline{\varphi}_i, \overline{\varphi}_j)_{L_2(\Omega)}, \quad \overline{A}_{rij} = (\overline{\varphi}_i, \Delta \overline{\varphi}_j)_{L_2(\Omega)}, \quad (41)$$

$$\overline{B}_{rij} = (\overline{\varphi}_i, \nabla \overline{\psi}_j)_{L_2(\Omega)}, \quad \overline{P}_{rij} = (\overline{\psi}_i, \nabla \cdot \overline{\varphi}_j)_{L_2(\Omega)}, \quad (42)$$

where $\overline{\varphi}_i$ and $\overline{\psi}_i$ are the basis functions in (25). Such system at time t^{n+1} is

$$\frac{\rho}{\Delta t} \overline{M}_r \bar{\beta}^{n+1} - \bar{\mu} \overline{A}_r \bar{\beta}^{n+1} + \overline{B}_r \bar{\gamma}^{n+1} = \frac{\rho}{\Delta t} \overline{M}_r^T \beta^{n+1}, \quad (43)$$

$$\overline{P}_r \bar{\beta}^{n+1} = 0, \quad (44)$$

where vectors $\bar{\beta}^{n+1}$ and $\bar{\gamma}^{n+1}$ contain the values of coefficients $\bar{\beta}_i$ and $\bar{\gamma}_i$ in (25) at time t^{n+1} .

Finally, we obtain the initial conditions for the ROM algebraic system (37) and (38) by performing a Galerkin projection of the initial full order condition onto the POD basis spaces:

$$\beta_i^0 = (v(x, t_0), \varphi_i)_{L_2(\Omega)}, \quad \bar{\beta}_i^0 = (u(x, t_0), \bar{\varphi}_i)_{L_2(\Omega)}.$$

The complete reduced algebraic system at time t^{n+1} is given by (37)–(38) and (43)–(44).

3.2. Pressure Fields Reconstruction and Pressure Stability

It is well known that the reduced problem (37)–(38) and (43)–(44) presents stability issues because the approximation spaces need to satisfy the inf-sup (Ladyzhenskaya-Brezzi-Babuska) condition [51,52]. In a standard finite element (FE) NSE framework, the inf-sup condition reads:

$$\inf_{q_h \in Q_h} \sup_{u_h \in V_h} \frac{(\nabla \cdot u_h, q_h)}{\|\nabla \cdot u_h\| \|q_h\|} \geq \gamma > 0, \quad (45)$$

where Q_h is the FE space for the pressure approximation, V_h is the FE space for the approximation of the velocity field, and γ is a constant that does not depend on the mesh size h . In order to obtain a stable and accurate reconstruction of the pressure field at the reduced level, different approaches have been proposed. One option is to use a global POD basis for both pressure and velocity field and same temporal coefficients [53,54]. Another option is represented by the supremizer enrichment method; see, e.g., [29,31–33]. Finally, one can take the divergence of the momentum equation to obtain a Poisson equation for the pressure that is projected onto a POD basis; see, e.g., [28,29]. This third method is called Poisson pressure Equation (PPE).

In this work, we test and compare the performances of two methods: the PPE method (already combined with the EF algorithm in [26]) and the supremizer enrichment method (not yet tested for the EF algorithm). As we will show, the extension of the supremizer enrichment method to the EF algorithm is not straightforward.

3.2.1. Pressure Poisson Equation Method

We take the divergence of Equation (9) and account for conditions (10) to obtain the Poisson pressure equation for the Evolve step:

$$\Delta q^{n+1} = -\rho \nabla \cdot \left(\nabla \cdot \left(u^* \otimes v^{n+1} \right) \right). \quad (46)$$

So, at the Evolve step instead of solving (9) and (10), we solve the modified systems (9), (46) with boundary condition (11) and

$$\partial_n q^{n+1} = -2\mu n \cdot \left(\nabla \times \nabla \times v^{n+1} \right) - n \cdot \left(\rho \frac{3}{2\Delta t} v^{n+1} - b^{n+1} \right) \quad \text{on } \partial\Omega_N \times (t_0, T), \quad (47)$$

where ∂_n denotes the derivative with respect to outgoing normal n .

We proceed similarly for the Filter step. We take the divergence of Equation (17) and account for condition (14) to obtain the Poisson pressure equation:

$$\Delta \bar{q}^{n+1} = 0. \quad (48)$$

The Filter step becomes solving (17), (48) with boundary condition (15) and

$$\partial_n \bar{q}^{n+1} = -2\bar{\mu} n \cdot \left(\nabla \times \nabla \times u^{n+1} \right) \quad \text{on } \partial\Omega_N \times (t_0, T). \quad (49)$$

The reader interested in enforcing non-homogeneous Neumann conditions for the pressure fields is referred to [55,56]. We remark that systems (9) and (10) and (17) and (14) are not equivalent to systems (9) and (46) and (17) and (48) for steady flows [55–57].

Using (24) to approximate v^{n+1} and q^{n+1} in (46), we obtain

$$\Delta q_r^{n+1} = -\rho \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_r^* \otimes \mathbf{v}_r^{n+1} \right) \right). \quad (50)$$

After space discretization, the matrix form of Equation (50) reads:

$$\mathbf{D}_r \gamma^{n+1} + \rho \mathbf{J}_r(\bar{\boldsymbol{\beta}}^n, \bar{\boldsymbol{\beta}}^{n-1}) \boldsymbol{\beta}^{n+1} - 2\mu \mathbf{N}_r \boldsymbol{\beta}^{n+1} - \frac{\rho}{2\Delta t} \left(3\mathbf{F}_r \boldsymbol{\beta}^{n+1} - 4\bar{\mathbf{F}}_r \boldsymbol{\beta}^n + \bar{\mathbf{F}}_r \boldsymbol{\beta}^{n-1} \right) = 0, \quad (51)$$

where

$$\mathbf{D}_{rij} = (\nabla \psi_i, \nabla \psi_j)_{L_2(\Omega)}, \quad \mathbf{N}_{rij} = (\mathbf{n} \times \nabla \psi_i, \nabla \times \boldsymbol{\varphi}_j)_{L_2(\partial\Omega)}, \quad (52)$$

$$\mathbf{F}_{rij} = (\psi_i, \mathbf{n} \cdot \boldsymbol{\varphi}_j)_{L_2(\partial\Omega)}, \quad \bar{\mathbf{F}}_{rij} = (\psi_i, \mathbf{n} \cdot \bar{\boldsymbol{\varphi}}_j)_{L_2(\partial\Omega)}. \quad (53)$$

The residual associated with the non-linear term in Equation (51) is evaluated with the same strategy used for Equation (37). We have

$$\left(\mathbf{J}_r(\bar{\boldsymbol{\beta}}^n, \bar{\boldsymbol{\beta}}^{n-1}) \boldsymbol{\beta}^{n+1} \right)_i = (2\bar{\boldsymbol{\beta}}^n - \bar{\boldsymbol{\beta}}^{n-1})^T \mathcal{J}_{r_{i..}} \boldsymbol{\beta}^{n+1}, \quad (54)$$

where $\mathcal{J}_r$ is a third-order tensor defined as follows

$$\mathcal{J}_{r_{ijk}} = (\nabla \psi_i, \nabla \cdot (\boldsymbol{\varphi}_j \otimes \bar{\boldsymbol{\varphi}}_k))_{L_2(\Omega)}. \quad (55)$$

Using (25) to approximate $\bar{q}^{n+1}$ in Equation (48), we get

$$\Delta \bar{q}_r^{n+1} = 0. \quad (56)$$

Once discretized in space, Equation (56) can be written in the matrix form as:

$$\bar{\mathbf{D}}_r \bar{\gamma}^{n+1} - 2\bar{\mu} \bar{\mathbf{N}}_r \bar{\boldsymbol{\beta}}^{n+1} = 0, \quad (57)$$

where

$$\bar{\mathbf{D}}_{rij} = (\nabla \bar{\psi}_i, \nabla \bar{\psi}_j)_{L_2(\Omega)}, \quad \bar{\mathbf{N}}_{rij} = (\mathbf{n} \times \nabla \bar{\psi}_i, \nabla \times \bar{\boldsymbol{\varphi}}_j)_{L_2(\partial\Omega)}. \quad (58)$$

With the PPE method, at every time step the ROM algebraic system that has to be solved is (37), (51), (43), (57).

3.2.2. Supremizer Enrichment Method

For a given pressure basis function, the supremizer is the solution that permits the realization of the inf-sup condition (45). One has to find the supremizer for each pressure basis function. Here, we use an approximated supremizer enrichment procedure: instead of pressure basis functions we use pressure snapshots. This procedure allows to drastically reduce the *online* computational cost. In fact, the supremizer basis functions do not depend on the particular pressure basis functions, but are computed directly from the pressure snapshots during the *offline* phase. The downside of this approximated procedure is that it is not possible to rigorously show that the inf-sup condition is satisfied. One only relies on heuristic criteria or checks during a post-processing stage.

For NSE, the supremizer $s(t^i)$ relative to pressure snapshot $q(t^i)$ is found by solving the following problem:

$$\Delta s(t^i) = -\nabla q(t^i) \quad \text{in } \Omega, \quad (59)$$

$$s(t^i) = 0 \quad \text{on } \partial\Omega, \quad (60)$$

for $i = 1, \dots, N_s$. For more details, the reader is referred to [29,31–33]. For the EF algorithm, in addition to (59) and (60) one has to solve an analogous problem for each auxiliary pressure snapshot $\bar{q}(t^i)$:

$$\Delta \bar{s}(t^i) = -\nabla \bar{q}(t^i) \quad \text{in } \Omega, \quad (61)$$

$$\bar{s}(t^i) = 0 \quad \text{on } \partial\Omega, \quad (62)$$

for $i = 1, \dots, N_s$. Once problems (59) and (60) and (61) and (62) are solved, two snapshots matrices of supremizer solutions are assembled:

$$\mathcal{S}_s = [s(t^1), \dots, s(t^{N_s})] \in \mathbb{R}^{N_{v_h} \times N_s}, \quad \mathcal{S}_{\bar{s}} = [\bar{s}(t^1), \dots, \bar{s}(t^{N_s})] \in \mathbb{R}^{N_{u_h} \times N_s}.$$

A POD procedure is applied to the matrices above in order to obtain the supremizer POD basis functions:

$$L_s = [\chi_1, \dots, \chi_{N_{s_r}}] \in \mathbb{R}^{N_{v_h} \times N_{s_r}}, \quad L_{\bar{s}} = [\bar{\chi}_1, \dots, \bar{\chi}_{N_{\bar{s}_r}}] \in \mathbb{R}^{N_{u_h} \times N_{\bar{s}_r}},$$

where N_{s_r} and $N_{\bar{s}_r}$ conform to the notation introduced in Section 3.1. These velocity supremizer basis functions are added to the reduced velocity spaces, which become:

$$\tilde{L}_v = [\varphi_1, \dots, \varphi_{N_{v_r}}, \chi_1, \dots, \chi_{N_{s_r}}] \in \mathbb{R}^{N_{v_h} \times (N_{v_r} + N_{s_r})}, \quad (63)$$

$$\tilde{L}_u = [\bar{\varphi}_1, \dots, \bar{\varphi}_{N_{u_r}}, \bar{\chi}_1, \dots, \bar{\chi}_{N_{\bar{s}_r}}] \in \mathbb{R}^{N_{u_h} \times (N_{u_r} + N_{\bar{s}_r})}, \quad (64)$$

Let us call SUP₁ the supremizer enrichment method described above. In Section 4, we will show that SUP₁ does not lead to an accurate reconstruction of the pressure fields. To increase the accuracy, we add the supremizer solutions associated to both pressure fields to the evolve and filtered velocity spaces, i.e.:

$$\tilde{\tilde{L}}_v = [\varphi_1, \dots, \varphi_{N_{v_r}}, \chi_1, \dots, \chi_{N_{s_r}}, \bar{\chi}_1, \dots, \bar{\chi}_{N_{\bar{s}_r}}] \in \mathbb{R}^{N_{v_h} \times (N_{v_r} + N_{s_r} + N_{\bar{s}_r})}, \quad (65)$$

$$\tilde{\tilde{L}}_u = [\bar{\varphi}_1, \dots, \bar{\varphi}_{N_{u_r}}, \bar{\chi}_1, \dots, \bar{\chi}_{N_{\bar{s}_r}}, \chi_1, \dots, \chi_{N_{s_r}}] \in \mathbb{R}^{N_{u_h} \times (N_{u_r} + N_{\bar{s}_r} + N_{s_r})}. \quad (66)$$

We call SUP₂ the realization of the supremizer enrichment.

With either SUP₁ or SUP₂, the ROM algebraic system that has to be solved at every time step is (37), (38), (43), (44).

3.3. Implementation of Dirichlet Boundary Conditions: The Lifting Function Method

We apply the lifting function method to homogenize the velocity fields snapshots, as well as to make them independent of the boundary conditions [29]. We notice that the lifting functions are problem-dependent, so they have to be divergence free in order to retain the divergence-free property of the basis functions and they have to satisfy the boundary conditions of the FOM.

The velocity snapshots are modified as follows:

$$v'_h = v_h - \sum_{j=1}^{N_{BC}} v_{BC_j}(t) \chi_j(x), \quad u'_h = u_h - \sum_{j=1}^{N_{BC}} u_{BC_j}(t) \chi_j(x),$$

where N_{BC} is the number of non-homogeneous Dirichlet boundary conditions, $\chi_j(x)$ are the lifting functions, and v_{BC_j} and u_{BC_j} are suitable temporal coefficients. We apply the POD procedure described in Section 3.1 to the snapshots satisfying the homogeneous boundary conditions. Then, the boundary value is added back in this way:

$$\mathbf{v}_r = \sum_{j=1}^{N_{BC}} v_{BCj}(t) \chi_j(\mathbf{x}) + \sum_{i=1}^{N_{vr}} \beta_i(t) \boldsymbol{\varphi}_i(\mathbf{x}), \quad \mathbf{u}_r = \sum_{j=1}^{N_{BC}} u_{BCj}(t) \chi_j(\mathbf{x}) + \sum_{i=1}^{N_{ur}} \bar{\beta}_i(t) \bar{\boldsymbol{\varphi}}_i(\mathbf{x}).$$

4. Numerical Results

We consider 2D flow past a cylinder [34,35]. This well-known benchmark will allow us to assess the ability of the ROM approaches presented in Section 3 to reconstruct the time evolution of the velocity and pressure fields. We have thoroughly investigated the 2D flow past a cylinder with a finite volume FOM in [19] and with a ROM using a PPE approach in [26,27]. Here, we aim at comparing the supremizer enrichment method and the PPE method in terms of errors and speed-up.

We consider a 2.2×0.41 rectangular channel. Let the origin of the axes be at the bottom left corner of the channel. Inside the channel, there is a cylinder of radius 0.05 and center is located at (0.2, 0.2). The fluid filling the channel has density $\rho = 1$ and viscosity $\mu = 10^{-3}$. We enforce a no slip boundary condition on the upper and lower walls and on the cylinder. On the other hand at the inflow, we consider the following velocity profile:

$$\mathbf{v}(0, y, t) = \left(\frac{6}{0.41^2} \sin(\pi t/8) y(0.41 - y), 0 \right), \quad y \in [0, 0.41], \quad t \in (0, 8], \quad (67)$$

and $\partial q / \partial \mathbf{n} = \partial \bar{q} / \partial \mathbf{n} = 0$. Finally, at the outflow we enforce $\nabla \mathbf{v} \cdot \mathbf{n} = 0$ and $q = \bar{q} = 0$. All the simulations run starting from the condition of fluid at rest. It should be noted that the Reynolds number is time dependent, with $0 \leq Re(t) \leq 100$ [34].

We use a hexaedral computational grid with $h_{min} = 4.2 \times 10^{-3}$ and $h_{max} = 1.1 \times 10^{-2}$ for a total of 1.59×10^4 cells. The mesh quality is high: it is characterized by very low values of maximum non-orthogonality (36°), average non-orthogonality (4°), skewness (0.7), and maximum aspect ratio (2). Figure 1 shows a part of the mesh. We note that this mesh is the coarsest among all the meshes investigated in [19]. Therefore, it is the most challenging for our LES approach. We set $\Delta t = 4 \times 10^{-4}$, which allows us to obtain a maximum Courant–Friedrichs–Lewy number $CFL_{max} = 0.2$ at the time when the velocity reaches its maximum value for both FOM and ROM simulations. Concerning the convective term, we adopt a second-order accurate Central Differencing scheme [58]. With this choice, we do not have to introduce artificial stabilization. So, the effect of the filter can be assessed.

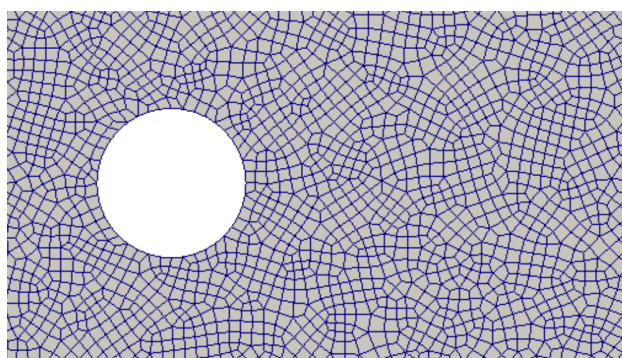


Figure 1. Part of the mesh under consideration.

We need to enforce a time-dependent non-uniform Dirichlet boundary condition at the inlet. For this purpose, we consider a divergence free function with the following non-uniform velocity distribution:

$$\chi(0, y) = \left(\frac{6}{0.41^2} y(0.41 - y), 0 \right), \quad y \in [0, 2.2], \quad (68)$$

and uniform null values on the rest of the boundary.

We will compare our findings with those in [29]. The choice of Ref. [29] is due to the fact that therein the authors develop a NSE-ROM finite volume framework both with PPE and supremizer enrichment methods for the reconstruction of the pressure field.

We set $\alpha = 0.0032$ and refer to [19] for details on this choice. The snapshots are collected every 0.1 (i.e., $N_s = 80$) using an equispaced grid method in time. Therefore, the dimension of the correlation matrix $\mathcal{C}^\Phi$ in (32) is 80×80 . For a convergence test as the number of snapshots increases, see [26]. Therein, we show that there is no substantial difference in the errors for the different sampling frequencies. Table 1 reports the first four cumulative eigenvalues for the velocity, pressure, and supremizer fields, based on (i.e., scaled with respect to the sum of) the first 15 most energetic POD modes. In order to retain 99.99% of the energy for the ROM, we need 2 modes for v , 2 modes for q , 2 modes for u , and 1 mode for $\bar{q}$. As for SUP₁, we consider a number of supremizer modes greater than pressure modes as suggested in [29]: four modes for s and three modes for $\bar{s}$. On the other hand, for SUP₂ we take into account an equal number of pressure and supremizer modes: two modes for s and one mode for $\bar{s}$.

Table 1. First four cumulative eigenvalues for the velocity, pressure and supremizer fields.

N Modes	u	v	q	$\bar{q}$	s	$\bar{s}$
1	0.999588	0.999582	0.967431	0.999985	0.736795	0.999899
2	0.999924	0.999924	0.999916	0.999997	0.999594	0.999986
3	0.999998	0.999998	0.999995	0.999999	0.999977	0.999999
4	0.999999	0.999999	0.999998	0.999995	0.999988	0.999999

We calculate the L^2 relative error:

$$E_\Phi = \frac{\|\Phi_h(t) - \Phi_r(t)\|_{L^2(\Omega)}}{\|\Phi_h(t)\|_{L^2(\Omega)}}, \quad (69)$$

where Φ_h and Φ_r are the FOM approximation of a given field (i.e., v_h , u_h , q_h or $\bar{q}_h$) and the corresponding ROM approximation (i.e., v_r , u_r , q_r or $\bar{q}_r$), respectively. Figure 2 shows error (69) for the two velocity and pressure fields over time for the three different ROM techniques under investigation: PPE, SUP₁ and SUP₂. We also present the errors related to the ROM computations with no stabilization technique, referred to as NOS. As one would expect, Figure 2 shows that the model with no stabilization is completely unreliable. From Figure 2, we see that the SUP₁-ROM is a big improvement over NOS-ROM. However, while the errors for the velocity fields are acceptable, the pressure errors remain large and far above the values obtained with NSE in [29]. This shows that SUP₁-ROM is not a reliable stabilization of the ROM for the EF algorithm. The SUP₂-ROM produces much better results in terms of the pressure fields, with errors for the velocity fields that are comparable with those given by the SUP₁-ROM. We speculate that the better performance of the SUP₂ model (which adds the supremizer solutions related to *both* pressure fields to the evolve and filtered velocity spaces) with respect to the SUP₁ model could be due to the strong coupling between the evolve velocity v and the filter velocity u ; however, the stability of the inf-sup ROM formulation for the EF algorithm needs to be investigated in more depth and will be the object of future work. Finally, from Figure 2, we observe that the PPE-ROM provides the lowest errors for the velocity fields, while the pressure errors are comparable (for q) or worse (for $\bar{q}$) than the errors given by the SUP₂-ROM. Our findings for the EF algorithm are in agreement with what observed in [29] for NSE. Indeed, therein it is shown that the PPE model produces better results for the velocity field, but worse results for the pressure field when compared to the supremizer enrichment model. For insights on the behavior of the errors at the first and last time steps of the simulation, we refer to [26].

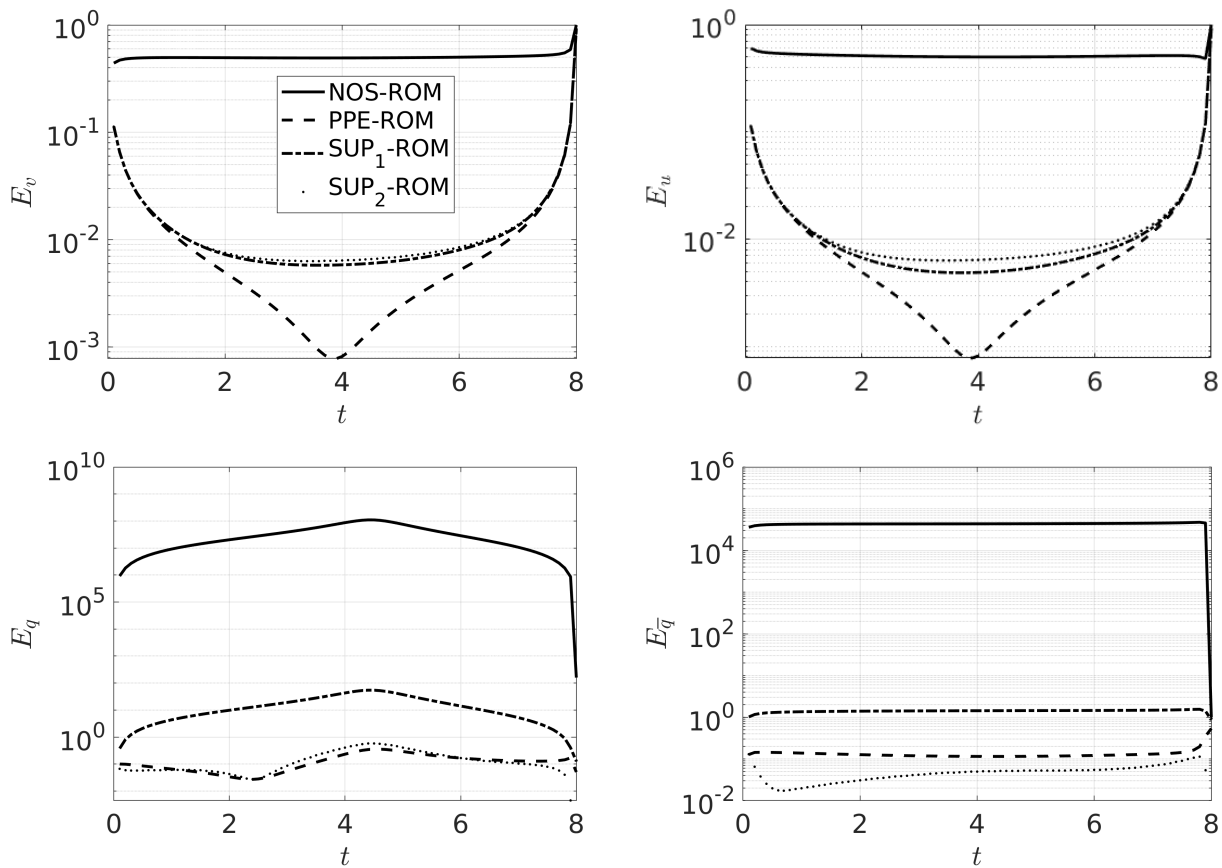


Figure 2. Time history of L^2 norm of the relative error for the velocity fields (**top**) and pressure fields (**bottom**) for NOS-ROM, PPE-ROM, SUP_1 -ROM, and SUP_2 -ROM. The legend in the top left panel is common to all the panels.

We report minimum, average, and maximum relative errors for PPE and SUP_2 in Table 2. The average errors for u and q are comparable to the values obtained in [29].

Table 2. Maximum, average, and minimum relative errors for the velocity and pressure fields for PPE-ROM and SUP_2 -ROM.

	v (PPE)	v (SUP_2)	u (PPE)	u (SUP_2)	q (PPE)	q (SUP_2)	$\bar{q}$ (PPE)	$\bar{q}$ (SUP_2)
Maximum E_Φ	9.1×10^{-1}	9.1×10^{-1}	9.2×10^{-1}	9.2×10^{-1}	3.6×10^{-1}	5.8×10^{-1}	5.4×10^{-1}	9.4×10^{-1}
Average E_Φ	2.3×10^{-2}	2.6×10^{-2}	2.4×10^{-2}	2.6×10^{-2}	1.4×10^{-1}	1.7×10^{-1}	1.3×10^{-1}	6×10^{-2}
Minimum E_Φ	7.8×10^{-4}	6.3×10^{-3}	7.8×10^{-4}	6.3×10^{-3}	2.7×10^{-2}	4.4×10^{-2}	1.1×10^{-1}	1.7×10^{-2}

For a visual comparison, we report in Figure 3 velocity and pressure fields at $t = 2$ computed by FOM, PPE-ROM, and SUP_2 -ROM. We observe that both PPE-ROM and SUP_2 -ROM can capture well the main flow features.

For a quantitative comparison of FOM, PPE-ROM, and SUP_2 -ROM, we consider the quantities of interest for this benchmark, i.e., the drag and lift coefficients [34,35]:

$$C_d(t) = \frac{2}{\rho L_r U_r^2} \int_S ((2\mu \nabla \mathbf{u} - q\mathbf{I}) \cdot \mathbf{n}) \cdot \mathbf{t} dS, \quad C_l(t) = \frac{2}{\rho L_r U_r^2} \int_S ((2\mu \nabla \mathbf{u} - q\mathbf{I}) \cdot \mathbf{n}) \cdot \mathbf{n} dS, \quad (70)$$

where $U_r = 1$ is the maximum velocity at the inlet/outlet, $L_r = 0.1$ is the cylinder diameter, S is the cylinder surface, and $\mathbf{t}$ and $\mathbf{n}$ are the tangential and normal unit vectors to the cylinder, respectively. See Figure 4 for the coefficients in (70) computed by the three approaches. We observe that the amplitude of both coefficients is slightly underestimated (resp., overestimated) by the PPE-ROM (resp., SUP_2 -ROM) over the entire time interval. The ROM reconstruction of the lift coefficient appears to be more critical, especially for the

SUP₂-ROM and around the center of the time interval. We note that, as expected [19], the over-diffusive nature of the EF algorithm (see Remark 2) used for both FOM and ROM smooths out the vortex shedding and completely dampens the related oscillations in the C_l .

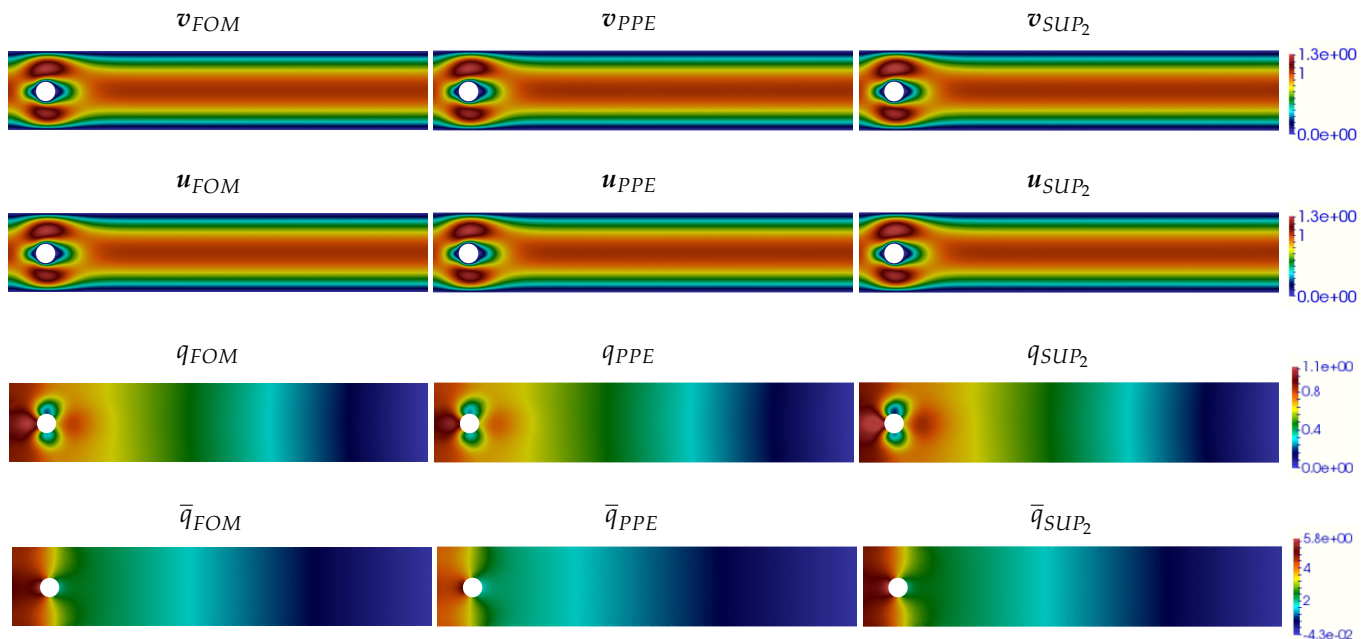


Figure 3. Velocity fields u (first row) and v (second row) and pressure fields q (third row) and $\bar{q}$ (fourth row) at time $t = 2$ computed by FOM (left), PPE-ROM (center), and SUP₂-ROM (right).

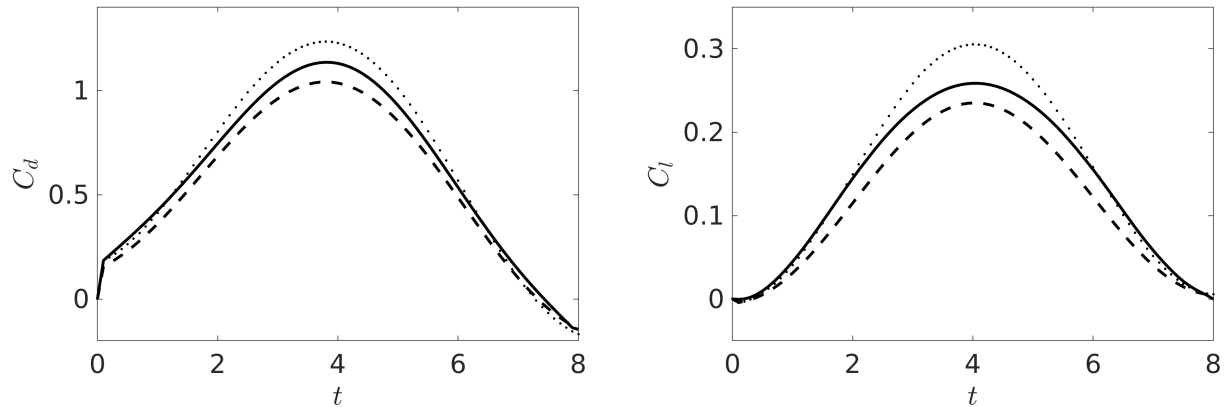


Figure 4. Aerodynamic coefficients C_d (left) and C_l (right) computed by FOM (solid line), SUP₂-ROM (dotted line), and PPE-ROM (dashed line).

For a further quantitative assessment of the reconstruction of the coefficients in (70), we computed the following errors

$$E_{c_d} = \frac{\|c_d(t)^{FOM} - c_d(t)^{ROM}\|_{L^2(0,8)}}{\|c_d(t)^{FOM}\|_{L^2(0,8)}}, \quad E_{c_l} = \frac{\|c_l(t)^{FOM} - c_l(t)^{ROM}\|_{L^2(0,8)}}{\|c_l(t)^{FOM}\|_{L^2(0,8)}}. \quad (71)$$

We notice that errors defined in (71) are different from the ones in [26], which are related to the maximum values only. Table 3 reports the errors (71) for PPE-ROM and SUP₂-ROM. We see that the two ROM strategies provide comparable results: slightly lower than 9% relative error for the the drag coefficient and around 14% relative error for the lift coefficient.

Table 3. Relative errors (71) for lift and drag coefficients for PPE-ROM and SUP₂-ROM.

	E_{c_d}	E_{c_l}
PPE-ROM	0.088	0.144
SUP ₂ -ROM	0.086	0.139

Finally, we provide some information on the computational cost. The total CPU time required by a FOM simulation is about 1900 s. The solution of the PPE-ROM algebraic systems (37), (51), (43) and (57) takes 1.61 s (resulting in a speed-up of about 1180), while the solution of the SUP₂-ROM systems (37), (38), (43) and (44) takes 2.56 s (resulting in a speed-up of about 742). So, the SUP₂-ROM is less efficient than the PPE-ROM. The additional cost of the SUP₂-ROM comes from the supremizer modes, which increases the size of the reduced dynamical system. However, it is possible to assert that both ROMs allow to obtain a considerable speed-up.

5. Conclusions and Future Perspectives

We presented a POD–Galerkin based reduced order method for a Leray model implemented through the Evolve-Filter (EF) algorithm. Unlike the large majority of the works on Leray-type models, we choose a Finite Volume method for the space discretization because of its computational efficiency. The novelty of this work is the investigation and comparison of two techniques for the stabilization of the pressure fields: (i) Poisson Pressure equation, and (ii) supremizer enrichment. We showed that the standard supremizer enrichment, which works well for the Navier–Stokes equations with no filter, needs to be modified in order to obtain stable and accurate solutions with the EF algorithm. The modification consists in adding to the evolve and filter velocity spaces the supremizer solutions related to both evolve and filter pressure fields. We assessed our ROM through the classical 2D flow past a cylinder benchmark. We found that our ROM with both Poisson Pressure equation and modified supremizer enrichment captures the flow features with an accuracy comparable to ROMs applied to the Navier–Stokes equations with no filter [29]. Moreover, we quantified the relative error of the drag and lift coefficients computed by ROM and FOM and found that both stabilization approaches produce comparable errors. We have limited our assessment to a 2D problem, although obviously more interesting applications of LES arise in 3D. We have no reason to believe that the conclusions drawn here do apply to 3D problems. The reader interested in 3D applications of our ROM approach with the Poisson Pressure equation is referred to [26,27].

In the future, we would like to investigate in more depth the inf-sup stability of a ROM formulation with supremizer enrichment for the EF algorithm. In addition, we would like to extend to the EF algorithm other efficient stabilization techniques, such as the one proposed in [59].

Author Contributions: Conceptualization, M.G., A.Q. and G.R.; methodology, M.G. and A.Q.; software, M.G.; supervision, A.Q. and G.R.; writing—original draft preparation, M.G.; writing—review and editing, A.Q. and G.R. All authors have read and agreed to the published version of the manuscript.

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

Conflicts of Interest: The authors declare no conflict of interest.

References

- Hesthaven, J.S.; Rozza, G.; Stamm, B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*; Springer: Cham, Switzerland, 2016.
- Quarteroni, A.; Manzoni, A.; Negri, F. *Reduced Basis Methods for Partial Differential Equations*; Springer: Cham, Switzerland, 2016.
- Benner, P.; Ohlberger, M.; Patera, A.; Rozza, G.; Urban, K. *Model Reduction of Parametrized Systems*, 1st ed.; MS&A Series; Springer: Cham, Switzerland, 2017.
- Benner, P.; Gugercin, S.; Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* **2015**, *57*, 483–531.
- Bader, E.; Kärcher, M.; Grepl, M.A.; Veroy, K. Certified Reduced Basis Methods for Parametrized Distributed Elliptic Optimal Control Problems with Control Constraints. *SIAM J. Sci. Comput.* **2016**, *38*, A3921–A3946.
- Benner, P.; Schilders, W.; Grivet-Talocia, S.; Quarteroni, A.; Rozza, G.; Silveira, L.M. *Model Order Reduction*; De Gruyter: Berlin, Germany; Boston, MA, USA, 2020.
- Longatte, E.; Erwan, L.; Pomarede, M.; Sigrist, J.F.; Hamdouni, A. Parametric study of flow-induced vibrations in cylinder arrays under single-phase fluid cross flows using POD-ROM. *J. Fluids Struct.* **2018**, *78*, 314–330.
- Shinde, V.; Longatte, E.; Baj, F.; Hoarau, Y.; Braza, M. A Galerkin-free model reduction approach for the Navier–Stokes equations. *J. Comput. Phys.* **2018**, *309*, 148–163.
- 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*, 623–647.
- Moin, P.; Mahesh, K. Direct Numerical Simulation: A Tool in Turbulence Research. *Annu. Rev. Fluid Mech.* **1998**, *30*, 539–578.
- Wang, Z.; Akhtar, I.; Borggaard, J.; Iliescu, T. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Comput. Methods Appl. Mech. Eng.* **2012**, *237–240*, 10–26.
- Aubry, N.; Holmes, P.; Lumley, J.L.; Stone, E. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *J. Fluid Mech.* **1988**, *192*, 115–173.
- Leray, J. Essai sur le mouvement d’un fluide visqueux emplissant l’espace. *J. Mathématiques Pures Appliquées* **1934**, *63*, 193–248.
- Bowers, A.; Rebholz, L. Numerical study of a regularization model for incompressible flow with deconvolution-based adaptive nonlinear filtering. *Comput. Methods Appl. Mech. Eng.* **2013**, *258*, 1–12.
- Boyd, J.P. Two Comments on Filtering (Artificial Viscosity) for Chebyshev and Legendre Spectral and Spectral Element Methods: Preserving Boundary Conditions and Interpretation of the Filter as a Diffusion. *J. Comput. Phys.* **1998**, *143*, 283–288.
- Fischer, P.; Mullen, J. Filter-based stabilization of spectral element methods. *Comptes Rendus L’Académie Sci. Ser. I Math.* **2001**, *332*, 265–270.
- Dunca, A.; Epshteyn, Y. On the Stolz-Adams deconvolution model for the large-eddy simulation of turbulent flows. *SIAM J. Math. Anal.* **2005**, *37*, 1890–1902.
- Layton, W.; Rebholz, L.; Trenchea, C. Modular Nonlinear Filter Stabilization of Methods for Higher Reynolds Numbers Flow. *J. Math. Fluid Mech.* **2012**, *14*, 325–354.
- Girfoglio, M.; Quaini, A.; Rozza, G. A Finite Volume approximation of the Navier–Stokes equations with nonlinear filtering stabilization. *Comput. Fluids* **2019**, *187*, 27–45.
- Girfoglio, M.; Quaini, A.; Rozza, G. Fluid–structure interaction simulations with a LES filtering approach in solids4Foam. *Commun. Appl. Ind. Math.* **2021**, accepted. Available online: <https://arxiv.org/abs/2102.08011> (accessed on 27 June 2021).
- Gunzburger, M.; Iliescu, T.; Schneier, M. A Leray regularized ensemble-proper orthogonal decomposition method for parameterized convection-dominated flows. *IMA J. Numer. Anal.* **2019**, *40*, 886–913.
- Xie, X.; Wells, D.; Wang, Z.; Iliescu, T. Numerical analysis of the Leray reduced order model. *J. Comput. Appl. Math.* **2018**, *328*, 12–29.
- Wells, D.; Wang, Z.; Xie, X.; Iliescu, T. An evolve-then-filter regularized reduced order model for convection-dominated flows. *Int. J. Numer. Methods Fluids* **2017**, *84*, 598–615.
- Gunzburger, M.; Iliescu, T.; Mohebbujaman, M.; Schneier, M. An Evolve-Filter-Relax stabilized reduced order stochastic collocation method for the time-dependent Navier–Stokes equations. *SIAM/ASA J. Uncertain. Quantif.* **2019**, *7*, 1162–1184.
- Xie, X.; Wells, D.; Wang, Z.; Iliescu, T. Approximate deconvolution reduced order modeling. *Comput. Methods Appl. Mech. Eng.* **2016**, *313*, 512–534.
- Girfoglio, M.; Quaini, A.; Rozza, G. A POD-Galerkin reduced order model for a LES filtering approach. *J. Comput. Phys.* **2021**, *436*, 110260.
- Girfoglio, M.; Quaini, A.; Rozza, G. A Hybrid Reduced Order Model for Nonlinear LES Filtering. Available online: <https://arxiv.org/abs/2107.12933> (accessed on 27 June 2021)
- Akhtar, I.; Nayfeh, A.H.; Ribbens, C.J. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theor. Comput. Fluid Dyn.* **2009**, *23*, 213–237.
- Stabile, G.; Rozza, G. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Comput. Fluids* **2018**, *173*, 273–284.
- Isoz, M. POD-DEIM based model order reduction for speed-up of flow parametric studies. *Ocean Eng.* **2019**, *186*, 106083.
- Rozza, G.; Veroy, K. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Eng.* **2007**, *196*, 1244–1260.

32. Ballarin, F.; Manzoni, A.; Quarteroni, A.; Rozza, G. Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *Int. J. Numer. Methods Eng.* **2014**, *102*, 1136–1161.
33. Gerner, A.L.; Veroy, K. Certified Reduced Basis methods for parametrized saddle point problems. *SIAM J. Sci. Comput.* **2011**, *34*, A2812–A2836.
34. Turek, S.; Schäfer, M. Benchmark computations of laminar flow around cylinder. In *Flow Simulation with High-Performance Computers II*; Hirschel, E., Ed.; Vieweg: Braunschweig/Wiesbaden, Germany, 1996; Volume 52.
35. John, V. Reference values for drag and lift of a two dimensional time-dependent flow around a cylinder. *Int. J. Numer. Methods Fluids* **2004**, *44*, 777–788.
36. Bertagna, L.; Quaini, A.; Veneziani, A. Deconvolution-based nonlinear filtering for incompressible flows at moderately large Reynolds numbers. *Int. J. Numer. Methods Fluids* **2016**, *81*, 463–488.
37. Quarteroni, A.; Sacco, R.; Saleri, F. *Numerical Mathematics*; Springer: Berlin/Heidelberg, Germany, 2007.
38. Weller, H.G.; Tabor, G.; Jasak, H.; Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **1998**, *12*, 620–631.
39. Issa, R.I. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.* **1986**, *62*, 40–65.
40. Van Doormaal, J.P.; Raithby, G.D. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numer. Heat Transf.* **1984**, *7*, 147–163.
41. Patankar, S.V.; Spalding, D.B. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transf.* **1972**, *15*, 1787–1806.
42. Stabile, G.; Rozza, G. ITHACA-FV—In Real Time Highly Advanced Computational Applications for Finite Volumes. Available online: <https://mathlab.sissa.it/ithaca-fv> (accessed on 30 January 2018).
43. Rozza, G.; Huynh, D.B.P.; Patera, A.T. Reduced Basis approximation and a Posteriori error estimation for affinely parametrized elliptic coercive Partial Differential Equations. *Arch. Comput. Methods Eng.* **2008**, *15*, 229.
44. Chinesta, F.; Huerta, A.; Rozza, G.; Willcox, K. Model Order Reduction. In *Encyclopedia of Computational Mechanics*; Wiley: Hoboken, NJ, USA, 2016.
45. Kalashnikova, I.; Barone, M.F. On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment. *Int. J. Numer. Methods Eng.* **2010**, *83*, 1345–1375.
46. Chinesta, F.; Ladeveze, P.; Cueto, E. A short review on Model Order Reduction based on Proper Generalized Decomposition. *Arch. Comput. Methods Eng.* **2011**, *18*, 395.
47. Dumon, A.; Allery, C.; Ammar, A. Proper General Decomposition (PGD) for the resolution of Navier–Stokes equations. *J. Comput. Phys.* **2011**, *230*, 1387–1407.
48. Tsiolakis, V.; Giacomini, M.; Sevilla, R.; Othmer, C.; Huerta, A. Parametric Solutions of Turbulent Incompressible Flows in OpenFOAM via the Proper Generalised Decomposition. Available online: <https://arxiv.org/abs/2006.07073> (accessed on 27 June 2021)
49. Kunisch, K.; Volkwein, S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.* **2002**, *40*, 492–515.
50. Rozza, G. Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. *Comput. Vis. Sci.* **2009**, *12*, 23–35.
51. Brezzi, F.; Bathe, K.J. A discourse on the stability conditions for mixed finite element formulations. *Comput. Methods Appl. Mech. Eng.* **1990**, *82*, 27–57.
52. Boffi, D.; Brezzi, F.; Fortin, M. *Mixed Finite Element Methods and Applications*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2013.
53. Bergmann, M.; Bruneau, C.H.; Iollo, A. Enablers for robust POD models. *J. Comput. Phys.* **2009**, *228*, 516–538.
54. 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. Eng.* **2016**, *311*, 151–179.
55. Orszag, S.A.; Israeli, M.; Deville, M. Boundary conditions for incompressible flows. *J. Sci. Comput.* **1986**, *1*, 75–111.
56. Johnston, H.; Liu, J.G. Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. *J. Comput. Phys.* **2004**, *199*, 221–259.
57. Li, L. A split-step finite-element method for incompressible Navier–Stokes equations with high-order accuracy up-to the boundary. *J. Comput. Phys.* **2020**, *408*, 213–237.
58. Lax, P.; Wendroff, B. System of conservation laws. *Commun. Pure Appl. Math.* **1960**, *13*, 217–237.
59. Stabile, G.; Zancanaro, M.; Rozza, G. Efficient Geometrical parametrization for finite-volume based reduced order methods. *Int. J. Numer. Methods Eng.* **2020**, *121*, 2655–2682.